Manual do usuário

AWS Criptografia de pagamento

AWS Criptografia de pagamento: Manual do usuário

Table of Contents

O que é a AWS Payment Cryptography?

A AWS Payment Cryptography é um serviço gerenciado da AWS que fornece acesso às funções criptográficas e ao gerenciamento de chaves usados no processamento de pagamentos de acordo com os padrões do payment card industry (PCI – setor de cartões de pagamento), sem a necessidade de adquirir instâncias HSM de pagamento dedicadas. AWS A Payment Cryptography fornece aos clientes que realizam funções de pagamento, como adquirentes, facilitadores de pagamento, redes, comutadores, processadores e bancos, a capacidade de aproximar suas operações criptográficas de pagamento dos aplicativos na nuvem e minimizar as dependências de datacenters auxiliares ou colocalização contendo HSMs de pagamento dedicados.

O serviço foi projetado para atender às regras aplicáveis do setor, incluindo PCI PIN, PCI P2PE e PCI DSS, e o serviço utiliza hardware com certificação PCI PTS HSM V3 e FIPS 140-2 Nível 3. Ele foi projetado para suportar baixa latência e altos níveis de disponibilidade e resiliência. AWS A Payment Cryptography é totalmente versátil e elimina muitos dos requisitos operacionais dos HSMs locais, como a necessidade de provisionar hardware, gerenciar com segurança o material essencial e manter backups de emergência em instalações seguras. AWS A Payment Cryptography também oferece a opção de compartilhar chaves eletronicamente com seus parceiros, eliminando a necessidade de compartilhar componentes de texto não criptografado em papel.

É possível usar a API do ambiente de gerenciamento de AWS Payment Cryptography para criar e gerenciar chaves.

É possível usar a API do plano de dados de AWS Payment Cryptography para usar chaves de criptografia para processamento de transações relacionadas a pagamentos e operações criptográficas associadas.

A AWS Payment Cryptography fornece atributos importantes que você pode usar para gerenciar suas chaves:

- Crie e gerencie chaves de AWS Payment Cryptography simétricas e assimétricas, incluindo chaves TDES, AES e RSA, e especifique a finalidade pretendida, como geração de CVV ou derivação de chaves DUKPT.
- Armazene automaticamente suas chaves de AWS Payment Cryptography com segurança, protegidas por hardware security modules (HSMs – módulos de segurança de hardware) e, ao mesmo tempo, reforçando a separação de chaves entre os casos de uso.
- Crie, exclua, liste e atualize aliases, que são “nomes amigáveis” que podem ser usados para acessar ou controlar o acesso às suas chaves de AWS Payment Cryptography.

- Marque suas chaves de AWS Payment Cryptography com tags para identificação, agrupamento, automação, controle de acesso e rastreamento de custos.
- Importe e exporte chaves simétricas entre a AWS Payment Cryptography e seu HSM (ou terceiros) usando Key Encryption Keys (KEK – chaves de criptografia de chave) seguindo a TR-31 (especificação interoperável de blocos de chaves de troca segura de chaves).
- Importe e exporte KEKs simétricas entre AWS Payment Cryptography e outros sistemas usando pares de chaves assimétricas, seguido pelo uso de meios eletrônicos, como o TR-34 (método para distribuição de chaves simétricas usando técnicas assimétricas).

É possível usar suas chaves de AWS Payment Cryptography em operações criptográficas como:

- Criptografar, descriptografar e recriptografar dados com chaves de AWS Payment Cryptography simétricas ou assimétricas.
- Traduzir com segurança dados confidenciais (como PINs do titular do cartão) entre chaves de criptografia sem expor o texto não criptografado de acordo com as regras de PCI PIN.
- Gerar ou validar dados do titular do cartão, como CVV, CVV2 ou ARQC.
- Gerar e validar PINs do titular do cartão.
- Gerar ou validar assinaturas MAC.

Conceitos

Aprenda os termos e conceitos básicos usados na AWS Payment Cryptography e como você pode usá-los para ajudar a proteger seus dados.

Alias

Um nome fácil de usar associado a uma chave de AWS Payment Cryptography. O alias pode ser usado de forma intercambiável com o ARN de chave em muitas das operações da API de AWS Payment Cryptography API. Os aliases permitem que as chaves sejam alternadas ou alteradas sem afetar o código do aplicativo. O nome do alias é uma string com até 256 caracteres. Ele identifica exclusivamente uma chave de AWS Payment Cryptography associada a uma conta e região. Na AWS Payment Cryptography, os alias sempre começam com `alias/`.

O formato de um nome de alias é o seguinte:

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

Por exemplo:

```
alias/sampleAlias2
```

ARN de chave

O ARN de chave é o nome do recurso da Amazon (ARN) de uma entrada de chave na AWS Payment Cryptography. É um identificador exclusivo e totalmente qualificado para a chave de AWS Payment Cryptography. Um ARN de chave inclui uma Conta da AWS, região e ID gerados aleatoriamente. O ARN não está relacionado nem é derivado do material de chave. Como eles são atribuídos automaticamente durante as operações de criação ou importação, esses valores não são idempotentes. Importar a mesma chave várias vezes resultará em vários ARNs de chave com seu próprio ciclo de vida.

O formato de um ARN de chave é o seguinte:

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

Este é um exemplo de ARN de chave:

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

Chaves de AWS Payment Cryptography

As chaves de AWS Payment Cryptography (chaves) são usadas para todas as funções criptográficas. As chaves são geradas diretamente por você usando o comando de criação de chaves ou adicionadas ao sistema ao chamar a importação de chaves. A origem de uma chave pode ser determinada revisando o atributo KeyOrigin. AWS A Payment Cryptography também suporta chaves derivadas ou intermediárias usadas durante operações criptográficas, como as usadas por DUKPT.

Essas chaves possuem atributos imutáveis e mutáveis definidos na criação. Atributos como algoritmo, comprimento e uso são definidos na criação e não podem ser alterados. Outros, como data de vigência ou data de expiração, podem ser modificados. Consulte as Referências da API de AWS Payment Cryptography para obter uma lista completa dos principais atributos da AWS Payment Cryptography.

As chaves de AWS Payment Cryptography têm tipos de chaves, definidos principalmente pela ANSI X9 TR 31, que restringem seu uso à finalidade pretendida, conforme especificado no Requisito 19 do PCI PIN v3.1.

Os atributos são vinculados às chaves usando blocos de chaves quando armazenados, compartilhados com outras contas ou exportados conforme especificado no Requisito 18-3 do PCI PIN v3.1.

As chaves são identificadas na plataforma de AWS Payment Cryptography usando um valor exclusivo conhecido como chave nome do recurso da Amazon (ARN).

> **Note**
>
> A chave ARN é gerada quando uma chave é inicialmente criada ou importada para o serviço de AWS Payment Cryptography. Portanto, se você adicionar o mesmo material de chave várias vezes usando a funcionalidade de importação de chave, o mesmo material de chave estará localizado em várias chaves de , mas cada um com um ciclo de vida de chave diferente.

Terminologia do setor

Tópicos

Tipos de chaves comuns

AWK

Uma chave de trabalho do adquirente (AWK) é uma chave normalmente usada para trocar dados entre um processador adquirente/adquirente e uma rede (como Visa ou Mastercard). Historicamente, a AWK utiliza 3DES para criptografia e seria representada como TR31_P0_PIN_ENCRYPTION_KEY.

BDK

Uma chave de derivação de base (BDK) é uma chave de trabalho usada para derivar chaves subsequentes e é comumente usada como parte do processo PCI PIN e PCI P2PE DUKPT. É indicada como TR31_B0_BASE_DERIVATION_KEY.

CMK

Uma chave mestra do cartão (CMK) corresponde a uma ou mais chaves específicas do cartão, normalmente derivadas de uma chave mestra do emissor, PAN e PSN e geralmente são chaves 3DES. Essas chaves são armazenadas no chip EMV durante a personalização. Exemplos de CMKs incluem chaves AC, SMI e SMC.

CMK-AC

Uma chave de criptograma de aplicativo (AC) é usada como parte das transações EMV para gerar o criptograma da transação e é um tipo de chave mestra do cartão.

CMK-SMI

Uma chave de integridade segura de mensagens (SIM) é usada como parte do EMV para verificar a integridade das cargas enviadas ao cartão usando MAC, como scripts de atualização de PIN. É um tipo de chave mestra de cartão.

CMK-SMC

Uma chave de confidencialidade segura de mensagens (SMC) é usada como parte do EMV para criptografar dados enviados ao cartão, como atualizações de PIN. É um tipo de chave mestra de cartão.

CVK

Uma chave de verificação de cartão (CVK) é uma chave usada para gerar CVV, CVV2 e valores similares usando um algoritmo definido, além de validar uma entrada. É indicada como TR31_C0_CARD_VERIFICATION_KEY.

IMK

Uma chave mestra do emissor (IMK) é uma chave mestra usada como parte da personalização do cartão com chip EMV. Normalmente, haverá três IMKs, uma para cada chave AC (criptograma), SMI (chave mestra de script para integridade/assinatura) e SMC (chave mestra de script para confidencialidade/criptografia).

IK

Uma chave inicial (IK) é a primeira chave usada no processo DUKPT e deriva da Chave de Derivação Base (BDK). Nenhuma transação é processada nessa chave, mas ela é usada para derivar chaves futuras que serão usadas para transações. O método de derivação para criar um IK foi definido em X9. 24-1:2017. Quando um TDES BDK é usado, o X9. 24-1:2009 é o padrão aplicável e o IK é substituído pela Chave de Criptografia de Pino Inicial (IPEK).

IPEK

Uma chave de criptografia PIN inicial (IPEK) é a chave inicial usada no processo de DUKPT e deriva da chave de derivação base (BDK). Nenhuma transação é processada nessa chave, mas ela é usada para derivar chaves futuras que serão usadas para transações. IPEK é um nome impróprio, pois essa chave também pode ser usada para derivar criptografia de dados e chaves mac. O método de derivação para criar um IPEK foi definido em X9. 24-1:2009. Quando um AES BDK é usado, o X9. 24-1:2017 é o padrão aplicável e o IPEK é substituído pela Chave Inicial (IK).

IWK

Uma chave de trabalho do emissor (IWK) é uma chave normalmente usada para trocar dados entre um emissor/processador emissor e uma rede (como Visa ou Mastercard). Historicamente, a IWK utiliza o 3DES para criptografia e é representada como TR31_P0_PIN_ENCRYPTION_KEY.

KEK

Uma chave de criptografia de chave (KEK) é uma chave usada para criptografar outras chaves para transmissão ou armazenamento. As chaves destinadas a proteger outras chaves geralmente têm um KeyUsage TR31_K0_KEY_ENCRYPTION_KEY de acordo com o padrão. TR-31

PEK

Uma chave de criptografia de PIN (PEK) é um tipo de chave de trabalho usada para criptografar PINs para armazenamento ou transmissão entre duas partes. IWK e AWK são dois exemplos de usos específicos de chaves de criptografia de PINs. Essas chaves são representadas como TR31_P0_PIN_ENCRYPTION_KEY.

PVK

Uma chave de verificação de PIN (PVK) é um tipo de chave de trabalho usada para gerar valores de verificação de PIN, como PVV. Os dois tipos mais comuns são TR31_V1_IBM3624_PIN_VERIFICATION_KEY, usada para gerar valores de deslocamento IBM3624 e TR31_V2_VISA_PIN_VERIFICATION_KEY, usada para valores de verificação Visa/ABA.

Outros termos

ARQC

O criptograma de solicitação de autorização (ARQC) é um criptograma gerado por um cartão com chip padrão EMV (ou implementação sem contato equivalente) no momento da transação.

Normalmente, um ARQC é gerado por um cartão com chip e encaminhado a um emissor ou seu agente para verificação no momento da transação.

DUKPT

A chave única derivada por transação (DUKPT) é um padrão de gerenciamento de chaves normalmente usado para definir o uso de chaves de criptografia de uso único em POS/POI físico. Historicamente, a DUKPT utiliza 3DES para criptografia. O padrão do setor para DUKPT é definido na ANSI X9.24-3-2017.

EMV

A EMV (originalmente Europay, Mastercard, Visa) é um órgão técnico que trabalha com as partes interessadas em pagamentos para criar padrões e tecnologias de pagamento interoperáveis. Um exemplo de padrão é para cartões com chip/sem contato e os terminais de pagamento com os quais eles interagem, incluindo a criptografia usada. A derivação de chave EMV se refere ao (s) método (s) de geração de chaves exclusivas para cada cartão de pagamento com base em um conjunto inicial de chaves, como um `IMK`

HSM

Um módulo de segurança de hardware (HSM) é um dispositivo físico que protege as operações criptográficas (por exemplo, criptografia, decodificação e assinaturas digitais), bem como as chaves subjacentes usadas para essas operações.

KCV

O valor de verificação de chave (KCV) se refere a uma variedade de métodos de soma de verificação usados principalmente para comparar as chaves entre si sem ter acesso ao material real da chave. O KCV também tem sido usado para validação de integridade (especialmente ao trocar chaves), embora essa função agora esteja incluída como parte de formatos de blocos de chaves, como `TR-31`. Para chaves TDES, o KCV é calculado criptografando 8 bytes, cada um com valor zero, com a chave a ser verificada e retendo os 3 bytes de ordem mais alta do resultado criptografado. Para chaves AES, o KCV é calculado usando um algoritmo CMAC em que os dados de entrada são 16 bytes de zero e retêm os 3 bytes de ordem mais alta do resultado criptografado.

KDH

Um host de distribuição de chaves (KDH) é um dispositivo ou sistema que está enviando chaves em um processo de troca de chaves, como o TR-34. Ao enviar chaves da criptografia de AWS pagamento, ela é considerada o KDH.

KIF

Uma instalação de injeção de chave (KIF) é um recurso seguro usado para inicializar terminais de pagamento, incluindo carregá-los com chaves de criptografia.

KRD

Um dispositivo de recebimento de chaves (KRD) é um dispositivo que está recebendo chaves em um processo de troca de chaves, como o TR-34. Ao enviar chaves para criptografia AWS de pagamento, ela é considerada o KRD.

KSN

Um número de série de chave (KSN) é um valor usado como entrada para criptografia/descriptografia DUKPT para criar chaves de criptografia exclusivas por transação. Normalmente, o KSN consiste em um identificador BDK, um ID de terminal semi-exclusivo e um contador de transações, que é incrementado em cada transição processada em um determinado terminal de pagamento.

PAN

Um número primário de conta (PAN) é um identificador exclusivo para uma conta, como um cartão de crédito ou débito. Normalmente, tem de 13 a 19 dígitos de comprimento. Os primeiros 6 a 8 dígitos identificam a rede e o banco emissor.

Bloco de PIN

Um bloco de dados contendo um PIN durante o processamento ou transmissão, bem como outros elementos de dados. Os formatos de bloco de PIN padronizam o conteúdo do bloco de PIN e como ele pode ser processado para recuperar o PIN. A maioria dos blocos de PIN é composta pelo PIN, pelo comprimento do PIN e frequentemente contém parte ou a totalidade do PAN. AWS A criptografia de pagamento suporta os formatos ISO 9564-1 0, 1, 3 e 4. O Formato 4 é necessário para chaves AES. Ao verificar ou traduzir PINs, é necessário especificar o bloco de PIN dos dados de entrada ou saída.

POI

O ponto de interação (POI), também usado frequentemente como sinônimo de ponto de venda (POS), é o dispositivo de hardware com o qual o titular do cartão interage para apresentar sua credencial de pagamento. Um exemplo de POI é o terminal físico em um estabelecimento comercial. Para obter a lista de terminais PCI PTS POI certificados, consulte o site da PCI.

PSN

O número de sequência PAN (PSN) é um valor numérico usado para diferenciar vários cartões emitidos com o mesmo PAN.

Chave pública

Ao usar cifras assimétricas (RSA), a chave pública é o componente público de um par de chaves público-privadas. A chave pública pode ser compartilhada e distribuída para entidades que precisam criptografar dados para o proprietário do par de chaves público-privadas. Para operações de assinatura digital, a chave pública é usada para verificar a assinatura.

Chave privada

Ao usar cifras assimétricas (RSA), a chave privada é o componente privado de um par de chaves público-privadas. A chave privada é usada para descriptografar dados ou criar assinaturas digitais. Semelhante às chaves simétricas AWS de criptografia de pagamento, as chaves privadas são criadas com segurança pelos HSMs. Elas são descriptografadas somente na memória volátil do HSM e somente pelo tempo necessário para processar sua solicitação criptográfica.

PVV

O valor de verificação do PIN (PVV) é um valor derivado algoritmicamente de uma série de entradas, como número do cartão e PIN, que gera um valor que pode ser usado para validação posterior. Um desses esquemas é conhecido como Visa PVV (também conhecido como método ABA), embora seja usado para PINs em qualquer rede.

Embrulhe/Desembrulhe RSA

O RSA wrap usa uma chave assimétrica para encapsular uma chave simétrica (como uma chave TDES) para transmissão para outro sistema. Somente o sistema com a chave privada correspondente pode descriptografar a carga e carregar a chave simétrica. Por outro lado, o RSA unwrap decodificará com segurança uma chave criptografada usando RSA e, em seguida, carregará a chave na Criptografia de Pagamento. AWS O RSA wrap é um método de troca de chaves de baixo nível e não transmite chaves no formato de bloco de chaves e não utiliza a assinatura de carga útil pela parte remetente. Controles alternativos devem ser considerados para verificar se a providência e os principais atributos não estão alterados.

O TR-34 também utiliza RSA internamente, mas é um formato separado e não é interoperável.

TR-31

O TR-31 (formalmente definido coma ANSI X9 TR 31) é um formato de bloco de chave definido pelo American National Standards Institute (ANSI) para oferecer suporte à definição de atributos

de chave na mesma estrutura de dados dos próprios dados de chave. O formato de bloco de teclas TR-31 define um conjunto de atributos-chave que são vinculados à chave para que sejam mantidos juntos. AWS A criptografia de pagamento usa termos padronizados do TR-31 sempre que possível para garantir a separação adequada das chaves e o propósito da chave. O TR-31 foi substituído pela ANSI X9.143-2022.

TR-34

O TR-34 é uma implementação do ANSI X9.24-2 que descreve um protocolo para distribuir chaves simétricas com segurança (como 3DES e AES) usando técnicas assimétricas (como RSA). AWS A criptografia de pagamento usa métodos TR-34 para permitir a importação e exportação seguras de chaves.

Serviços relacionados

AWS Key Management Service

O AWS Key Management Service (AWS KMS) é um serviço gerenciado que facilita a criação e o controle de chaves criptográficas usadas para proteger seus dados. AWS O KMS usa módulos de segurança de hardware (HSMs) para proteger e validar suas chaves do AWS KMS.

AWS CloudHSM

O AWS CloudHSM fornece aos clientes instâncias HSM dedicadas de uso geral na nuvem da AWS. O AWS CloudHSM pode fornecer uma variedade de funções criptográficas, como criação de chaves, assinatura de dados ou criptografia e descriptografia de dados.

Para obter mais informações

- Para saber mais sobre os termos e conceitos usados na AWS Payment Cryptography, consulte Conceitos de AWS Payment Cryptography.
- Para obter informações sobre a API do ambiente de gerenciamento de AWS Payment Cryptography, consulte as Referências da API do ambiente de gerenciamento de AWS Payment Cryptography.
- Para obter informações sobre a API do plano de dados de AWS Payment Cryptography, consulte as Referências da API do plano de dados da AWS Payment Cryptography.
- Para obter informações técnicas detalhadas sobre como a AWS Payment Cryptography usa criptografia e protege as chaves da AWS Payment Cryptography, consulte Detalhes criptográficos.

Endpoints para AWS Payment Cryptography

Para se conectar programaticamente AWS Payment Cryptography, você usa um endpoint, a URL do ponto de entrada do serviço. Os AWS SDKs e as ferramentas de linha de comando usam automaticamente o endpoint padrão do serviço Região da AWS com base no contexto regional de uma solicitação, portanto, normalmente não há necessidade de definir explicitamente esses valores. Quando necessário, você pode especificar um endpoint diferente para suas solicitações de API.

Endpoints do ambiente de gerenciamento

Nome da região	Região	Endpoint	Protocolo
Leste dos EUA (Norte da Virgínia)	`us-east-1`	controlplane.payment-cryptography.us-east-1.amazonaws.com	HTTPS
Leste dos EUA (Ohio)	`us-east-2`	plano de controle. payment-cryptography.us-east-2.amazonaws.com	HTTPS
Oeste dos EUA (Oregon)	`us-west-2`	controlplane.payment-cryptography.us-west-2.amazonaws.com	HTTPS

Endpoints do plano de dados

Nome da região	Região	Endpoint	Protocolo
Leste dos EUA (Norte da Virgínia)	`us-east-1`	dataplane.payment-cryptography.us-east-1.amazonaws.com	HTTPS
Leste dos EUA (Ohio)	`us-east-2`	plano de dados. payment-cryptography.us-east-2.amazonaws.com	HTTPS
Oeste dos EUA (Oregon)	`us-west-2`	dataplane.payment-cryptography.us-west-2.amazonaws.com	HTTPS

Conceitos básicos da AWS Payment Cryptography

Para começar a usar a AWS Payment Cryptography, primeiro você deve criar chaves e depois usá-las em várias operações criptográficas. O tutorial abaixo fornece um caso de uso simples para gerar uma chave a ser usada para gerar/verificar valores CVV2. Para conhecer outros exemplos e explorar padrões de implantação na AWS, experimente o seguinte Workshop de AWS Payment Cryptography ou explore nosso projeto de amostras disponíveis no Github

Este tutorial explica como criar uma chave única e realizar operações criptográficas usando a chave. Depois disso, você exclui a chave se não tiver uso para ela, o que completa o ciclo de vida da chave.

Tópicos

Pré-requisitos

Antes de começar, verifique se:

- Você tem permissão para acessar o serviço. Para obter mais informações, consulte as políticas do IAM.
- Você tem o AWS CLI instalado. Também é possível usar SDKs da AWS ou APIs da AWS para acessar a AWS Payment Cryptography, mas as instruções deste tutorial usam AWS CLI.

Etapa 1: criar uma chave

A primeira etapa é criar uma chave. Neste tutorial, você cria uma chave CVK 3DES de comprimento duplo (2KEY TDES) para gerar e verificar valores CVV/CVV2.

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

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

A resposta reflete os parâmetros da solicitação, incluindo um ARN para chamadas subsequentes, bem como um valor de verificação chave (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"
    }
}
```

Anote o `KeyArn` que representa a chave, por exemplo, arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi. Isso será necessário na próxima etapa.

Etapa 2: gerar um valor CVV2 usando a chave

Nesta etapa, você gera um CVV2 para um determinado PAN e data de validade usando a chave da etapa 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"
}
```

Anote o `cardDataValue`; neste caso, o número 144, de 3 dígitos. Isso será necessário na próxima etapa.

Etapa 3: verificar o valor gerado na etapa 2

Neste exemplo, você valida o CVV2 da etapa 2 usando a chave criada na etapa 1.

Execute o seguinte comando para validar o 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"
```

```
}
```

O serviço retorna uma resposta HTTP de 200 para indicar que validou o CVV2.

Etapa 4: realizar um teste negativo

Nesta etapa você cria um teste negativo em que o CVV2 está incorreto e não é validado. Você tenta validar um CVV2 incorreto usando a chave que criou na etapa 1. Essa é uma operação esperada, por exemplo, se o titular do cartão digitou o CVV2 errado na finalização da 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.
```

O serviço retorna uma resposta HTTP de 400 com a mensagem “Falha na verificação dos dados de validação do cartão” e um motivo INVALID_VALIDATION_DATA.

Etapa 5 (opcional): limpeza

Agora, você pode excluir a chave criada na etapa 1. Para minimizar as alterações irrecuperáveis, o período padrão da exclusão da chave é de sete dias.

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

```
{
    "Key": {
        "CreateTimestamp": "2022-10-27T08:27:51.795000-07:00",
        "DeletePendingTimestamp": "2022-11-03T13:37:12.114000-07:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
```

```
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P0_PIN_ENCRYPTION_KEY"
        },
        "KeyCheckValue": "CADDA1",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "DELETE_PENDING",
        "UsageStartTimestamp": "2022-10-27T08:27:51.753000-07:00"
    }
}
```

Observe os dois campos na saída. Por padrão, `deletePendingTimestamp` é definido para sete dias no futuro. O keyState está definido como `DELETE_PENDING`. Você pode cancelar esta exclusão a qualquer momento antes do horário programado ao chamar `restore-key`.

Managing keys

Para começar a usar a criptografia de AWS pagamento, crie uma chave de criptografia AWS de pagamento.

Os tópicos desta seção explicam como criar e gerenciar vários tipos de chaves de criptografia de AWS pagamento, da criação à exclusão. Ela inclui tópicos sobre criação, edição e visualização de chaves, tags de chaves, criação de aliases de chaves, assim como a habilitação e desabilitação de chaves.

Tópicos

Gerar chaves

Você pode criar chaves AWS de criptografia de pagamento usando a operação da CreateKey API. Durante esse processo, você especificará vários atributos da chave ou da saída resultante, como o algoritmo de chave (por exemplo, TDES_3KEY), (por exemplo, TR31_P0_PIN_ENCRYPTION_KEY), operações permitidas KeyUsage (por exemplo, criptografia, assinatura) e se é exportável. Você não pode alterar essas propriedades após a criação da chave de criptografia de AWS pagamento.

Exemplos

- Gerar uma chave de valor de verificação de PIN (PVV)

Gerar uma chave TDES 2KEY

Example

Esse comando gera uma chave TDES 2KEY com a finalidade de gerar e verificar valores de CVV/CVV2. A resposta reflete os parâmetros da solicitação, incluindo um ARN para chamadas subsequentes, bem como um valor de verificação de chave (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}'
```


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

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

Gerar uma chave de criptografia de PIN

Example Gerar uma chave de criptografia de PINs (PEK)

Este comando gera uma chave 3KEY TDES com a finalidade de criptografar valores PIN (conhecida como chave de criptografia PIN). Essa chave pode ser usada para proteger o armazenamento de PINs ou para descriptografar os PINs fornecidos durante uma tentativa de verificação, por exemplo, durante uma transação. A resposta reflete os parâmetros da solicitação, incluindo um ARN para chamadas subsequentes, bem como um valor de verificação de chave (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}'
```

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

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

Criar uma chave assimétrica (RSA)

Example

Neste exemplo, geraremos um novo par de chaves RSA assimétricas de 2048 bits. Uma nova chave privada será gerada, bem como a chave pública correspondente. A chave pública pode ser recuperada usando a getPublicCertificateAPI.

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

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

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

Gerar uma chave de valor de verificação de PIN (PVV)

Example

Este comando gera uma chave 3KEY TDES com a finalidade de gerar valores PVV (conhecidos como valor de verificação de PIN). É possível usar esta chave para gerar um valor de PVV que pode ser comparado com um PVV calculado subsequentemente. A resposta reflete os parâmetros da solicitação, incluindo um ARN para chamadas subsequentes, bem como um valor de verificação de chave (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}'
```

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

```
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": false,
                "DeriveKey": false,
                "Encrypt": false,
                "Generate": true,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": false,
                "Verify": true,
                "Wrap": false
            },
            "KeyUsage": "TR31_V2_VISA_PIN_VERIFICATION_KEY"
        },
        "KeyCheckValue": "5132",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-10-27T10:22:59.614000-07:00"
    }
}
```

Listar chaves

Listar chaves apresenta uma lista de chaves acessíveis ao chamador nesta conta e região.

Example

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

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

```
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_P1_PIN_GENERATION_KEY"
        },
        "KeyCheckValue": "369D",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStopTimestamp": "2022-10-27T14:19:42.488000-07:00"
    }
    ]
}
```


Habilitar e desabilitar chaves

Você pode desativar e reativar as chaves AWS de criptografia de pagamento. Quando uma chave é criada, ela é habilitada por padrão. Se você desabilitar uma chave, ela não poderá ser usada em nenhuma operação criptográfica até que seja reabilitada. Os comandos para iniciar/interromper o uso têm efeito imediato, por isso é conveniente que você revise o uso antes de fazer essa alteração. Também é possível definir uma alteração (iniciar ou interromper o uso) para entrar em vigor no futuro usando o parâmetro opcional `timestamp`.

Por ser temporário e fácil de desfazer, desativar uma chave de criptografia de AWS pagamento é uma alternativa mais segura do que excluir uma chave de criptografia de AWS pagamento, uma ação destrutiva e irreversível. Se você estiver pensando em excluir uma chave de criptografia de AWS pagamento, desative-a primeiro e certifique-se de que não precisará usar a chave para criptografar ou descriptografar dados no futuro.

Tópicos

Iniciar o uso de chaves

O uso de chaves deve ser habilitado para que seja possível usar uma chave para operações criptográficas. Se uma chave não estiver ativada, você usará essa operação para torná-la utilizável. O campo `UsageStartTimestamp` representará quando a chave ficou/ficará ativa. Isso acontecerá no passado para um token ativado e no futuro se a ativação estiver pendente.

Example

Neste exemplo, solicita-se que uma chave seja habilitada para que seja usada. A resposta inclui as principais informações e o sinalizador de ativação foi alterado para verdadeiro. Isso também será refletido no objeto de resposta list-keys.

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

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

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

Interromper o uso de chaves

Se você não planeja mais usar uma chave, pode interromper o uso dela para evitar mais operações criptográficas. Essa operação não é permanente e você pode revertê-la usando Iniciar o uso de chaves. Também é possível definir uma chave para ser desativada no futuro. O campo `UsageStopTimestamp` representará quando a chave foi/será desativada.

Example

Neste exemplo, é solicitado que você interrompa o uso da chave no futuro. Após a execução, essa chave não pode ser usada para operações criptográficas, a menos que seja reativada por meio da opção Iniciar o uso de chaves. A resposta inclui as informações da chave e o sinalizador de ativação foi alterado para falso. Isso também será refletido no objeto de resposta list-keys.

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

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

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

Excluir chaves

A exclusão de uma chave AWS de criptografia de pagamento exclui o material da chave e todos os metadados associados à chave e é irreversível, a menos que uma cópia da chave esteja disponível fora da criptografia de pagamento. AWS Após a exclusão de uma chave, não é mais possível descriptografar os dados criptografados sob essa chave, o que significa que os dados podem se tornar irrecuperáveis. Você deve excluir uma chave somente quando tiver certeza de que não precisará mais usá-la e que nenhuma outra pessoa a utiliza. Caso não tenha certeza, desabilite a chave em vez de excluí-la. Você pode reativar uma chave desativada se precisar usá-la novamente mais tarde, mas não poderá recuperar uma chave de criptografia de AWS pagamento excluída, a menos que consiga reimportá-la de outra fonte.

Antes de excluir uma chave, você deve se certificar de que não precisa mais da chave. AWS A criptografia de pagamento não armazena os resultados de operações criptográficas como o CVV2 e não consegue determinar se uma chave é necessária para qualquer material criptográfico persistente.

AWS A criptografia de pagamento nunca exclui chaves pertencentes a AWS contas ativas, a menos que você as agende explicitamente para exclusão e o período de espera obrigatório expire.

No entanto, você pode optar por excluir uma chave de criptografia de AWS pagamento por um ou mais dos seguintes motivos:

- Para concluir o ciclo de vida de uma chave que você não precisa mais

- Para evitar a sobrecarga de gerenciamento associada à manutenção de chaves de criptografia de AWS pagamento não utilizadas

> **Note**
>
> Se você fechar ou excluir sua Conta da AWS, sua chave AWS de criptografia de pagamento ficará inacessível. Você não precisa agendar a exclusão da sua chave de criptografia de AWS pagamento separadamente do fechamento da conta.

AWS A criptografia de pagamento registra uma entrada em seu AWS CloudTrailregistro quando você agenda a exclusão da chave de criptografia de AWS pagamento e quando a chave de criptografia de AWS pagamento é realmente excluída.

Sobre o período de espera

Como a exclusão de uma chave é irreversível, a criptografia de AWS pagamento exige que você defina um período de espera entre 3 e 180 dias. O período de espera padrão é de sete dias.

No entanto, o período de espera real pode ser até 24 horas mais longo do que o programado. Para obter a data e a hora reais em que a chave AWS de criptografia de pagamento será excluída, use as GetKey operações. Certifique-se de anotar o fuso horário.

Durante o período de espera, o status e o estado da chave de criptografia de AWS pagamento são Exclusão pendente.

> **Note**
>
> Uma chave AWS de criptografia de pagamento pendente de exclusão não pode ser usada em nenhuma operação criptográfica.

Após o término do período de espera, a Criptografia AWS de Pagamento exclui a chave de Criptografia AWS de Pagamento, seus aliases e todos os metadados relacionados à AWS Criptografia de Pagamento.

Use o período de espera para garantir que você não precise da chave AWS de criptografia de pagamento agora ou no futuro. Se você achar que precisa da chave durante o período de espera,

basta cancelar a exclusão de chaves antes do término do período de espera. Após o término do período de espera, você não poderá cancelar a exclusão da chave e o serviço excluirá a chave.

Example

Neste exemplo, é solicitada a exclusão de uma chave. Além das informações básicas da chave, dois campos relevantes são que o estado da chave foi alterado para DELETE_PENDING e deletePendingTimestamp representa quando a chave está programada para ser excluída no momento.

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

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

```
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "CreateTimestamp": "2023-06-05T12:01:29.969000-07:00",
        "UsageStopTimestamp": "2023-06-05T14:31:13.399000-07:00",
        "DeletePendingTimestamp": "2023-06-12T14:58:32.865000-07:00"
    }
}
```

Example

Neste exemplo, uma exclusão pendente é cancelada. Depois de concluída com sucesso, uma chave não será mais excluída de acordo com a programação anterior. A resposta contém as principais informações básicas; além disso, dois campos relevantes foram alterados: `KeyState` e `deletePendingTimestamp`. `KeyState` é retornado para um valor de CREATE_COMPLETE, enquanto `DeletePendingTimestamp` é removido.

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

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

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

Importar e exportar chaves

AWS As chaves de criptografia de pagamento podem ser importadas de outras soluções ou exportadas para outras soluções (como outros HSMs). É um caso de uso comum trocar chaves com provedores de serviços usando a funcionalidade de importação e exportação. Como um serviço em nuvem, a criptografia de AWS pagamento adota uma abordagem eletrônica moderna para o gerenciamento de chaves, ajudando você a manter a conformidade e os controles aplicáveis. O objetivo de longo prazo é abandonar os principais componentes impressos e adotar meios eletrônicos de troca de chaves baseados em padrões.

Troca de chave de criptografia de chave (KEK)

AWS A criptografia de pagamento incentiva o uso da criptografia de chave pública (RSA) para a troca inicial de chaves usando a bem estabelecida norma ANSI X9.24 TR-34. Os nomes comuns desse tipo de chave inicial incluem chave de criptografia de chave (KEK), chave mestra de zona (ZMK) e chave mestra de controle de zona (ZCMK). Se seus sistemas ou parceiros ainda não puderem oferecer suporte ao TR-34, você também pode considerar a utilização do RSA Wrap/Unwrap.

Se precisar continuar processando os componentes principais em papel até que todos os parceiros ofereçam suporte à troca eletrônica de chaves, considere a possibilidade de manter um HSM off-line para essa finalidade.

> **Note**
>
> Se quiser importar suas próprias chaves de teste, confira o projeto de amostra disponível no Github. Para obter instruções sobre como importar/exportar chaves de outras plataformas, consulte o guia do usuário dessas plataformas.

Troca de chaves de trabalho (WK)

AWS A criptografia de pagamento usa a norma relevante do setor (ANSI X9.24 TR 31-2018) para trocar chaves de trabalho. O TR-31 assume que uma KEK foi trocada anteriormente. Isso é consistente com o requisito do PCI PIN de vincular criptograficamente o material da chave ao seu tipo e uso de chaves em todos os momentos. As chaves de trabalho têm vários nomes, incluindo chaves de trabalho do adquirente, chaves de trabalho do emissor, BDK, IPEK, etc.

Tópicos

Importar chaves

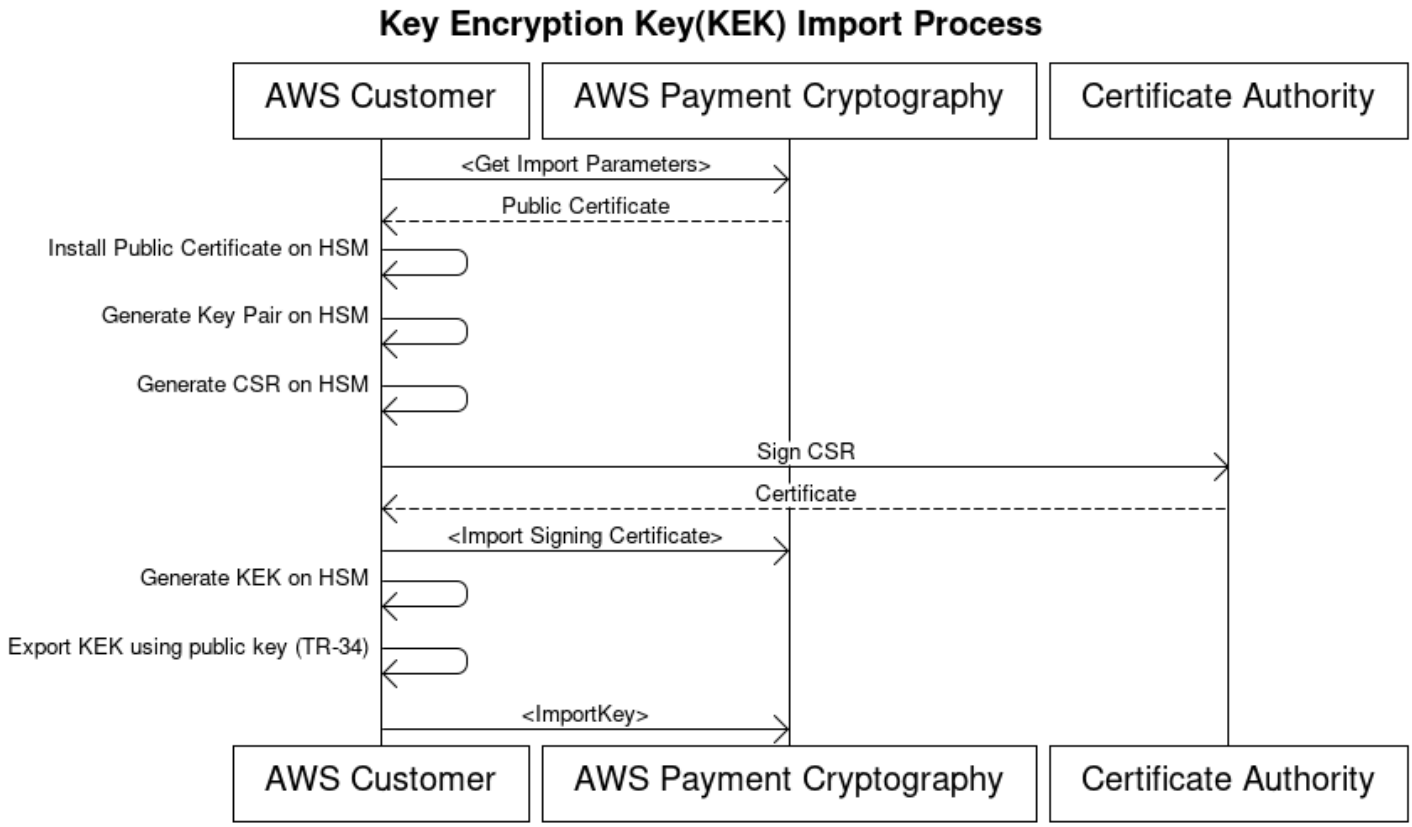
> **Important**
>
> Os exemplos podem exigir a versão mais recente da AWS CLI V2. Antes de começar, verifique se você atualizou para a versão mais recente.

Tópicos

Importar chaves simétricas

Tópicos

Importar chaves usando técnicas assimétricas (TR-34)



Visão geral: o TR-34 utiliza criptografia assimétrica RSA para criptografar chaves simétricas para troca, além de garantir a origem dos dados (assinatura). Isso garante a confidencialidade (criptografia) e a integridade (assinatura) da chave encapsulada.

Se quiser importar suas próprias chaves, confira o projeto de amostra disponível no Github. Para obter instruções sobre como importar/exportar chaves de outras plataformas, consulte o guia do usuário dessas plataformas.

1. Chamar o comando de inicialização de importação

   Chame `get-parameters-for-import` para inicializar o processo de importação. Essa API gerará um par de chaves para fins de importação de chaves, assinará a chave e retornará o certificado e a raiz do certificado. Em última análise, a chave a ser exportada deve ser criptografada usando essa chave. Na terminologia do TR-34, isso é conhecido como Certificado de KRD. Observe que esses certificados têm vida curta e se destinam apenas a essa finalidade.

2. Instalar o certificado público no sistema de origem da chave

   Com muitos HSMs, pode ser necessário instalar/carregar/confiar no certificado público gerado na etapa 1 para exportar chaves com o uso dele.

3. Gere a chave pública e forneça a raiz do certificado para criptografia AWS de pagamento

   Para garantir a integridade da carga transmitida, ela é assinada pela parte remetente (conhecida como host de distribuição de chaves ou KDH). A parte remetente desejará gerar uma chave pública para essa finalidade e, em seguida, criar um certificado de chave pública (X509) que possa ser devolvido à AWS Payment Cryptography. AWS Private CA é uma opção para gerar certificados, mas não há restrições quanto à autoridade de certificação usada.

   Depois de ter o certificado, você desejará carregar o certificado raiz na Criptografia AWS de Pagamento usando o `importKey` comando e KeyMaterialType de `ROOT_PUBLIC_KEY_CERTIFICATE` e KeyUsageType de`TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`.

4. Exportar chave do sistema de origem

   Muitos HSMs e sistemas relacionados oferecem suporte à capacidade de exportar chaves usando a norma TR-34. Você deve especificar a chave pública da etapa 1 como o certificado KRD (criptografia) e a chave da etapa 3 como o certificado KDH (assinatura). Para importar para a criptografia de AWS pagamento, você deve especificar o formato de duas passagens TR-34.2012, não CMS, que também pode ser chamado de formato TR-34 Diebold.

5. Chamar chave de importação

   Como última etapa, você chamará a API importKey com um KeyMaterialType de`TR34_KEY_BLOCK`. O `certificate-authority-public-key-identifier` será o keyARN da CA raiz importada na etapa 3, `key-material` será o material de chave encapsulado da etapa 4 e `signing-key-certificate` será o certificado de entidade final da etapa 3. Você também precisará fornecer o token de importação da etapa 1.

6. Use a chave importada para operações criptográficas ou importação subsequente

   Se o importado KeyUsage foi TR31_K0_KEY_ENCRYPTION_KEY, essa chave poderá ser usada para importações de chaves subsequentes usando TR-31. Se o tipo de chave for qualquer outro tipo (como TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY), a chave poderá ser usada diretamente para operações criptográficas.

Importe chaves usando técnicas assimétricas (RSA Unwrap)

Visão geral: a criptografia AWS de pagamento suporta RSA wrap/unwrap para troca de chaves quando o TR-34 não é viável. Semelhante ao TR-34, essa técnica utiliza criptografia assimétrica RSA para criptografar chaves simétricas para troca. No entanto, diferentemente do TR-34, esse método não tem a carga assinada pela parte remetente. Além disso, essa técnica de encapsulamento RSA não mantém a integridade dos metadados da chave durante a transferência por não incluir blocos de chaves.

> **Note**
>
> O RSA wrap pode ser usado para importar ou exportar chaves TDES e AES-128.

1. Chamar o comando de inicialização de importação

   Ligue `get-parameters-for-import` para inicializar o processo de importação com um tipo de material de chave de KEY_CRYPTOGRAM. WrappingKeyAlgorithm pode ser RSA_2048 ao trocar chaves TDES. O RSA_3072 ou o RSA_4096 podem ser usados ao trocar chaves TDES ou AES-128. Essa API gerará um par de chaves para fins de importação de chaves, assinará a chave usando uma raiz de certificado e retornará tanto o certificado quanto a raiz do certificado. Em última análise, a chave a ser exportada deve ser criptografada usando essa chave. Observe que esses certificados têm vida curta e se destinam apenas a essa finalidade.

```
$ 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 o certificado público no sistema de origem da chave

   Com muitos HSMs, talvez seja necessário instalar/carregar/confiar no certificado público (e/ou em sua raiz) gerado na etapa 1 para exportar chaves usando-o.

3. Exportar chave do sistema de origem

   Muitos HSMs e sistemas relacionados oferecem suporte à capacidade de exportar chaves usando o RSA wrap. Você desejará especificar a chave pública da etapa 1 como o certificado () (de criptografiaWrappingKeyCertificate). Se você precisar da cadeia de confiança, ela está contida no campo de resposta WrappingKeyCertificateChain na etapa #1. Ao exportar a chave do seu HSM, você deve especificar o formato como RSA, Padding Mode = PKCS #1 v2.2 OAEP (com SHA 256 ou SHA 512).

4. Chamar chave de importação

   Como última etapa, você chamará a API importKey com um KeyMaterialType de`KeyMaterial`. Você precisará do token de importação da etapa 1 e do `key-material` (material da chave embrulhada) da etapa 3. Você precisará fornecer os principais parâmetros (como o uso da chave), pois o RSA wrap não utiliza blocos de chaves.

```
$ 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"
                    }
                }
            }
```

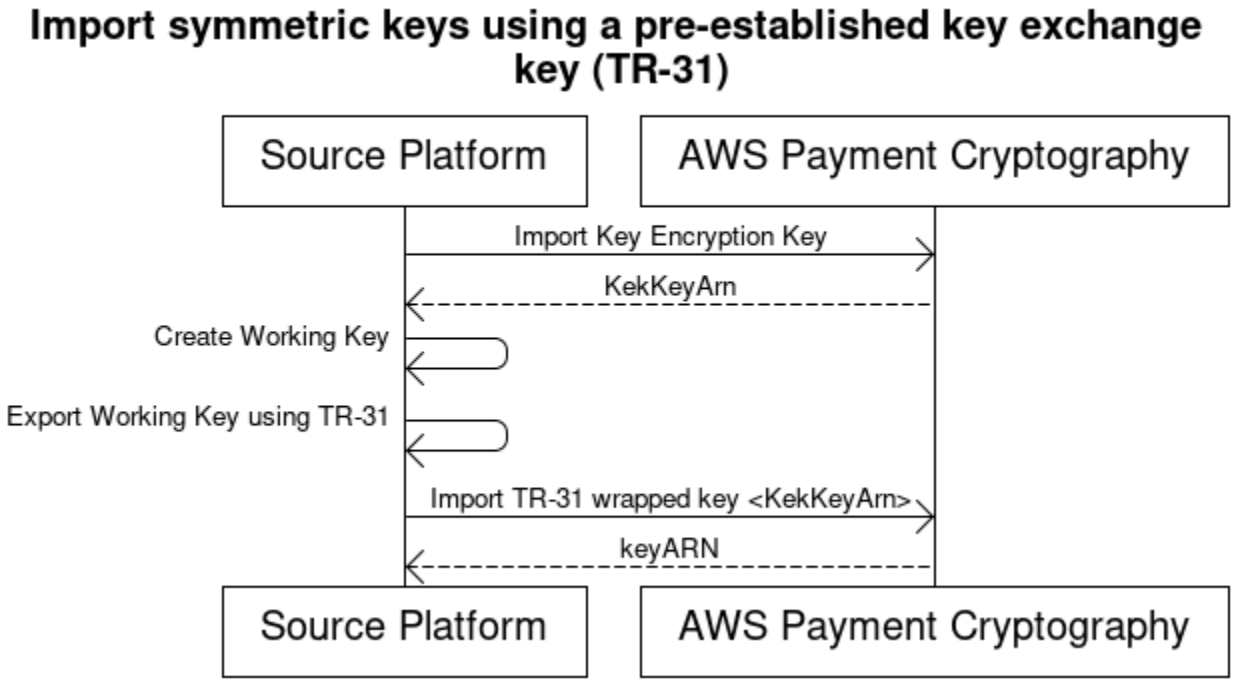
```
$ aws payment-cryptography import-key --cli-input-json file://import-key-
cryptogram.json
```

```
{
      "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. Use a chave importada para operações criptográficas ou importação subsequente

   Se o importado KeyUsage foi TR31_K0_KEY_ENCRYPTION_KEY, essa chave poderá ser usada para importações de chaves subsequentes usando TR-31. Se o tipo de chave for qualquer outro tipo (como TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY), a chave poderá ser usada diretamente para operações criptográficas.

Importe chaves simétricas usando uma chave de troca de chaves preestabelecida (TR-31)



Quando os parceiros estão trocando várias chaves (ou para oferecer suporte à rotação de chaves), é comum trocar primeiro uma chave de criptografia de chave inicial (KEK) usando técnicas como componentes de chave em papel ou, no caso de criptografia de AWS pagamento, usando TR-34.

Depois que uma KEK é estabelecida, você pode usar essa chave para transportar as chaves subsequentes (incluindo outras KEKs). AWS A criptografia de pagamento suporta esse tipo de troca de chaves usando o ANSI TR-31, que é amplamente usado e amplamente suportado pelos fornecedores de HSM.

1. Importar chave de criptografia de chave (KEK)

   Supõe-se que você já tenha importado sua KEK e tenha o keyARN (ou keyAlias) disponível para você.

2. Criar uma chave na plataforma de origem

   Se a chave ainda não existir, crie-a na plataforma de origem. Por outro lado, você pode criar a chave no AWS Payment Cryptography e usar o comando `export` em vez disso.

3. Exportar chave da plataforma de origem

   Ao exportar, especifique o formato de exportação como TR-31. A plataforma de origem também solicitará a exportação da chave e a chave de criptografia a ser usada.

4. Importar para criptografia AWS de pagamento

   Ao chamar o comando importKey, WrappingKeyIdentifier deve ser o keyArn (ou alias) da chave de criptografia da chave e a saída da WrappedKeyBlock plataforma de origem.

   Example

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

 WrappedKeyBlock="D0112B0AX00E00002E0A3D58252CB67564853373D1EBCC1E23B2ADE7B15E967CC27B85D599
```

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

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

Importar chaves assimétricas (RSA)

Importar chaves públicas RSA

AWS A criptografia de pagamento suporta a importação de chaves RSA públicas na forma de certificados X.509. Para importar um certificado, você precisará primeiro importar seu certificado raiz. Todos os certificados devem estar válidos no momento da importação. O certificado deve estar no formato PEM e ser codificado em base64.

1. Importar para o certificado raiz para criptografia AWS de pagamento

   Example

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

 "PublicKeyCertificate":"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURKVENDQWcyZ0F3SUJBZ0lCWkR
```

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

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

2. Importar certificado de chave pública para criptografia AWS de pagamento

   Agora, você pode importar uma chave pública. Existem duas opções para importar chaves públicas. `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE` pode ser usado se o objetivo da chave for verificar assinaturas (por exemplo, ao importar usando TR-34). `TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION` pode ser usado ao criptografar dados destinados ao uso com outro sistema.

   Example

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

```
{
```

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

Exportar chaves

Tópicos

- Exportar chaves simétricas
- Exportação de chaves assimétricas (RSA)

Exportar chaves simétricas

> **Important**
>
> Os exemplos podem exigir a versão mais recente da AWS CLI V2. Antes de começar, verifique se você atualizou para a versão mais recente.

Tópicos

Exportar chaves usando técnicas assimétricas (TR-34)

Visão geral: o TR-34 utiliza criptografia assimétrica RSA para criptografar chaves simétricas para troca, além de garantir a origem dos dados (assinatura). Isso garante a confidencialidade (criptografia) e a integridade (assinatura) da chave encapsulada. Ao exportar, a criptografia AWS de pagamento se torna o host de distribuição de chaves (KDH) e o sistema de destino se torna o dispositivo de recebimento de chaves (KRD).

1. Chamar o comando de inicialização de exportação

   Chame `get-parameters-for-export` para inicializar o processo de exportação. Essa API gerará um par de chaves para fins de exportação de chaves, assinará a chave e retornará o certificado e a raiz do certificado. Em última análise, a chave privada gerada por este comando é usada para assinar a carga útil de exportação. Na terminologia TR-34, isso é conhecido como certificado de assinatura KDH. Observe que esses certificados têm vida curta e se destinam apenas a essa finalidade. O parâmetro ParametersValidUntilTimestamp especifica sua duração.

   NOTA: todos os certificados são retornados em um formato codificado em base64

   Example

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

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

```
}
```

2. Importar certificado AWS de criptografia de pagamento para o sistema de recebimento

   Importe a cadeia de certificados fornecida na etapa 1 para o sistema de recebimento, conforme necessário.

3. Gere um key pair, crie um certificado público e forneça a raiz do certificado para AWS Payment Cryptography

   Para garantir a confidencialidade da carga transmitida, ela é criptografada pela parte remetente (conhecida como host de distribuição de chaves ou KDH). A parte receptora (normalmente seu HSM ou o HSM de seus parceiros) desejará gerar uma chave pública para essa finalidade e, em seguida, criar um certificado de chave pública (x.509) que possa ser devolvido à Criptografia de Pagamento. AWS AWS Private CA é uma opção para gerar certificados, mas não há restrições quanto à autoridade de certificação usada.

   Depois de ter o certificado, você desejará carregar o certificado raiz na Criptografia AWS de Pagamento usando o `ImportKey` comando e KeyMaterialType de `ROOT_PUBLIC_KEY_CERTIFICATE` e KeyUsageType de`TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`.

   O KeyUsageType deste certificado é TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE porque é a chave raiz e é usada para assinar o certificado folha. Os certificados Leaf para importação/exportação não são importados para a criptografia AWS de pagamento, mas são passados em linha.

   > **Note**
   >
   > Se o certificado raiz tiver sido importado anteriormente, essa etapa poderá ser ignorada.

4. Chamar a chave de exportação

   Como última etapa, você chamará a ExportKey API com um KeyMaterialType de`TR34_KEY_BLOCK`. O `certificate-authority-public-key-identifier` será o keyARN da importação raiz da CA na etapa 3, `WrappingKeyCertificate` será o certificado de entidade final da etapa 3 e `export-key-identifier` será o keyARN (ou alias) a ser exportado. Você também precisará fornecer o token de exportação da etapa 1.

Exporte chaves usando técnicas assimétricas (RSA Wrap)

Visão geral: a criptografia AWS de pagamento suporta o empacotamento e desempacotamento RSA para troca de chaves quando o TR-34 não é uma opção disponível pela contraparte. Semelhante ao TR-34, essa técnica utiliza criptografia assimétrica RSA para criptografar chaves simétricas para troca. No entanto, diferentemente do TR-34, esse método não tem a carga assinada pela parte remetente. Além disso, essa técnica de encapsulamento de RSA não inclui blocos de chaves que são usados para manter a integridade dos metadados da chave durante o transporte.

> **Note**
>
> O RSA wrap pode ser usado para exportar chaves TDES e AES-128.

1. Gere uma chave RSA e um certificado no sistema receptor

   Crie (ou identifique) uma chave RSA que será usada para receber a chave encapsulada. AWS A criptografia de pagamento espera chaves no formato de certificado X.509. O certificado deve ser assinado por um certificado raiz que seja importado (ou possa ser importado) para a criptografia AWS de pagamento.
2. Instale o certificado público raiz na criptografia AWS de pagamento

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

```
    {
        "Key": {
        "CreateTimestamp": "2023-09-14T10:50:32.365000-07:00",
        "Enabled": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
nsq2i3mbg6sn775f",
        "KeyAttributes": {
        "KeyAlgorithm": "RSA_4096",
        "KeyClass": "PUBLIC_KEY",
        "KeyModesOfUse": {
        "Decrypt": false,
        "DeriveKey": false,
        "Encrypt": false,
```

```
        "Generate": false,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": false,
        "Verify": true,
        "Wrap": false
        },
        "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"
        },
        "KeyOrigin": "EXTERNAL",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2023-09-14T10:50:32.365000-07:00“
        }
       }
```

3. Chave de exportação de chamadas

Em seguida, você deve instruir a AWS Payment Cryptography a exportar sua chave usando seu certificado de folha. Você especificará o ARN para o certificado raiz importado anteriormente, o certificado folha a ser usado para exportação e a chave simétrica a ser exportada. A saída será uma versão binária encapsulada (criptografada) com codificação hexadecimal da sua chave simétrica.

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

```
  {
      "ExportKeyIdentifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/tqv5yij6wtxx64pi",
      "KeyMaterial": {
          "KeyCryptogram": {
              "CertificateAuthorityPublicKeyIdentifier": "arn:aws:payment-
cryptography:us-east-2:111122223333:key/zabouwe3574jysdl",
              "WrappingKeyCertificate": "LS0tLS1CRUdJTiBD...",
              "WrappingSpec": "RSA_OAEP_SHA_256"
          }
      }
  }
```

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

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

4. Importar chave para o sistema receptor

   Muitos HSMs e sistemas relacionados oferecem suporte à capacidade de importar chaves usando o RSA unwrap (incluindo criptografia de AWS pagamento). Para fazer isso, especifique a chave pública da etapa 1 como o certificado (de criptografia) e o formato deve ser especificado como RSA, Padding Mode = PKCS #1 v2.2 OAEP (com SHA 256). A terminologia exata pode variar de acordo com o HSM.

   > **Note**
   >
   > AWS A criptografia de pagamento gera a chave encapsulada em HexBinary. Talvez seja necessário converter o formato antes de importar se o sistema exigir uma representação binária diferente, como base64.

Exportar chaves simétricas usando uma chave de troca de chaves preestabelecida (TR-31)

Quando os parceiros estão trocando várias chaves (ou para oferecer suporte à rotação de chaves), é comum trocar primeiro uma chave de criptografia de chave inicial (KEK) usando técnicas como componentes de chave em papel ou, no caso de criptografia de AWS pagamento, usando TR-34. Depois que uma KEK é estabelecida, você pode usar essa chave para transportar as chaves subsequentes (incluindo outras KEK). AWS A criptografia de pagamento suporta esse tipo de troca de chaves usando o ANSI TR-31, que é amplamente usado e amplamente suportado pelos fornecedores de HSM.

1. Troca de chave de criptografia de chave (KEK)

   Supõe-se que você já tenha trocado sua KEK e tenha o KeyARN (ou keyAlias) disponível para você.

2. Crie uma chave na criptografia AWS de pagamento

   Se a chave ainda não existir, crie a chave. Por outro lado, você pode criar a chave no outro sistema e usar o comando importar em vez disso.

3. Exportar chave da criptografia AWS de pagamento

   Na exportação, o formato será TR-31. Ao chamar a API, você especificará a chave a ser exportada e a chave de encapsulamento a ser usada.

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

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

4. Importe para o seu sistema

   Você ou seu parceiro usarão a implementação da chave de importação em seu sistema para importar a chave.

Exportar chaves iniciais DUKPT (IPEK/IK)

Ao usar o DUKPT, uma única chave de derivação de base (BDK) pode ser gerada para uma frota de terminais. Os terminais, no entanto, nunca têm acesso ao BDK original, mas cada um é injetado com uma chave de terminal inicial exclusiva conhecida como IPEK ou Chave Inicial (IK). Cada IPEK é uma chave derivada do BDK e deve ser exclusiva por terminal, mas é derivada do BDK original. Os dados de derivação para esse cálculo são conhecidos como Número de Série da Chave (KSN). Por X9.24, para TDES, o KSN de 10 bytes normalmente consiste em 24 bits para o ID do conjunto de chaves, 19 bits para o ID do terminal e 21 bits para o contador de transações. Para o AES, o KSN de 12 bytes normalmente consiste em 32 bits para o ID do BDK, 32 bits para o identificador de derivação (ID) e 32 bits para o contador da transação.

AWS A criptografia de pagamento fornece um mecanismo para gerar e exportar essas chaves iniciais. Depois de geradas, essas chaves podem ser exportadas usando os métodos de encapsulamento TR-31, TR-34 e RSA. As chaves IPEK não são mantidas e não podem ser usadas para operações subsequentes em criptografia de pagamento AWS

AWS A criptografia de pagamento não impõe a divisão entre as duas primeiras partes do KSN. Se quiser armazenar o identificador de derivação junto com o BDK, você pode usar o recurso de AWS tags para essa finalidade.

> **Note**
>
> A parte do contador do KSN (32 bits para AES DUKPT) não é usada para derivação IPEK/IK. Portanto, uma entrada de 12345678901234560001 e 12345678901234569999 produzirá o mesmo 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"
    }
}
```

Exportação de chaves assimétricas (RSA)

Chame `get-public-key-certificate` para exportar uma chave pública em formato de certificado. Essa API exportará o certificado e seu certificado raiz codificado no formato base64.

OBSERVAÇÃO: essa API não é idempotente. As chamadas subsequentes podem resultar em certificados diferentes, mesmo que a chave subjacente seja a mesma.

Example

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

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

Usar aliases

Um alias é um nome amigável para uma chave de criptografia AWS de pagamento. Por exemplo, um alias permite fazer referência a uma chave como `alias/test-key` em vez de `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h`.

É possível usar um alias para identificar uma chave na maioria das operações de gerenciamento de chaves (ambiente de gerenciamento) e em operações criptográficas (plano de dados).

Você também pode permitir e negar o acesso à chave AWS de criptografia de pagamento com base em seus aliases sem editar políticas ou gerenciar subsídios. Esse atributo faz parte do suporte do serviço para controle de acesso por atributo (ABAC).

Grande parte do poder dos aliases vem da sua capacidade de alterar a chave associada a um alias a qualquer momento. Aliases podem tornar seu código mais fácil de escrever e manter. Por exemplo, suponha que você use um alias para se referir a uma chave AWS de criptografia de pagamento específica e queira alterar a chave de criptografia AWS de pagamento. Nesse caso, basta associar o alias a uma chave diferente. Você não precisa alterar o código ou a configuração do aplicativo.

Aliases também facilitam a reutilização do mesmo código em Regiões da AWS diferentes. Crie aliases com o mesmo nome em várias regiões e associe cada alias a uma chave de criptografia AWS de pagamento em sua região. Quando o código é executado em cada região, o alias se refere à chave de criptografia AWS de pagamento associada nessa região.

Você pode criar um alias para uma chave AWS de criptografia de pagamento usando a `CreateAlias` API.

A API AWS Payment Cryptography fornece controle total dos aliases em cada conta e região. A API inclui operações para criar um alias (CreateAlias), visualizar nomes de alias e o keyArn vinculado (list-aliases), alterar a chave de criptografia de AWS pagamento associada a um alias (update-alias) e excluir um alias (delete-alias).

Tópicos

Sobre aliases

Saiba como os aliases funcionam na criptografia AWS de pagamento.

Um alias é um recurso independente AWS

Um alias não é propriedade de uma chave de criptografia AWS de pagamento. As ações executadas no alias não afetam a chave associada. Você pode criar um alias para uma chave de criptografia AWS de pagamento e, em seguida, atualizar o alias para que seja associado a uma chave de criptografia de AWS pagamento diferente. Você pode até mesmo excluir o alias sem qualquer efeito na chave de criptografia AWS de pagamento associada. Se você excluir uma chave do AWS Payment Cryptography, todos os aliases associados a essa chave deixarão de ser atribuídos.

Se você especificar um alias como recurso em uma política do IAM, a política se referirá ao alias, não à chave de criptografia de AWS pagamento associada.

Cada alias tem um nome fácil de usar

Ao criar um alias, você especifica o nome do alias prefixado por `alias/`. Por exemplo, `alias/test_1234`

Cada alias é associado a uma chave AWS de criptografia de pagamento por vez

O alias e sua chave AWS de criptografia de pagamento devem estar na mesma conta e região.

Uma chave AWS de criptografia de pagamento pode ser associada a mais de um alias simultaneamente, mas cada alias só pode ser mapeado para uma única chave

Por exemplo, esta saída de `list-aliases` mostra que o alias `alias/sampleAlias1` está associado exatamente a uma chave de AWS Payment Cryptography de destino, que é representada pela propriedade `KeyArn`.

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

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

Vários aliases podem ser associados à mesma chave de criptografia AWS de pagamento

Por exemplo, você pode associar os aliases `alias/sampleAlias1;` e `alias/sampleAlias2` à mesma chave.

```
$ 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"
            }
        ]
```

```
}
```

Um alias deve ser exclusivo para uma determinada conta e região

Por exemplo, é possível ter apenas um alias `alias/sampleAlias1` em cada conta e região. Os aliases diferenciam maiúsculas de minúsculas, mas não recomendamos usar aliases que diferem apenas no tamanho das letras, pois podem estar propensos a erros. Não é possível alterar um nome de alias. No entanto, você pode excluir o alias e criar um novo com o nome desejado.

É possível criar um alias com o mesmo nome em regiões diferentes

Por exemplo, você pode ter um alias `alias/sampleAlias2` no Leste dos EUA (Norte da Virgínia) e um alias `alias/sampleAlias2` no Oeste dos EUA (Oregon). Cada alias seria associado a uma chave de criptografia AWS de pagamento em sua região. Se o seu código se referir a um nome de alias como `alias/finance-key`, você poderá executá-lo em várias regiões. Em cada região, ele usa um alias/sampleAlias2 diferente. Para obter detalhes, consulte Usar aliases em suas aplicações.

Você pode alterar a chave AWS de criptografia de pagamento associada a um alias

Você pode usar a `UpdateAlias` operação para associar um alias a uma chave de criptografia AWS de pagamento diferente. Por exemplo, se o `alias/sampleAlias2` alias estiver associado à chave de criptografia de `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h` AWS pagamento, você poderá atualizá-lo para que fique associado à `arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi` chave.

> **Warning**
>
> AWS A criptografia de pagamento não valida que as chaves antigas e novas tenham todos os mesmos atributos, como o uso da chave. A atualização com um tipo de chave diferente pode resultar em problemas em seu aplicativo.

Algumas chaves não têm aliases

Um alias é um atributo opcional e nem todas as chaves terão aliases, a menos que você opte por operar seu ambiente dessa maneira. As chaves podem ser associadas a aliases usando o comando `create-alias`. Além disso, você pode usar a operação update-alias para alterar a chave de AWS Payment Cryptography associada a um alias e a operação delete-alias para

excluir um alias. Como resultado, algumas chaves AWS de criptografia de pagamento podem ter vários aliases e outras podem não ter nenhum.

Mapear uma chave para um alias

É possível mapear uma chave (representada por um ARN) para um ou mais aliases usando o comando `create-alias`. Esse comando não é idempotente. Para atualizar um alias, use o 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 aliases em suas aplicações

Você pode usar um alias para representar uma chave AWS de criptografia de pagamento no código do aplicativo. O `key-identifier` parâmetro nas operações de dados AWS de criptografia de pagamento, bem como em outras operações, como chaves de lista, aceita um nome de alias ou 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}
```

Ao usar um ARN de alias, lembre-se de que o mapeamento de alias para AWS uma chave de criptografia de pagamento é definido na conta que possui AWS a chave de criptografia de pagamento e pode ser diferente em cada região.

Um dos usos mais poderosos dos aliases é em aplicações executadas em várias Regiões da AWS.

Você pode criar uma versão diferente do seu aplicativo em cada região ou usar um dicionário, uma configuração ou um extrato de switch para selecionar a chave de criptografia de AWS pagamento

certa para cada região. Mas pode ser mais fácil criar um alias com o mesmo nome em cada região. Lembre-se de que o nome do alias diferencia maiúsculas de minúsculas.

APIs relacionadas

Tags

As tags são pares de chaves e valores que atuam como metadados para organizar suas chaves AWS de criptografia de pagamento. Elas podem ser usadas para identificar chaves de forma flexível ou agrupar uma ou mais chaves.

Obter chaves

Uma chave AWS de criptografia de pagamento representa uma única unidade de material criptográfico e só pode ser usada para operações criptográficas desse serviço. A GetKeys API usa a KeyIdentifier como entrada e retorna os atributos imutáveis e mutáveis da chave, mas não contém nenhum material criptográfico.

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"
  }
}
```

Obter a chave/certificado público associado a um par de chaves

Obter chave pública/certificado retorna a chave pública indicada pela `KeyArn`. Essa pode ser a parte da chave pública de um par de chaves gerado na criptografia AWS de pagamento ou uma chave pública importada anteriormente. O caso de uso mais comum é fornecer a chave pública a um serviço externo que fará a criptografia dos dados. Esses dados podem então ser passados para um aplicativo usando a criptografia de AWS pagamento e os dados podem ser descriptografados usando a chave privada protegida na criptografia de pagamento. AWS

O serviço retorna as chaves públicas como um certificado público. O resultado da API contém a CA e o certificado de chave pública. Ambos os elementos de dados são codificados em base64.

> **Note**
>
> O certificado público retornado deve durar pouco e não ser idempotente. É possível receber um certificado diferente em cada chamada de API, mesmo que a chave pública permaneça inalterada.

Example

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


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


Marcar chaves com tags

Na Criptografia de AWS pagamento, você pode adicionar tags a uma chave de criptografia de AWS pagamento ao criar uma chave e marcar ou desmarcar as chaves existentes, a menos que estejam pendentes de exclusão. Etiquetas são opcionais, mas podem ser bastante úteis.

Para obter informações gerais sobre tags, incluindo melhores práticas, estratégias de marcação e o formato e a sintaxe das tags, consulte AWS Recursos de marcação no. Referência geral da Amazon Web Services

Tópicos

- Sobre tags na criptografia AWS de pagamento
- Visualizar tags de chave no console
- Gerenciar tags de chave com operações de API
- Controlar o acesso às tags
- Usar tags para controlar o acesso a chaves

Sobre tags na criptografia AWS de pagamento

Uma tag é um rótulo de metadados opcional que você pode atribuir (ou AWS atribuir) a um AWS recurso. Cada tag consiste em uma chave de tag e um valor de tag, sendo ambos strings que diferenciam maiúsculas de minúsculas. O valor da tag pode ser uma string vazia (nula). Cada tag em um recurso precisa ter uma chave de tag diferente, mas você pode adicionar a mesma tag a vários AWS recursos. Cada recurso pode ter até 50 tags criadas pelo usuário.

Não inclua informações confidenciais ou sigilosas na chave ou no valor da tag. As tags podem ser acessadas por muitos Serviços da AWS, incluindo o faturamento.

Na criptografia AWS de pagamento, você pode adicionar tags a uma chave ao criar a chave e marcar ou desmarcar as chaves existentes, a menos que estejam pendentes de exclusão. Não é possível marcar aliases com tags. Etiquetas são opcionais, mas podem ser bastante úteis.

Por exemplo, você pode adicionar uma `"Project"="Alpha"` tag a todas as chaves de criptografia AWS de pagamento e buckets do Amazon S3 que você usa para o projeto Alpha. Outro exemplo é adicionar uma tag `"BIN"="20130622"` a todas as chaves associadas a um número de identificação bancária (BIN) específico.

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

Para obter informações gerais sobre tags, incluindo o formato e a sintaxe, consulte AWS Recursos de marcação no. Referência geral da Amazon Web Services

As tags ajudam a:

- Identifique e organize seus AWS recursos. Muitos AWS serviços oferecem suporte à marcação, então você pode atribuir a mesma tag a recursos de serviços diferentes para indicar que os recursos estão relacionados. Por exemplo, você pode atribuir a mesma tag a uma chave de

criptografia AWS de pagamento e a um volume ou segredo do Amazon Elastic Block Store (Amazon EBS). AWS Secrets Manager Você também pode usar tags para identificar chaves para automação.

- Acompanhe seus AWS custos. Quando você adiciona tags aos seus AWS recursos, AWS gera um relatório de alocação de custos com uso e custos agregados por tags. Você pode usar esse recurso para rastrear os custos AWS de criptografia de pagamento de um projeto, aplicativo ou centro de custos.

  Para obter mais informações sobre como usar etiquetas para alocação de custos, consulte Usar etiquetas de alocação de custos no Manual do usuário do AWS Billing . Para obter informações sobre as regras para chaves e valores de etiquetas, consulte Restrições de etiquetas definidas pelo usuário, no Manual do usuário do AWS Billing .
- Controle o acesso aos seus AWS recursos. Permitir e negar o acesso às chaves com base em suas tags faz parte do suporte à criptografia de AWS pagamento para controle de acesso baseado em atributos (ABAC). Para obter informações sobre como controlar o acesso à AWS Payment Cryptography com base em suas tags, consulte Autorização baseada em tags do AWS Payment Cryptography. Para obter mais informações gerais sobre o uso de tags para controlar o acesso aos AWS recursos, consulte Como controlar o acesso aos AWS recursos usando tags de recursos no Guia do usuário do IAM.

AWS A criptografia de pagamento grava uma entrada em seu AWS CloudTrail registro quando você usa as ListTagsForResource operações TagResource, UntagResource, ou.

Visualizar tags de chave no console

Para visualizar tags no console, é necessário ter permissão para marcar com tags na chave de uma política do IAM que inclua a chave. Essas permissões são necessárias além das requeridas para visualizar chaves no console.

Gerenciar tags de chave com operações de API

É possível usar a API de AWS Payment Cryptography para adicionar, excluir e listar tags para as chaves que você gerencia. Estes exemplos usam a AWS Command Line Interface (AWS CLI), mas você pode usar qualquer linguagem de programação compatível. Você não pode marcar Chaves gerenciadas pela AWS.

Para adicionar, editar, visualizar e excluir tags de uma chave, é necessário ter as permissões apropriadas. Para obter detalhes, consulte Controlar o acesso às tags.

Tópicos

CreateKey: Adicionar tags a uma nova chave

Você pode adicionar tags ao criar uma chave. Para especificar as tags, use o `Tags` parâmetro da CreateKeyoperação.

Para adicionar tags ao criar uma chave, o chamador deve ter a permissão `payment-cryptography:TagResource` em uma política do IAM. Essa permissão deve abranger, no mínimo, todas as chaves na conta e na região. Para obter detalhes, consulte Controlar o acesso às tags.

O valor do parâmetro `Tags` de `CreateKey` é uma coleção de pares de chave de etiqueta e valor de etiqueta que faz distinção entre maiúsculas e minúsculas. Cada tag em uma chave deve ter um nome de tag diferente. O valor da tag pode ser uma string nula ou vazia.

Por exemplo, o AWS CLI comando a seguir cria uma chave de criptografia simétrica com uma `Project:Alpha` tag. Ao especificar mais de um par de chave-valor, use um espaço 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"}]'
```

Quando esse comando é bem-sucedido, ele retorna um objeto `Key` com informações sobre a nova chave. No entanto, o `Key` não inclui tags. Para obter as tags, use a ListResourceTagsoperação.

TagResource: Adicionar ou alterar tags para uma chave

A TagResourceoperação adiciona uma ou mais tags a uma chave. Não é possível usar essa operação para adicionar ou editar etiquetas em uma Conta da AWS diferente.

Para adicionar uma tag, especifique uma nova chave e um valor de tag. Para editar uma tag, especifique uma chave de tag existente e um novo valor de tag. Cada tag em uma chave deve ter uma chave de tag diferente. O valor da tag pode ser uma string nula ou vazia.

Por exemplo, o comando a seguir adiciona as tags **`UseCase`** e **`BIN`** a uma chave demonstrativa.

```
$ 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"}]'
```

Quando esse comando for executado com êxito, ele não retornará nenhuma saída. Para visualizar as tags em uma chave, use a ListResourceTagsoperação.

Você também pode usar TagResource para alterar o valor de uma tag existente. Para substituir um valor de tag, especifique a mesma chave de tag com um valor diferente. Tags não listadas em um comando de modificação não são alteradas nem removidas.

Por exemplo, esse comando altera o valor da tag `Project` de `Alpha` para `Noe`.

O comando retornará http/200 sem conteúdo. Para ver suas alterações, use `ListTagsForResource`

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

ListResourceTags: Obtenha as etiquetas para uma chave

A ListResourceTagsoperação obtém as etiquetas de uma chave. O parâmetro `ResourceArn` (keyARN ou keyAlias) é obrigatório. Essa operação não pode ser usada para visualizar as tags nas chaves em uma Conta da AWS diferente.

Por exemplo, o comando a seguir obtém as tags para uma chave demonstrativa.

```
$ 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: Excluir tags de uma chave

A UntagResourceoperação exclui as tags de uma chave. Para identificar as etiquetas a serem excluídas, especifique as chaves de etiqueta. Essa operação não pode ser usada para excluir tags de chaves em uma Conta da AWS diferente.

Quando é bem-sucedida, a operação `UntagResource` não retorna nenhuma saída. Além disso, se a chave da tag especificada não for encontrada na chave, ela não lançará uma exceção nem retornará uma resposta. Para confirmar se a operação funcionou, use a ListResourceTagsoperação.

Por exemplo, esse comando exclui a tag **`Purpose`** e seu valor com base na chave especificada.

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

Controlar o acesso às tags

Para adicionar, visualizar e excluir tags usando a API, as entidades principais precisam de permissões de marcação nas políticas do IAM.

Você também pode limitar essas permissões usando chaves de condição AWS globais para tags. Na criptografia AWS de pagamento, essas condições podem controlar o acesso às operações de marcação, como e. TagResourceUntagResource

Para mais informações e exemplos de políticas, consulte Controlar o acesso baseado em chaves de etiqueta, no Guia do Usuário do IAM.

As permissões para criar e gerenciar aliases funcionam como a seguir.

criptografia de pagamento: TagResource

Permite que as entidades principais adicionem ou editem etiquetas. Para adicionar tags ao criar uma chave, a entidade principal deve ter permissão em uma política do IAM que não esteja restrita a chaves específicas.

criptografia de pagamento: ListTagsForResource

Permite que as entidades principais visualizem tags em chaves.

criptografia de pagamento: UntagResource

Permite que as entidades principais excluam tags de chaves.

Permissões de etiquetas em políticas

Você pode fornecer permissões de marcação em uma política de chaves ou política do IAM. O seguinte exemplo de política de chaves concede permissão de marcação a usuários selecionados na chave. Ele concede a todos os usuários que podem assumir os exemplos de funções Administrador ou Desenvolvedor permissão para visualizar etiquetas.

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

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

Para conceder permissão de marcação de entidades principais em várias chaves, é possível usar uma política do IAM. Para que essa política seja eficiente, a política de chaves de cada chave deve permitir que a conta utilize políticas do IAM para controlar o acesso à chave.

Por exemplo, a seguinte política do IAM permite que as entidades principais criem chaves. Ela também permite que eles criem e gerenciem tags em todas as chaves na conta especificada. Essa combinação permite que os diretores usem o parâmetro tags da CreateKeyoperação para adicionar tags a uma chave enquanto a criam.

```
{
  "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 permissões de etiquetas

É possível limitar permissões de marcação usando condições de política. As seguintes condições de política podem ser aplicadas às permissões `payment-cryptography:TagResource` e `payment-cryptography:UntagResource`. Por exemplo, você pode usar a condição `aws:RequestTag/`*`tag-key`* para permitir que uma entidade principal adicione apenas etiquetas específicas, ou pode impedir que uma entidade principal adicione etiquetas com chaves de etiqueta específicas.

- leis: RequestTag
- aws:ResourceTag/tag-key (somente políticas do IAM)
- leis: TagKeys

Como prática recomendada ao usar tags para controlar o acesso a chaves, use as chaves de condição `aws:RequestTag/`*`tag-key`* ou `aws:TagKeys` para determinar quais tags (ou chaves de tag) são permitidas.

Por exemplo, a política do IAM a seguir é semelhante à anterior. No entanto, essa política permite que as entidades principais criem etiquetas (`TagResource`) e excluam etiquetas `UntagResource` somente para etiquetas com um chave de etiqueta `Project`.

Como `TagResource` as `UntagResource` solicitações podem incluir várias tags, você deve especificar um operador `ForAllValues` ou `ForAnyValue` definir com a TagKeys condição aws:. O operador `ForAnyValue` exige que pelo menos uma das chaves de etiqueta na solicitação corresponda a uma das chaves de etiqueta na política. O operador `ForAllValues` requer que todas as chaves de etiqueta na solicitação correspondam a uma das chaves de etiqueta na política. O `ForAllValues` operador também retorna `true` se não houver tags na solicitação, mas TagResource UntagResource falhará quando nenhuma tag for especificada. Para detalhes sobre os operadores de conjunto, consulteUsar várias chaves e valores, no Manual do usuário do IAM.

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

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

Usar tags para controlar o acesso a chaves

Você pode controlar o acesso à criptografia de AWS pagamento com base nas tags na chave. Por exemplo, você pode escrever uma política do IAM que permite que as entidades principais habilitem e desabilitem somente as chaves que possuem uma tag específica. Ou você pode usar uma política do IAM para impedir que as entidades principais usem chaves em operações de criptografia, a menos que a chave tenha uma tag específica.

Esse recurso faz parte do suporte à criptografia de AWS pagamento para controle de acesso baseado em atributos (ABAC). Para obter informações sobre o uso de tags para controlar o acesso aos AWS recursos, consulte Para AWS que serve o ABAC? e controlar o acesso aos AWS recursos usando tags de recursos no Guia do usuário do IAM.

> **Note**

AWS A criptografia de pagamento é compatível com a chave de contexto de condição global aws:ResourceTag/tag-key, que permite controlar o acesso às chaves com base nas tags da chave. Como várias chaves podem ter a mesma tag, esse atributo permite que você aplique a permissão a um conjunto selecionado de chaves. Também é possível alterar facilmente as chaves no conjunto alterando suas tags.

Na criptografia AWS de pagamento, a chave de `aws:ResourceTag/`*`tag-key`* condição é suportada somente nas políticas do IAM. Ela não é suportada em políticas de chave, que se aplicam somente a uma chave, ou em operações que não usam uma chave específica, como as ListAliasesoperações ListKeysou.

Controlar o acesso com etiquetas é uma maneira simples, escalável e flexível de gerenciar permissões. No entanto, se isso não for projetado e gerenciado corretamente, poderá permitir ou negar acesso às chaves inadvertidamente. Se estiver usando etiquetas para controlar o acesso, considere as seguintes práticas.

- Use tags para reforçar a prática recomendada do acesso acesso com privilégio mínimo. Conceda às entidades principais do IAM somente as permissões de que eles precisam nas chaves que elas devem usar ou gerenciar. Por exemplo, use tags para rotular as chaves usadas para um projeto. Em seguida, dê permissão à equipe do projeto para usar somente chaves com a tag do projeto.
- Tenha cuidado ao conceder às entidades principais as permissões `payment-cryptography:TagResource` e `payment-cryptography:UntagResource`, com as quais elas podem adicionar, editar e excluir etiquetas. Quando você usa tags para controlar o acesso a chaves, a alteração de uma tag pode dar permissão às entidades principais para usar chaves que, de outra forma, elas não teriam permissão de usar. Ele também pode negar acesso a chaves que outras entidades principais exigem para realizar seus trabalhos. Os administradores de chaves que não tiverem permissão para alterar políticas de chave ou criar concessões poderão controlar o acesso às chaves se tiverem permissão para gerenciar tags.

  Sempre que possível, use uma condição de política, como `aws:RequestTag/`*`tag-key`* ou `aws:TagKeys`, para limitar as permissões de marcação de uma entidade principal para tags ou padrões de tags específicos em chaves específicas.
- Revise os diretores Conta da AWS que atualmente têm permissões de marcação e desmarcação e ajuste-os, se necessário. As políticas do IAM podem conceder permissões de marcação ou desmarcação em todas as chaves. Por exemplo, a política gerenciada pelo administrador permite que as entidades principais marquem, desmarquem e listem tags em todas as chaves.

- Antes de definir uma política que dependa de uma tag, revise as tags nas chaves do seu Conta da AWS. Certifique-se de que sua política se aplica somente às etiquetas que você pretende incluir. Use CloudTrail registros e CloudWatch alarmes para alertá-lo sobre alterações nas tags que possam afetar o acesso às suas chaves.
- As condições de políticas baseadas em etiquetas usam correspondência de padrões. Elas não estão vinculadas a uma instância específica de uma etiqueta. Uma política que usa chaves de condição baseadas em etiquetas afeta todas as etiquetas novas e existentes que correspondem ao padrão. Se você excluir e recriar uma etiqueta que corresponde a uma condição de política, a condição se aplicará à nova etiqueta, assim como à antiga.

Por exemplo, considere a seguinte política do IAM. Isso permite que as entidades principais chamem as operações de Decrypt somente em chaves em sua conta que estejam na região Leste dos EUA (Norte da Virgínia) e tenham uma tag `"Project"="Alpha"`. Você pode anexar essa política a funções no exemplo do projeto Alpha.

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

O exemplo de política do IAM a seguir permite que as entidades principais usem qualquer chave na conta para determinadas operações de criptografia. Porém, ela proíbe as entidades principais de usar estas operações criptográficas em chaves com uma tag `"Type"="Reserved"` ou sem a tag `"Type"`.

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

```
  ]
}
```

Compreendendo os principais atributos da chave AWS de criptografia de pagamento

Um princípio do gerenciamento adequado de chaves é que as chaves têm um escopo adequado e podem ser usadas apenas para operações permitidas. Dessa forma, certas chaves só podem ser criadas com determinados modos de uso de chaves. Sempre que possível, isso se alinha aos modos de uso disponíveis, conforme definido pelo TR-31.

Embora a criptografia de AWS pagamento impeça que você crie chaves inválidas, combinações válidas são fornecidas aqui para sua conveniência.

Chaves simétricas

- TR31_B0_BASE_DERIVATION_KEY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: { DeriveKey = true}, { NoRestrictions = true}
- TR31_C0_CARD_VERIFICATION_KEY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: {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 de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: { DeriveKey = true}, { NoRestrictions = true}
- TR31_E1_EMV_MKEY_CONFIDENTIALITY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: { DeriveKey = true}, { NoRestrictions = true}

- TR31_E2_EMV_MKEY_INTEGRITY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: { DeriveKey = true}, { NoRestrictions = true}
- TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: { DeriveKey = true}, { NoRestrictions = true}
- TR31_E5_EMV_MKEY_CARD_PERSONALIZATION
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: { DeriveKey = true}, { NoRestrictions = true}
- TR31_E6_EMV_MKEY_OTHER
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: { DeriveKey = true}, { NoRestrictions = true}
- TR31_K0_KEY_ENCRYPTION_KEY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_M3_ISO_9797_3_MAC_KEY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_M6_ISO_9797_5_CMAC_KEY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_M7_HMAC_KEY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_P0_PIN_ENCRYPTION_KEY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256

  - Combinação permitida dos principais modos de uso: {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 chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_V2_VISA_PIN_VERIFICATION_KEY
  - Algoritmos de chave permitidos: TDES_2KEY ,TDES_3KEY ,AES_128 ,AES_192 ,AES_256
  - Combinação permitida dos principais modos de uso: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}

Chaves assimétricas

- TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION
  - Algoritmos de chave permitidos: RSA_2048 ,RSA_3072 ,RSA_4096
  - Combinação permitida dos principais modos de uso: { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } ,{ Encrypt = true, Wrap = true } ,{ Decrypt = true, Unwrap = true }
  - OBSERVAÇÃO:: {Encrypt = true, Wrap = true} é a única opção válida ao importar uma chave pública destinada a criptografar dados ou agrupar uma chave
- TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE
  - Algoritmos de chave permitidos: RSA_2048 ,RSA_3072 ,RSA_4096
  - Combinação permitida dos principais modos de uso: {Sign = true}, {Verify = true}
  - NOTA:: {Verify = true} é a única opção válida ao importar uma chave destinada à assinatura, como certificado raiz, certificado intermediário ou certificados de assinatura para TR-34.

Operações de dados

Depois de estabelecer uma chave AWS de criptografia de pagamento, ela pode ser usada para realizar operações criptográficas. Operações diferentes realizam diferentes tipos de atividade, desde criptografia, hashing até algoritmos específicos de domínio, como geração de CVV2.

Os dados criptografados não podem ser descriptografados sem a chave de decodificação correspondente (a chave simétrica ou a chave privada, dependendo do tipo de criptografia). Da mesma forma, os algoritmos de hash e específicos de domínio não podem ser verificados sem a chave simétrica ou a chave pública.

Para obter informações sobre tipos de chaves válidas para operações específicas, consulte Chaves válidas para operações criptográficas

> **Note**
>
> Recomendamos o uso de dados de teste em um ambiente que não seja de produção. O uso de chaves e dados de produção (PAN, BDK ID etc.) em um ambiente que não seja de produção pode afetar seu escopo de conformidade, como PCI DSS e PCI P2PE.

Tópicos

Criptografe, descriptografe e recriptografe dados

Métodos de criptografia e descriptografia podem ser usados para criptografar ou descriptografar dados usando uma variedade de técnicas simétricas e assimétricas, incluindo TDES, AES e RSA. Esses métodos também oferecem suporte a chaves derivadas usando as técnicas DUKPT e EMV.

Para casos de uso em que você deseja proteger dados com uma nova chave sem expor os dados subjacentes, o ReEncrypt comando também pode ser usado.

> **Note**
>
> Ao usar as funções de criptografia/descriptografia, presume-se que todas as entradas estejam em hexBinary - por exemplo, um valor de 1 será inserido como 31 (hexadecimal) e um t minúsculo será representado como 74 (hexadecimal). Todas as saídas também serão geradas em hexBinary.

Para obter detalhes sobre todas as opções disponíveis, consulte o Guia de API para criptografar, descriptografare recriptografar.

Tópicos

- Criptografar dados
- Descriptografar dados

Criptografar dados

A `Encrypt Data` API é usada para criptografar dados usando chaves de criptografia de dados simétricas e assimétricas, bem como chaves derivadas de DUKPT e EMV. Vários algoritmos e variações são compatíveis, incluindo `TDES`, `RSA` e `AES`.

As entradas primárias são a chave de criptografia usada para criptografar os dados, os dados de texto simples no formato HexBinary a serem criptografados e os atributos de criptografia, como vetor e modo de inicialização, para cifras de bloco, como TDES. Os dados em texto simples precisam estar em múltiplos de 8 bytes para`TDES`, 16 bytes para `AES` e o tamanho da chave no caso de. `RSA` As entradas de chave simétricas (TDES, AES, DUKPT, EMV) devem ser preenchidas nos casos em que os dados de entrada não atendam a esses requisitos. A tabela a seguir mostra o tamanho máximo do texto simples para cada tipo de chave e o tipo de preenchimento que você define `EncryptionAttributes` para as chaves RSA.

Tipo de preenchimento	RSA_2048	RSA_3072	RSA_4096
`OAEP SHA1`	428	684	940

Tipo de preenchimento	RSA_2048	RSA_3072	RSA_4096
OAEP_SHA256	380	636	892
OAEP_SHA512	252	508	764
PKCS1	488	744	1000
None	488	744	1000

As saídas primárias incluem os dados criptografados como texto cifrado no formato hexBinary e o valor da soma de verificação da chave de criptografia. Para obter detalhes sobre todas as opções disponíveis, consulte o Guia de API do Encrypt.

Exemplos

- Criptografe dados usando a chave simétrica AES
- Criptografe dados usando a chave DUKPT
- Criptografe dados usando a chave simétrica derivada do EMV
- Criptografe dados usando uma chave RSA

Criptografe dados usando a chave simétrica AES

> **Note**
>
> Todos os exemplos presumem que a chave relevante já existe. As chaves podem ser criadas usando a CreateKeyoperação ou importadas usando a ImportKeyoperação.

Example

Neste exemplo, criptografaremos dados em texto simples usando uma chave simétrica que foi criada usando a CreateKeyOperação ou importada usando a Operação. ImportKey Para essa operação, a chave deve ter sido KeyModesOfUse definida como `Encrypt` e KeyUsage definida como`TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`. Consulte Chaves para operações criptográficas para obter mais opções.

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

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

Criptografe dados usando a chave DUKPT

Example

Neste exemplo, criptografaremos dados em texto simples usando uma chave DUKPT. AWS Suportes de criptografia de pagamento TDES e chaves AES DUKPT. Para essa operação, a chave deve ter sido KeyModesOfUse definida como DeriveKey e KeyUsage definida comoTR31_B0_BASE_DERIVATION_KEY. Consulte Chaves para operações criptográficas para obter mais opções.

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

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

Criptografe dados usando a chave simétrica derivada do EMV

Example

Neste exemplo, criptografaremos dados de texto não criptografado usando uma chave simétrica derivada do EMV que já foi criada. Você pode usar um comando como esse para enviar dados para um cartão EMV. Para essa operação, a chave deve ter sido KeyModesOfUse definida como `Derive` e KeyUsage definida como `TR31_E1_EMV_MKEY_CONFIDENTIALITY` ou`TR31_E6_EMV_MKEY_OTHER`. Consulte Chaves para operações criptográficas para obter mais detalhes.

```
$ 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"
}
```

Criptografe dados usando uma chave RSA

Example

Neste exemplo, criptografaremos dados em texto simples usando uma chave pública RSA que foi importada usando a operação. ImportKey Para essa operação, a chave deve ter sido KeyModesOfUse definida como `Encrypt` e KeyUsage definida como`TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION`. Consulte Chaves para operações criptográficas para obter mais opções.

Para esquemas de preenchimento como o PKCS #7 ou outros atualmente não compatíveis, aplique antes de chamar o serviço e selecione nenhum preenchimento ao omitir o indicador de preenchimento 'Asymmetric={}'

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

```
{
 "CipherText":
 "12DF6A2F64CC566D124900D68E8AFEAA794CA819876E258564D525001D00AC93047A83FB13 \
 E73F06329A100704FA484A15A49F06A7A2E55A241D276491AA91F6D2D8590C60CDE57A642BC64A897F4832A3930
 \
 0FAEC7981102CA0F7370BFBF757F271EF0BB2516007AB111060A9633D1736A9158042D30C5AE11F8C5473EC70F067
 \
 72590DEA1638E2B41FAE6FB1662258596072B13F8E2F62F5D9FAF92C12BB70F42F2ECDCF56AADF0E311D4118FE3591
 \
 FB672998CCE9D00FFFE05D2CD154E3120C5443C8CF9131C7A6A6C05F5723B8F5C07A4003A5A6173E1B425E2B5E42AD
 \
 7A2966734309387C9938B029AFB20828ACFC6D00CD1539234A4A8D9B94CDD4F23A",
 "KeyArn": "arn:aws:payment-cryptography:us-east-1:529027455495:key/5dza7xqd6soanjtb",
 "KeyCheckValue": "FF9DE9CE"
}
```

Descriptografar dados

A `Decrypt Data` API é usada para descriptografar dados usando chaves de criptografia de dados simétricas e assimétricas, bem como chaves derivadas de DUKPT e EMV. Vários algoritmos e variações são compatíveis, incluindo `TDES`, `RSA` e `AES`.

As entradas primárias são a chave de descriptografia usada para descriptografar dados, dados de texto cifrado no formato hexBinary a serem descriptografados e atributos de descriptografia, como vetor de inicialização, modo como cifras de bloco etc. As saídas primárias incluem os dados descriptografados como texto simples no formato hexBinary e o valor da soma de verificação da chave de decodificação. Para obter detalhes sobre todas as opções disponíveis, consulte o Guia de API para desencriptação.

Exemplos

- Descriptografe dados usando a chave simétrica AES
- Descriptografe dados usando a chave DUKPT
- Descriptografe dados usando a chave simétrica derivada do EMV
- Descriptografe dados usando uma chave RSA

Descriptografe dados usando a chave simétrica AES

Example

Neste exemplo, decifraremos dados de texto cifrado usando uma chave simétrica. Este exemplo mostra uma `AES` chave, mas `TDES_2KEY` ela também `TDES_3KEY` é suportada. Para essa operação, a chave deve ter sido KeyModesOfUse definida como `Decrypt` e KeyUsage definida como`TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`. Consulte Chaves para operações criptográficas para obter mais opções.

```
$ 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"
}
```

Descriptografe dados usando a chave DUKPT

> **Note**
>
> O uso de dados de decodificação com DUKPT para transações P2PE pode retornar o PAN do cartão de crédito e outros dados do titular do cartão ao seu aplicativo, que precisarão ser contabilizados ao determinar o escopo do PCI DSS.

Example

Neste exemplo, decifraremos dados de texto cifrado usando uma chave DUKPT que foi criada usando a Operação ou importada usando a CreateKeyOperação. ImportKey Para essa operação, a chave deve ter sido KeyModesOfUse definida como `DeriveKey` e KeyUsage definida como`TR31_B0_BASE_DERIVATION_KEY`. Consulte Chaves para operações criptográficas para obter mais opções. Quando você usa `DUKPT`, para o algoritmo `TDES`, o comprimento dos dados do texto cifrado deve ser um múltiplo de 16 bytes. Para o algoritmo `AES`, o comprimento dos dados do texto cifrado deve ser um 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"
}
```

Descriptografe dados usando a chave simétrica derivada do EMV

Example

Neste exemplo, decifraremos dados de texto cifrado usando uma chave simétrica derivada do EMV que foi criada usando a operação ou importada usando a operação. CreateKeyImportKey Para essa operação, a chave deve ter sido KeyModesOfUse definida como `Derive` e KeyUsage definida como `TR31_E1_EMV_MKEY_CONFIDENTIALITY` ou`TR31_E6_EMV_MKEY_OTHER`. Consulte Chaves para operações criptográficas para obter mais detalhes.

```
$ 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"
}
```

Descriptografe dados usando uma chave RSA

Example

Neste exemplo, decifraremos dados de texto cifrado usando um par de chaves RSA que foi criado usando a operação. CreateKey Para essa operação, a chave deve estar KeyModesOfUse configurada para habilitar `Decrypt` e KeyUsage definida como`TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION`. Consulte Chaves para operações criptográficas para obter mais opções.

Para esquemas de preenchimento como o PKCS #7 ou outros atualmente não compatíveis, selecione nenhum preenchimento ao omitir o indicador de preenchimento 'Asymmetric= {}' e remova o preenchimento após chamar o serviço.

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

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-
east-1:529027455495:key/5dza7xqd6soanjtb",
    "KeyCheckValue": "FF9DE9CE",
    "PlainText": "31323334313233343132333431323334"
}
```

Gerar e verificar dados do cartão

Gerar e verificar dados do cartão incorpora dados derivados dos dados do cartão, por exemplo, CVV, CVV2, CVC e DCVV.

Tópicos

Gerar dados do cartão

A API `Generate Card Data` é usada para gerar dados do cartão usando algoritmos como CVV, CVV2 ou Dynamic CVV2. Para saber quais chaves podem ser usadas para esse comando, consulte a seção Chaves válidas para operações criptográficas.

Example

Neste exemplo, geraremos um CVV/CVV2 para um PAN específico com entradas de PAN e a data de validade do cartão. A data de validade do cartão pode estar no formato MMAA ou AAMM, mas deve corresponder a todos os usos subsequentes para que a validação funcione corretamente. Isso pressupõe que você tenha gerado uma chave de verificação do cartão.

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pig --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue2={CardExpiryDate=0123}
```

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

Verificar dados do cartão

O `Verify Card Data` é usado para verificar dados que foram criados usando algoritmos de pagamento que dependem de entidades principais de criptografia, como `DISCOVER_DYNAMIC_CARD_VERIFICATION_CODE`.

Os valores de entrada são normalmente fornecidos como parte de uma transação de entrada para um emissor ou parceiro de plataforma compatível. Para verificar um criptograma ARQC (usado para cartões com chips EMV), consulte Verificar ARQC.

Se o valor for verificado, a API retornará http/200. Se o valor não for verificado, ela retornará http/400.

Example

Neste exemplo, validaremos um CVV/CVV2 para um PAN específico. O CVV2 normalmente é fornecido pelo titular ou usuário do cartão durante a transação para validação. Para validar sua entrada, os seguintes valores serão fornecidos em runtime: chave a ser usada para validação (CVK), PAN, data de validade do cartão e CVV2 inserido. O formato de vencimento do cartão deve corresponder ao utilizado na geração inicial do valor.

```
$ 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"
    }
```

Gerar, traduzir e verificar dados de PIN

As funções de dados de PIN permitem gerar PINs aleatórios, valores de verificação de PIN (PVV) e validar PINs criptografados de entrada em relação a compensações de PIN ou PVV.

A tradução de PINs permite traduzir um PIN de uma chave funcional para outra sem expor o PIN em texto não criptografado, conforme especificado pelo requisito n.º 1 do PCI PIN.

> **Note**
>
> Como a geração e validação de PINs geralmente são funções do emissor e a tradução de PINs é uma função típica do adquirente, recomendamos que você considere o acesso menos privilegiado e defina políticas apropriadas para o caso de uso do seu sistema.

Tópicos

Traduzir dados de PIN

As funções de tradução de dados de PIN são usadas para traduzir dados de PIN criptografados de um conjunto de chaves para outro sem que os dados criptografados saiam do HSM. Isso é usado para criptografia P2PE, na qual as chaves de trabalho devem mudar, mas o sistema de processamento não precisa ou não tem permissão para descriptografar os dados. As entradas primárias são os dados criptografados, a chave de criptografia usada para criptografar os dados, os parâmetros usados para gerar os valores de entrada. O outro conjunto de entradas são os parâmetros de saída solicitados, como a chave a ser usada para criptografar e os parâmetros usados para criar essa saída. As saídas primárias são um conjunto de dados recém-criptografado, bem como os parâmetros usados para gerá-lo.

> **Note**
>
> Os tipos de chave AES são compatíveis apenas com blocos de PIN ISO Format 4.

Tópicos

PIN de PEK para DUKPT

Example

Neste exemplo, traduziremos um PIN da criptografia PEK TDES usando um bloco de PIN ISO 0 para um bloco de PIN ISO 4 AES usando o algoritmo DUKPT. Normalmente, isso pode ser feito ao contrário, quando um terminal de pagamento criptografa um PIN em ISO 4 e, em seguida, ele pode ser traduzido de volta para TDES para processamento posterior.

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

```
    {
            "PinBlock": "1F4209C670E49F83E75CC72E81B787D9",
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt  --outgoing-key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/4pmyquwjs3yj4vwe",
            "KeyCheckValue": "7CC9E2"
        }
```

PIN de DUKPT para AWK

Example

Neste exemplo, traduziremos um PIN de um PIN criptografado DUKPT AES para um PIN criptografado sob uma AWK. Funcionalmente, é o inverso do exemplo anterior.

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

```
east-2:111122223333:key/4pmyquwjs3yj4vwe --incoming-translation-attributes
 IsoFormat4="{PrimaryAccountNumber=171234567890123}" --incoming-dukpt-attributes
 KeySerialNumber="FFFF9876543210E00008"
```

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

Gerar dados de PIN

As funções de geração de dados de PIN são usadas para gerar valores relacionados ao PIN, como PVV e compensações de bloco de PIN usados para validar a entrada de PINs pelos usuários durante a transação ou a autorização. Essa API também pode gerar um novo PIN aleatório usando vários algoritmos.

Example

Neste exemplo, geraremos um novo pino (aleatório) usando o esquema de pinos Visa, onde as saídas serão criptografadas `PIN block` (PinData. PinBlock) e um `PVV` (PinData.offset). As principais entradas são `PAN`, `Pin Verification Key`, `Pin Encryption Key` e `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 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"
        }
    }
```

Verificar dados de PIN

As funções de verificação de dados de PIN são usadas para verificar se um PIN está correto. Isso normalmente envolve comparar o valor do PIN armazenado anteriormente com o que foi inserido pelo titular do cartão em um POI. Essas funções comparam dois valores sem expor o valor subjacente de nenhuma das fontes.

Validar PINs criptografados

Example

Neste exemplo, validaremos um PIN para um PAN específico. Em geral, o PIN é fornecido pelo titular do cartão ou pelo usuário durante a transação para validação e é comparado com o valor registrado (fornecido como um valor criptografado). Para validar esta entrada, os seguintes valores também serão fornecidos em runtime: a chave usada para criptografar o PIN de entrada (geralmente é um `IWK`), PAN e o valor a ser verificado (um `PVV` ou `PIN offset`).

Se a criptografia AWS de pagamento conseguir validar o PIN, um http/200 será retornado. Se o PIN não for validado, ela retornará um 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",

        }
```

Verificar o criptograma de solicitação de autenticação (ARQC)

A API do criptograma de solicitação de autenticação de verificação é usada para verificar o ARQC. A geração do ARQC está fora do escopo da criptografia de AWS pagamento e normalmente é realizada em um cartão com chip EMV (ou equivalente digital, como carteira móvel) durante o período de autorização da transação. Um ARQC é exclusivo para cada transação e tem como objetivo mostrar criptograficamente a validade do cartão e garantir que os dados da transação correspondam exatamente à transação atual (esperada).

AWS A criptografia de pagamento fornece uma variedade de opções para validar o ARQC e gerar valores ARPC opcionais, incluindo aqueles definidos no EMV 4.4 Livro 2 e outros esquemas usados pela Visa e pela Mastercard. Para obter uma lista completa de todas as opções disponíveis, consulte a VerifyCardValidationData seção no Guia da API.

Os criptogramas ARQC normalmente requerem as seguintes entradas (embora isso possa variar de acordo com a implementação):

- PAN - Especificado no PrimaryAccountNumber campo
- Número de sequência PAN (PSN) - especificado no campo PanSequenceNumber
- Método de derivação de chave, como Chave de sessão comum (CSK) - especificado no SessionKeyDerivationAttributes
- Modo de derivação de chave mestra (como EMV Opção A) - Especificado no MajorKeyDerivationMode
- Dados da transação - uma sequência de vários dados de transação, terminal e cartão, como valor e data - especificados no TransactionData campo
- Chave mestra do emissor - a chave mestra usada para derivar a chave de criptograma (AC) usada para proteger transações individuais e especificada no campo KeyIdentifier

Tópicos

- Construir dados de transação
- Preenchimento de dados da transação

- Exemplos

Construir dados de transação

O conteúdo exato (e a ordem) do campo de dados da transação variam de acordo com a implementação e o esquema de rede, mas os campos mínimos recomendados (e a sequência de concatenação) são definidos no EMV 4.4 Livro 2, Seção 8.1.1 - Seleção de dados. Se os três primeiros campos forem valor (17,00), outro valor (0,00) e país de compra, isso resultaria nos dados da transação começando da seguinte forma:

- 000000001700: quantidade: 12 posições decimais implícitas de dois dígitos
- 000000000000: outro valor: 12 posições implícitas decimais de dois dígitos
- 0124: código de país de quatro dígitos
- Dados da transação de saída (parcial): 00000000170000000000000000124

Preenchimento de dados da transação

Os dados da transação devem ser preenchidos antes de serem enviados para o serviço. A maioria dos esquemas usa preenchimento ISO 9797 Método 2, em que uma string hexadecimal é anexada por hexadecimal 80 seguido por 00 até que o campo seja um múltiplo do tamanho do bloco de criptografia; 8 bytes ou 16 caracteres para TDES e 16 bytes ou 32 caracteres para AES. A alternativa (método 1) não é tão comum, mas usa somente 00 como caracteres de preenchimento.

Preenchimento ISO 9797 Método 1

Sem preenchimento:
00000000170000000000000008400080008000084016051700000000093800000B03011203 (74 caracteres ou 37 bytes)

Com preenchimento:
00000000170000000000000008400080008000084016051700000000093800000B03011203000000 (80 caracteres ou 40 bytes)

Preenchimento ISO 9797 Método 2

Sem preenchimento:
00000000170000000000000008400080008000084016051700000000093800000B1F220103000000 (80 caracteres ou 40 bytes)

Com preenchimento:
0000000017000000000000000840008000800008401605170000000093800000B1F220103000000800000
(88 caracteres ou 44 bytes)

Exemplos

Visa CVN10

Example

Neste exemplo, validaremos um ARQC gerado usando Visa CVN10.

Se a criptografia AWS de pagamento for capaz de validar o ARQC, um http/200 será retornado. Se o ARQC não for validado, ele retornará uma resposta 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
 0000000017000000000000000840008000800008401605170000000093800000B03011203000000 \
--session-key-derivation-attributes='{"Visa":{"PanSequenceNumber":"01" \
,"PrimaryAccountNumber":"9137631040001422"}}'
```

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

Visa CVN18 e Visa CVN22

Example

Neste exemplo, validaremos um ARQC gerado com o uso do Visa CVN18 ou CVN22. As operações criptográficas são as mesmas entre CVN18 e CVN22, mas os dados contidos nos dados da transação variam. Comparado ao CVN10, um criptograma completamente diferente é gerado mesmo com as mesmas entradas.

Se a criptografia AWS de pagamento for capaz de validar o ARQC, um http/200 será retornado. Se o ARQC não for validado, ela retornará um 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
 00000000170000000000000084000800080000840160517000000000938000 0B1F22010300000000000
 \
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"
}
```

Gerar e verificar MAC

Os Message Authentication Codes (MAC – códigos de autenticação de mensagens) são normalmente usados para autenticar a integridade de uma mensagem (independentemente de ela ter sido modificada). Hashes criptográficos como Hash-Based Message Authentication Code (HMAC – código de autenticação de mensagens baseado em hash), Cipher-based Message Authentication Code (CBC-MAC e CMAC – código de autenticação de mensagens baseado em cifra) também fornecem garantia adicional ao remetente do MAC ao utilizar criptografia. O HMAC é baseado em funções de hash, enquanto o CMAC é baseado em cifras de bloco.

Todos os algoritmos MAC deste serviço combinam uma função hash criptográfica e uma chave secreta compartilhada. Eles usam uma mensagem e uma chave secreta, como o material de chave em uma chave, e retornam uma tag ou MAC exclusivo. Se até mesmo um caractere da mensagem mudar, ou se a chave secreta mudar, a tag resultante será totalmente diferente. Ao exigir uma chave secreta, os MACs criptográficos também fornecem autenticidade. É impossível gerar um MAC idêntico sem a chave secreta. Os MACs criptográficos às vezes são chamados de assinaturas simétricas porque funcionam como assinaturas digitais, mas usam uma única chave para assinatura e verificação.

A AWS Payment Cryptography oferece suporte a vários tipos de MACs:

ALGORITMO 1 ISO9797

Indicado por `KeyUsage` de ISO9797_ALGORITHM1

ALGORITMO 3 ISO9797 (MAC de varejo)

Indicado por `KeyUsage` de ISO9797_ALGORITHM3

ALGORITMO 5 ISO9797 (CMAC)

Indicado por `KeyUsage` de TR31_M6_ISO_9797_5_CMAC_KEY

HMAC

Indicado por `KeyUsage` de TR31_M7_HMAC_KEY, incluindo HMAC_SHA224, HMAC_SHA256, HMAC_SHA384 e HMAC_SHA512

Tópicos

Gerar MAC

A API de gerar MAC é usada para autenticar dados relacionados ao cartão, como rastrear dados de uma tarja magnética, usando valores de dados conhecidos para gerar um MAC (Código de autenticação de mensagens) para validação de dados entre as partes emissoras e receptoras. Os dados usados para gerar MACs incluem dados de mensagens, chaves secretas de criptografia MAC e algoritmo MAC para gerar um valor MAC exclusivo para transmissão. A parte receptora do MAC usará os mesmos dados da mensagem MAC, chave de criptografia MAC e algoritmo para reproduzir outro valor MAC para comparação e autenticação de dados. Se qualquer caractere da mensagem for alterado ou a chave MAC usada para verificação não for idêntica, o valor MAC resultante será diferente. A API suporta chaves de criptografia DUPKT MAC, HMAC e EMV MAC para essa operação.

O valor de entrada para `message-data` devem ser dados hexBinary.

Neste exemplo, geraremos um HMAC (Código de autenticação de mensagens baseado em hash) para autenticação de dados do cartão usando o algoritmo HMAC `HMAC_SHA256` e a chave de criptografia HMAC. A chave deve ter KeyUsage definido como `TR31_M7_HMAC_KEY` e

KeyModesOfUse como `Generate`. A chave MAC pode ser criada com AWS Payment Cryptography ao chamar CreateKey ou importada ao chamar ImportKey.

Example

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

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

Verificar MAC

A API de verificação de MAC é usada para verificar o MAC (Código de autenticação de mensagens) para autenticação de dados relacionados a cartões. Ela deve usar a mesma chave de criptografia usada durante a geração do MAC para reproduzir o valor do MAC para autenticação. A chave de criptografia MAC pode ser criada com AWS Payment Cryptography ao chamar CreateKey ou importada ao chamar ImportKey. A API suporta chaves de criptografia DUPKT MAC, HMAC e EMV MAC para essa operação.

Se o valor for verificado, o parâmetro de resposta `MacDataVerificationSuccessful` retornará `Http/200`, caso contrário, `Http/400` com uma mensagem indicando que `Mac verification failed`.

Neste exemplo, verificaremos um HMAC (Código de autenticação de mensagens baseado em hash) para autenticação de dados do cartão usando o algoritmo HMAC `HMAC_SHA256` e a chave de criptografia HMAC. A chave deve ter KeyUsage definido como `TR31_M7_HMAC_KEY` e KeyModesOfUse como `Verify`.

Example

```
$ aws payment-cryptography-data verify-mac \
```

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

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

Chaves válidas para operações criptográficas

Certas chaves podem ser usadas apenas para operações específicas. Além disso, algumas operações podem limitar os principais modos de uso das chaves. Consulte a tabela a seguir para ver as combinações permitidas.

> **Note**
>
> Certas combinações, embora permitidas, podem criar situações inutilizáveis, como gerar códigos CVV (`generate`), mas não conseguir verificá-los (`verify`).

Tópicos

- GenerateCardData
- VerifyCardData
- GeneratePinData (para esquemas VISA/ABA)
- GeneratePinData (paraIBM3624)
- VerifyPinData (para esquemas VISA/ABA)
- VerifyPinData (paraIBM3624)
- Descriptografar dados
- Criptografar dados
- Traduzir dados de PIN

- VerifyAuthRequestCryptogram
- Tipos de chave não utilizados

GenerateCardData

Endpoint de API	Operação criptográfica ou algoritmo	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
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_VERIFICATION_VALUE	TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	{ DeriveKey = verdadeiro}
GenerateCardData	• DYNAMIC_CARD_VERIFICATION_CODE	TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS	• TDES_2KEY	{ DeriveKey = verdadeiro}

Endpoint de API	Operação criptográfica ou algoritmo	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
GenerateCardData	• DYNAMIC_CARD_VERIFICATION_VALUE	TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	{ DeriveKey = verdadeiro}

VerifyCardData

Operação criptográfica ou algoritmo	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
• 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 • CARD_VERIFICATION_VALUE_2	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KEY	{ Generate = true }, { Generate = true, Verify = true }
• CARDHOLDER_AUTHENTICATION_V	TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	{ DeriveKey = verdadeiro}

Operação criptográfica ou algoritmo	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
ERIFICATION_VALUE			
• DYNAMIC_CARD_VERIFICATION_CODE	TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS	• TDES_2KEY	{ DeriveKey = verdadeiro}
• DYNAMIC_CARD_VERIFICATION_VALUE	TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	{ DeriveKey = verdadeiro}

GeneratePinData (para esquemas VISA/ABA)

`VISA_PIN or VISA_PIN_VERIFICATION_VALUE`

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
Chave de criptografia de PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY	• { Encrypt = true, Wrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } • { NoRestrictions = verdadeiro}
Chave de geração de PIN	TR31_V2_VISA_PIN_VERIFICATION_KEY	• TDES_3KEY	• { Generate = true }

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
			• { Generate = true, Verify = true }

GeneratePinData (para**IBM3624**)

`IBM3624_PIN_OFFSET,IBM3624_NATURAL_PIN,IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET)`

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
Chave de criptografia 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 } • { NoRestrictions = verdadeiro} Para IBM3624_PIN_OFFSET

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
			• { Encrypt = true, Unwrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } • { NoRestrictions = verdadeiro}
Chave de geração de PIN	TR31_V1_IBM3624_PIN_VERIFICATION_KEY	• TDES_3KEY	• { Generate = true } • { Generate = true, Verify = true }

VerifyPinData (para esquemas VISA/ABA)

`VISA_PIN`

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
Chave de criptografia de PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY	• { Decrypt = true, Unwrap = true } • { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } • { NoRestrictions = verdadeiro}

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
Chave de geração de PIN	TR31_V2_VISA_PIN_VERIFICATION_KEY	• 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 chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
Chave de criptografia 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 } • { NoRestrictions = verdadeiro}

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
Chave de verificação de PIN	TR31_V1_IBM3624_PIN_VERIFICATION_KEY	• TDES_3KEY	• { Verify = true } • { Generate = true, Verify = true }

Descriptografar dados

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = verdadeiro} • { NoRestrictions = verdadeiro}
EMV	TR31_E1_EMV_MKEY_CONFIDENTIALITY TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	• { DeriveKey = verdadeiro}
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}

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
Chaves simétricas	TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• {Decrypt = true, Unwrap=true} • {Encrypt=true, Wrap=true,Decrypt = true, Unwrap=true} • { NoRestrictions = verdadeiro}

Criptografar dados

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = verdadeiro} • { NoRestrictions = verdadeiro}
EMV	TR31_E1_EMV_MKEY_CONFIDENTIALITY TR31_E6_EMV_MKEY_OTHER	• TDES_2KEY	• { DeriveKey = verdadeiro}
RSA	TR31_D1_ASYMMETRIC	• RSA_2048 • RSA_3072	• { Encrypt = true, Wrap=true}

Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
	_KEY_FOR_ DATA_ENCRYPTION	• RSA_4096	• {Encrypt=true, Wrap=true,Decrypt = true, Unwrap=true}
Chaves 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 = verdadeiro}

Traduzir dados de PIN

Direction	Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
Fonte de dados de entrada	DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = verdadeiro} • { NoRestrictions = verdadeiro}
Fonte de dados de entrada	Não-DUKPT (PEK, AWK, IWK etc)	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128	• { Decrypt = true, Unwrap = true }

Direction	Tipo de chave	Uso de chave permitido	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
			• AES_192 • AES_256	• { Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true } • { NoRestrictions = verdadeiro}
Destino de dados de saída	DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = verdadeiro} • { NoRestrictions = verdadeiro}
Destino de dados de saída	Não 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 = verdadeiro}

VerifyAuthRequestCryptogram

Uso de chave permitido	Opção de EMV	Algoritmo de chave permitido	Combinação permitida dos principais modos de uso
• OPÇÃO A • OPÇÃO B	TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS	• TDES_2KEY	• { DeriveKey = verdadeiro}

Tipos de chave não utilizados

Os seguintes tipos de chave não são usados atualmente pela criptografia AWS de pagamento:

- TR31_K1_KEY_BLOCK_PROTECTION_KEY
- TR31_P1_PIN_GENERATION_KEY
- TR31_K3_ASYMMETRIC_KEY_FOR_KEY_AGREEMENT

Segurança na criptografia AWS de pagamentos

A segurança na nuvem AWS é a maior prioridade. Como AWS cliente, você se beneficia de uma arquitetura de data center e rede criada para atender aos requisitos das organizações mais sensíveis à segurança.

A segurança é uma responsabilidade compartilhada entre você AWS e você. O modelo de responsabilidade compartilhada descreve isso como a segurança da nuvem e a segurança na nuvem:

- Segurança da nuvem —AWS é responsável por proteger a infraestrutura que executa AWS os serviços na AWS nuvem. AWS também fornece serviços que você pode usar com segurança. Auditores terceirizados testam e verificam regularmente a eficácia de nossa segurança como parte dos Programas de Conformidade Programas de AWS de . Para saber mais sobre os programas de conformidade que se aplicam à criptografia de AWS pagamento, consulte Serviços da AWS no escopo do programa de conformidade .
- Segurança na nuvem — Sua responsabilidade é determinada pelo AWS serviço que você usa. Você também é responsável por outros fatores, incluindo a confidencialidade de seus dados, os requisitos da empresa e as leis e regulamentos aplicáveis.

Este tópico ajuda você a entender como aplicar o modelo de responsabilidade compartilhada ao usar a criptografia AWS de pagamento. Ele mostra como configurar a criptografia AWS de pagamento para atender aos seus objetivos de segurança e conformidade. Você também aprenderá a usar outros AWS serviços que ajudam a monitorar e proteger seus recursos AWS de criptografia de pagamento.

Tópicos

- Proteção de dados em criptografia AWS de pagamento
- Resiliência na criptografia AWS de pagamentos
- Segurança da infraestrutura em AWS Payment Cryptography
- Melhores práticas de segurança para criptografia AWS de pagamentos

Proteção de dados em criptografia AWS de pagamento

O modelo de responsabilidade AWS compartilhada de se aplica à proteção de dados na criptografia AWS de pagamentos. Conforme descrito neste modelo, AWS é responsável por proteger a infraestrutura global que executa todos os Nuvem AWS. Você é responsável por manter o controle sobre seu conteúdo hospedado nessa infraestrutura. Você também é responsável pelas tarefas de configuração e gerenciamento de segurança dos Serviços da AWS que usa. Para ter mais informações sobre a privacidade de dados, consulte as Perguntas frequentes sobre privacidade de dados. Para ter mais informações sobre a proteção de dados na Europa, consulte a postagem do blog AWS Shared Responsibility Model and GDPR no Blog de segurança da AWS .

Para fins de proteção de dados, recomendamos que você proteja Conta da AWS as credenciais e configure usuários individuais com AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Dessa maneira, cada usuário receberá apenas as permissões necessárias para cumprir suas obrigações de trabalho. Recomendamos também que você proteja seus dados das seguintes formas:

- Use uma autenticação multifator (MFA) com cada conta.
- Use SSL/TLS para se comunicar com os recursos. AWS Exigimos TLS 1.2 e recomendamos TLS 1.3.
- Configure a API e o registro de atividades do usuário com AWS CloudTrail.
- Use soluções de AWS criptografia, juntamente com todos os controles de segurança padrão Serviços da AWS.
- Use serviços gerenciados de segurança avançada, como o Amazon Macie, que ajuda a localizar e proteger dados sigilosos armazenados no Amazon S3.
- Se você precisar de módulos criptográficos validados pelo FIPS 140-2 ao acessar AWS por meio de uma interface de linha de comando ou de uma API, use um endpoint FIPS. Para ter mais informações sobre endpoints do FIPS, consulte Federal Information Processing Standard (FIPS) 140-2.

É altamente recomendável que nunca sejam colocadas informações de identificação confidenciais, como endereços de e-mail dos seus clientes, em marcações ou campos de formato livre, como um campo Nome. Isso inclui quando você trabalha com criptografia AWS de pagamento ou outra Serviços da AWS usando o console, a API ou os AWS SDKs. AWS CLI Quaisquer dados inseridos em tags ou campos de texto de formato livre usados para nomes podem ser usados para logs de faturamento ou de diagnóstico. Se você fornecer um URL para um servidor externo, recomendamos

fortemente que não sejam incluídas informações de credenciais no URL para validar a solicitação a esse servidor.

A AWS Payment Cryptography armazena e protege suas chaves de criptografia de pagamento para torná-las altamente disponíveis e, ao mesmo tempo, fornecer controle de acesso forte e flexível.

Tópicos

Proteger material de chave

Por padrão, a AWS Payment Cryptography protege o material de chave criptográfica das chaves de pagamento gerenciadas pelo serviço. Além disso, a AWS Payment Cryptography oferece opções para importar material de chave criado fora do serviço. Para obter detalhes técnicos sobre chaves de pagamento e material de chave, consulte Detalhes criptográficos da AWS Payment Cryptography.

Criptografia de dados

Os dados na AWS Payment Cryptography consistem em chaves de AWS Payment Cryptography, o material de chave de criptografia que elas representam e seus atributos de uso. O material de chaves existe em texto simples apenas nos módulos de segurança de hardware (HSMs) da AWS Payment Cryptography, e somente quando em uso. Caso contrário, o material e os atributos de chave serão criptografados e armazenados de maneira persistente durável.

O material de chave que a AWS Payment Cryptography gera ou carrega para chaves de pagamento nunca deixam os limites dos HSMs da AWS Payment Cryptography sem criptografia. Ele pode ser exportado criptografado pelas operações da API de AWS Payment Cryptography.

Criptografia inativa

A AWS Payment Cryptography gera material de chave para chaves de pagamento em HSMs listadas no PCI PTS HSM. Quando não estiver em uso, o material de chave é criptografado por uma chave

do HSM e gravado em um armazenamento durável e persistente. O material de chave para chaves de Payment Cryptography e as chaves de criptografia que protegem o material de chave nunca saem dos HSMs em formato de texto simples.

A criptografia e o gerenciamento do material de chave para chaves de criptografia de pagamento são administrados inteiramente pelo serviço.

Para obter mais detalhes, consulte Detalhes criptográficos do serviço de gerenciamento de chaves da AWS.

Criptografia em trânsito

O material de chave que a AWS Payment Cryptography gera ou carrega para as chaves de pagamento nunca é exportado ou transmitido nas operações da API de AWS Payment Cryptography em texto não criptografado. A AWS Payment Cryptography usa identificadores de chave para representar as chaves nas operações com API.

No entanto, algumas operações da API de AWS Payment Cryptography exportam chaves criptografadas por uma chave de troca de chaves previamente compartilhada ou assimétrica. Além disso, os clientes podem usar operações de API para importar material de chave criptografada para chaves de pagamento.

Todas as chamadas de API da AWS Payment Cryptography devem ser assinadas e transmitidas usando Transport Layer Security (TLS). A AWS Payment Cryptography exige versões TLS e pacotes de criptografia definidos pelo PCI como “criptografia robusta”. Todos os endpoints de serviço oferecem suporte ao TLS 1.0–1.3 e ao TLS híbrido pós-quântico.

Para obter mais detalhes, consulte Detalhes criptográficos do serviço de gerenciamento de chaves da AWS.

Privacidade do tráfego entre redes

O AWS Payment Cryptography oferece suporte ao Console de gerenciamento da AWS e a um conjunto de operações de API que permitem criar e gerenciar chaves de pagamento e usá-las em operações criptográficas.

O AWS Payment Cryptography oferece suporte a duas opções de conectividade de rede da sua rede privada para a AWS.

- Uma conexão VPN IPSec pela Internet.

- O AWS Direct Connect, que conecta sua rede interna a um local do AWS Direct Connect por meio de um cabo de fibra óptica Ethernet padrão.

Todas as chamadas da API de Payment Cryptography devem ser assinadas e transmitidas usando Transport Layer Security (TLS). As chamadas também exigem um conjunto de codificação moderno que seja compatível com o sigilo de encaminhamento perfeito. O tráfego para os módulos de segurança de hardware (HSMs) que armazenam material de chave para chaves de pagamento é permitido somente a partir de hosts conhecidos da API de AWS Payment Cryptography na rede interna da AWS.

Para se conectar diretamente à criptografia de pagamento da AWS a partir da sua nuvem privada virtual (VPC) sem enviar tráfego pela Internet pública, use endpoints de VPC, desenvolvidos pela AWS. PrivateLink Para obter mais informações, consulte a Conexão à AWS Payment Cryptography por meio de um endpoint da VPC.

O AWS Payment Cryptography também oferece suporte a uma opção híbrida de troca de chaves pós-quânticas para o protocolo de criptografia de rede Transport Layer Security (TLS). É possível usar essa opção com TLS ao se conectar aos endpoints da API de AWS Payment Cryptography.

Resiliência na criptografia AWS de pagamentos

AWS a infraestrutura global é construída em torno de AWS regiões e zonas de disponibilidade. As regiões fornecem várias zonas de disponibilidade separadas e isoladas fisicamente, que são conectadas com baixa latência, alta throughput e redes altamente redundantes. Com as zonas de disponibilidade, é possível projetar e operar aplicações e bancos de dados que automaticamente executam o failover entre as zonas sem interrupção. As zonas de disponibilidade são mais altamente disponíveis, tolerantes a falhas e escaláveis que uma ou várias infraestruturas de datacenter tradicionais.

Para obter mais informações sobre AWS regiões e zonas de disponibilidade, consulte Infraestrutura AWS global.

Isolamento regional

O AWS Payment Cryptography é um serviço regional que está disponível em várias regiões.

O design isolado regionalmente da AWS Payment Cryptography garante que um problema de disponibilidade em uma região da AWS não afete a operação de AWS Payment Cryptography em nenhuma outra região. A AWS Payment Cryptography foi projetada para garantir zero tempo

de inatividade planejado, com todas as atualizações de software e operações de escalabilidade realizadas de forma perfeita e imperceptível.

O Acordo de Serviço (SLA) de AWS Payment Cryptography inclui um compromisso de serviço de 99,99% para todas as APIs de Payment Cryptography. Para cumprir esse compromisso, a AWS Payment Cryptography garante que todos os dados e informações de autorização necessários para executar uma solicitação de API estejam disponíveis em todos os hosts regionais que recebem a solicitação.

A infraestrutura da AWS Payment Cryptography é replicada em pelo menos três zonas de disponibilidade (AZs) em cada região. Para garantir que várias falhas de host não afetem o desempenho da AWS Payment Cryptography, a AWS Payment Cryptography foi projetada para atender o tráfego de clientes de qualquer uma das AZs em uma região.

As alterações feitas nas propriedades ou permissões de uma chave de pagamento são replicadas para todos os hosts na região a fim de garantir que a solicitação subsequente possa ser processada corretamente por qualquer host na região. As solicitações de operações criptográficas usando sua chave de pagamento são encaminhadas para uma frota de módulos de segurança de hardware (HSMs) da AWS Payment Cryptography, e qualquer um deles pode executar a operação com a chave de pagamento.

Design de vários locatários

O design de vários locatários da AWS Payment Cryptography permite cumprir o SLA de disponibilidade e sustentar altas taxas de solicitação, ao mesmo tempo que protege a confidencialidade de suas chaves e dados.

Há vários mecanismos de imposição de integridade implantados para garantir que a chave de pagamento especificada para a operação criptográfica sempre seja a chave usada.

O material de chave em texto simples para suas chaves de Payment Cryptography é amplamente protegido. Assim que é criado, o material de chave é criptografado no HSM e o material de chave criptografado é imediatamente movido para armazenamento seguro. A chave criptografada é recuperada e descriptografada no HSM no momento do uso. A chave em texto não criptografado permanece na memória do HSM apenas pelo tempo necessário para a conclusão da operação criptográfica. O material de chave em texto não criptografado nunca deixa os HSMs e nunca é gravado no armazenamento persistente.

Para obter mais informações sobre os mecanismos que a AWS Payment Cryptography usa para proteger suas chaves, consulte Detalhes criptográficos da AWS Payment Cryptography.

Segurança da infraestrutura em AWS Payment Cryptography

Como serviço gerenciado, AWS Payment Cryptography é protegido pelos procedimentos AWS globais de segurança de rede descritos no whitepaper Amazon Web Services: Visão geral dos processos de segurança.

Você usa chamadas de API AWS publicadas para acessar AWS Payment Cryptography pela rede. Os clientes devem oferecer suporte a Transport Layer Security (TLS) 1.2 ou posterior. Os clientes também devem ter suporte a conjuntos de criptografia com perfect forward secrecy (PFS) como Ephemeral Diffie-Hellman (DHE) ou Ephemeral Elliptic Curve Diffie-Hellman (ECDHE). A maioria dos sistemas modernos como Java 7 e versões posteriores oferece suporte a esses modos.

Além disso, as solicitações devem ser assinadas usando um ID da chave de acesso e uma chave de acesso secreta associada a uma entidade principal do IAM. Ou é possível usar o AWS Security Token Service (AWS STS) para gerar credenciais de segurança temporárias para assinar solicitações.

Isolamento de hosts físicos

A segurança da infraestrutura física usada pela AWS Payment Cryptography está sujeita aos controles descritos na seção Segurança física e ambiental da Amazon Web Services: visão geral dos processos de segurança. É possível encontrar mais detalhes em relatórios de conformidade e em descobertas de auditoria de terceiros listados na seção anterior.

A criptografia de pagamento da AWS é suportada por módulos de segurança de hardware (HSMs) dedicados listados no commercial-off-the-shelf PCI PTS HSM. O material de chave para chaves de AWS Payment Cryptography é armazenado somente na memória volátil dos HSMs e enquanto a chave de criptografia de pagamento estiver em uso. Os HSMs ficam em racks de acesso controlado nos datacenters da Amazon que impõem controle duplo para qualquer acesso físico. Para obter informações detalhadas sobre a operação de HSMs da AWS Payment Cryptography, consulte Detalhes criptográficos da AWS Payment Cryptography.

Melhores práticas de segurança para criptografia AWS de pagamentos

AWS A criptografia de pagamento oferece suporte a muitos recursos de segurança integrados ou que você pode implementar opcionalmente para aprimorar a proteção de suas chaves de criptografia

e garantir que elas sejam usadas para a finalidade pretendida, incluindo políticas de IAM, um amplo conjunto de chaves de condição de política para refinar suas principais políticas e políticas do IAM e a aplicação integrada das regras de PIN do PCI em relação aos blocos de chaves.

> **Important**
>
> As diretrizes gerais fornecidas não representam uma solução de segurança completa. Como nem todas as melhores práticas são apropriadas para todas as situações, elas não se destinam a ser prescritivas.

- Uso da chave e modos de uso: a criptografia de AWS pagamento segue e impõe as restrições de uso e modo de uso da chave, conforme descrito na Especificação de bloco de chaves interoperável de troca segura de chaves ANSI X9 TR 31-2018 e é consistente com o requisito de segurança de PIN PCI 18-3. Isso limita a capacidade de usar uma única chave para várias finalidades e vincula criptograficamente os metadados da chave (como operações permitidas) ao próprio material da chave. AWS A criptografia de pagamento impõe automaticamente essas restrições, como a de que uma chave de criptografia de chave (TR31_K0_KEY_ENCRYPTION_KEY) também não possa ser usada para decodificação de dados. Consulte Compreendendo os principais atributos da chave AWS de criptografia de pagamento para obter mais detalhes.
- Limite o compartilhamento de material de chave simétrica: compartilhe material de chave simétrica (como chaves de criptografia PIN ou chaves de criptografia de chave) com, no máximo, uma outra entidade. Se houver necessidade de transmitir material confidencial para mais entidades ou parceiros, crie chaves adicionais. AWS A criptografia de pagamento nunca expõe material de chave simétrica ou material de chave privada assimétrica de forma clara.
- Use aliases ou tags para associar chaves a determinados casos de uso ou parceiros: aliases podem ser usados para denotar facilmente o caso de uso associado a uma chave como alias/BIN_12345_CVK para denotar uma chave de verificação de cartão associada ao BIN 12345. Para fornecer mais flexibilidade, considere criar tags como bin=12345, use_case=acquiring,country=us,partner=foo. Aliases e tags também podem ser usados para limitar o acesso, como impor controles de acesso entre a emissão e a aquisição de casos de uso.
- Pratique o acesso com privilégio mínimo: o IAM pode ser usado para limitar o acesso de produção a sistemas em vez de indivíduos, como proibir usuários individuais de criar chaves ou executar operações criptográficas. O IAM também pode ser usado para limitar o acesso a comandos e chaves que podem não ser aplicáveis ao seu caso de uso, como limitar a capacidade de gerar ou

validar PINs para um adquirente. Outra forma de usar o acesso com privilégio mínimo é restringir operações confidenciais (como importação de chaves) a contas de serviço específicas. Consulte AWS Exemplos de políticas baseadas em identidade de criptografia de pagamento para ver exemplos.

Consulte também

- Gerenciamento de identidade e acesso para criptografia AWS de pagamento
- Práticas recomendadas de segurança no IAM, noManual do usuário do IAM

Validação de conformidade para AWS Payment Cryptography

Auditores terceirizados avaliam a segurança e a conformidade da AWS Payment Cryptography como parte de vários programas de conformidade da AWS. Isso inclui SOC, PCI e outros.

A AWS Payment Cryptography foi avaliada por vários padrões PCI além do PCI DSS. Isso inclui PCI PIN Security (PCI PIN) e criptografia PCI ponto a ponto (P2PE). Consulte AWS Artifact para ver os atestados e guias de conformidade disponíveis.

Para obter uma lista de serviços da AWS no escopo de programas de conformidade específicos, consulte Serviços da AWS no escopo pelo programa de conformidade. Para obter informações gerais, consulte Programas de conformidade da AWS.

É possível fazer download de relatórios de auditoria de terceiros usando o AWS Artifact. Para obter mais informações, consulte Fazer download de relatórios no AWS Artifact.

Sua responsabilidade de conformidade ao usar AWS Payment Cryptography é determinada pela confidencialidade dos seus dados, pelos objetivos de conformidade da sua empresa e pelos regulamentos e leis aplicáveis. A AWS fornece os seguintes recursos para ajudar com a conformidade:

- Guias de início rápido de segurança e compatibilidade: estes guias de implantação abordam as considerações de arquitetura e fornecem etapas para implantação de ambientes de linha de base focados em compatibilidade e segurança na AWS.
- Recursos de conformidade da AWS: essa coleção de manuais e guias pode ser aplicada a seu setor e local.
- Avaliar recursos com regras no Guia do desenvolvedor do AWS Config: AWS Config; avalia como suas configurações de recursos estão em conformidade com práticas internas, diretrizes e regulamentos do setor.
- AWS Security Hub: esse serviço da AWS fornece uma visão abrangente do estado da segurança na AWS que ajuda você a verificar a conformidade com os padrões do setor de segurança e as melhores práticas.

Gerenciamento de identidade e acesso para criptografia AWS de pagamento

AWS Identity and Access Management (IAM) é uma ferramenta AWS service (Serviço da AWS) que ajuda o administrador a controlar com segurança o acesso aos AWS recursos. Os administradores do IAM controlam quem pode ser autenticado (conectado) e autorizado (tem permissões) para usar os recursos de criptografia AWS de pagamento. O IAM é um AWS service (Serviço da AWS) que você pode usar sem custo adicional.

Tópicos

Público

A forma como você usa o AWS Identity and Access Management (IAM) difere, dependendo do trabalho que você faz na criptografia AWS de pagamento.

Usuário do serviço — Se você usa o serviço de criptografia de AWS pagamento para fazer seu trabalho, seu administrador fornecerá as credenciais e as permissões de que você precisa. À medida que você usa mais recursos de criptografia de AWS pagamento para fazer seu trabalho, talvez precise de permissões adicionais. Entender como o acesso é gerenciado pode ajudá-lo a solicitar as permissões corretas ao seu administrador. Se não conseguir acessar um atributo no AWS Payment Cryptography, consulte Solução de problemas AWS de identidade e acesso à criptografia de pagamento.

Administrador de serviços — Se você é responsável pelos recursos de criptografia de AWS pagamento em sua empresa, provavelmente tem acesso total à criptografia AWS de pagamento. É seu trabalho determinar quais recursos e recursos AWS de criptografia de pagamento seus usuários do serviço devem acessar. Assim, você deve enviar solicitações ao administrador do IAM para alterar

as permissões dos usuários de seu serviço. Revise as informações nesta página para entender os Introdução ao IAM. Para saber mais sobre como sua empresa pode usar o IAM com criptografia AWS de pagamento, consulteComo a criptografia AWS de pagamento funciona com o IAM.

Administrador do IAM — Se você for administrador do IAM, talvez queira saber detalhes sobre como criar políticas para gerenciar o acesso à criptografia AWS de pagamento. Para ver exemplos AWS de políticas baseadas em identidade de criptografia de pagamento que você pode usar no IAM, consulte. AWS Exemplos de políticas baseadas em identidade de criptografia de pagamento

Autenticando com identidades

A autenticação é a forma como você faz login AWS usando suas credenciais de identidade. Você deve estar autenticado (conectado AWS) como o Usuário raiz da conta da AWS, como usuário do IAM ou assumindo uma função do IAM.

Você pode entrar AWS como uma identidade federada usando credenciais fornecidas por meio de uma fonte de identidade. AWS IAM Identity Center Usuários (IAM Identity Center), a autenticação de login único da sua empresa e suas credenciais do Google ou do Facebook são exemplos de identidades federadas. Quando você faz login como uma identidade federada, o administrador já configurou anteriormente a federação de identidades usando perfis do IAM. Ao acessar AWS usando a federação, você está assumindo indiretamente uma função.

Dependendo do tipo de usuário que você é, você pode entrar no AWS Management Console ou no portal de AWS acesso. Para obter mais informações sobre como fazer login em AWS, consulte Como fazer login Conta da AWS no Guia do Início de Sessão da AWS usuário.

Se você acessar AWS programaticamente, AWS fornece um kit de desenvolvimento de software (SDK) e uma interface de linha de comando (CLI) para assinar criptograficamente suas solicitações usando suas credenciais. Se você não usa AWS ferramentas, você mesmo deve assinar as solicitações. Para obter mais informações sobre como usar o método recomendado para assinar solicitações por conta própria, consulte Assinatura de solicitações de AWS API no Guia do usuário do IAM.

Independentemente do método de autenticação usado, também pode ser exigido que você forneça informações adicionais de segurança. Por exemplo, AWS recomenda que você use a autenticação multifator (MFA) para aumentar a segurança da sua conta. Para saber mais, consulte Autenticação multifator no Guia AWS IAM Identity Center do usuário. Usar a autenticação multifator (MFA) na AWS no Guia do usuário do IAM.

Conta da AWS usuário root

Ao criar uma Conta da AWS, você começa com uma identidade de login que tem acesso completo a todos Serviços da AWS os recursos da conta. Essa identidade é chamada de usuário Conta da AWS raiz e é acessada fazendo login com o endereço de e-mail e a senha que você usou para criar a conta. É altamente recomendável não usar o usuário raiz para tarefas diárias. Proteja as credenciais do usuário raiz e use-as para executar as tarefas que somente ele pode executar. Para obter a lista completa das tarefas que exigem login como usuário raiz, consulte Tarefas que exigem credenciais de usuário raiz no Guia do usuário do IAM.

Grupos e usuários do IAM

Um usuário do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas para uma única pessoa ou aplicativo. Sempre que possível, recomendamos depender de credenciais temporárias em vez de criar usuários do IAM com credenciais de longo prazo, como senhas e chaves de acesso. No entanto, se você tiver casos de uso específicos que exijam credenciais de longo prazo com usuários do IAM, recomendamos alternar as chaves de acesso. Para mais informações, consulte Alterne as chaves de acesso regularmente para casos de uso que exijam credenciais de longo prazo no Guia do usuário do IAM.

Um grupo do IAM é uma identidade que especifica uma coleção de usuários do IAM. Não é possível fazer login como um grupo. É possível usar grupos para especificar permissões para vários usuários de uma vez. Os grupos facilitam o gerenciamento de permissões para grandes conjuntos de usuários. Por exemplo, você pode ter um grupo chamado IAMAdmins e atribuir a esse grupo permissões para administrar recursos do IAM.

Usuários são diferentes de perfis.. Um usuário é exclusivamente associado a uma pessoa ou a uma aplicação, mas um perfil pode ser assumido por qualquer pessoa que precisar dele. Os usuários têm credenciais permanentes de longo prazo, mas os perfis fornecem credenciais temporárias. Para saber mais, consulte Quando criar um usuário do IAM (em vez de um perfil) no Guia do usuário do IAM.

Perfis do IAM

Uma função do IAM é uma identidade dentro da sua Conta da AWS que tem permissões específicas. Ele é semelhante a um usuário do IAM, mas não está associado a uma pessoa específica. Você pode assumir temporariamente uma função do IAM no AWS Management Console trocando de funções. Você pode assumir uma função chamando uma operação de AWS API AWS CLI ou usando

uma URL personalizada. Para mais informações sobre métodos para o uso de perfis, consulte Usar perfis do IAM no Guia do usuário do IAM.

Perfis do IAM com credenciais temporárias são úteis nas seguintes situações:

- Acesso de usuário federado: para atribuir permissões a identidades federadas, você pode criar um perfil e definir permissões para ele. Quando uma identidade federada é autenticada, essa identidade é associada ao perfil e recebe as permissões definidas pelo mesmo. Para mais informações sobre perfis para federação, consulte Criar um perfil para um provedor de identidades de terceiros no Guia do usuário do IAM. Se você usar o IAM Identity Center, configure um conjunto de permissões. Para controlar o que suas identidades podem acessar após a autenticação, o IAM Identity Center correlaciona o conjunto de permissões a um perfil no IAM. Para obter informações sobre conjuntos de permissões, consulte Conjuntos de permissões no Guia do Usuário do AWS IAM Identity Center .
- Permissões temporárias para usuários do IAM: um usuário ou um perfil do IAM pode assumir um perfil do IAM para obter temporariamente permissões diferentes para uma tarefa específica.
- Acesso entre contas: é possível usar um perfil do IAM para permitir que alguém (uma entidade principal confiável) em outra conta acesse recursos em sua conta. Os perfis são a principal forma de conceder acesso entre contas. No entanto, com alguns Serviços da AWS, você pode anexar uma política diretamente a um recurso (em vez de usar uma função como proxy). Para saber a diferença entre funções e políticas baseadas em recurso para acesso entre contas, consulte Como os perfis do IAM diferem das políticas baseadas em recurso no Guia do usuário do IAM.
- Acesso entre serviços — Alguns Serviços da AWS usam recursos em outros Serviços da AWS. Por exemplo, quando você faz uma chamada em um serviço, é comum que esse serviço execute aplicações no Amazon EC2 ou armazene objetos no Amazon S3. Um serviço pode fazer isso usando as permissões do principal de chamada, usando um perfil de serviço ou um perfil vinculado ao serviço.
  - Sessões de acesso direto (FAS) — Quando você usa um usuário ou uma função do IAM para realizar ações AWS, você é considerado principal. Ao usar alguns serviços, você pode executar uma ação que inicia outra ação em um serviço diferente. O FAS usa as permissões do diretor chamando um AWS service (Serviço da AWS), combinadas com a solicitação AWS service (Serviço da AWS) para fazer solicitações aos serviços posteriores. As solicitações do FAS são feitas somente quando um serviço recebe uma solicitação que requer interações com outros Serviços da AWS ou com recursos para ser concluída. Nesse caso, você precisa ter permissões para executar ambas as ações. Para obter detalhes da política ao fazer solicitações de FAS, consulte Encaminhar sessões de acesso.

- Perfil de serviço: um perfil de serviço é um perfil do IAM que um serviço assume para realizar ações em seu nome. Um administrador do IAM pode criar, modificar e excluir um perfil de serviço do IAM. Para obter mais informações, consulte Criar um perfil para delegar permissões a um no Guia do usuário do IAM.
- Função vinculada ao serviço — Uma função vinculada ao serviço é um tipo de função de serviço vinculada a um. AWS service (Serviço da AWS) O serviço pode assumir o perfil para executar uma ação em seu nome. As funções vinculadas ao serviço aparecem em você Conta da AWS e são de propriedade do serviço. Um administrador do IAM pode visualizar, mas não pode editar as permissões para perfis vinculados ao serviço.

- Aplicativos em execução no Amazon EC2 — Você pode usar uma função do IAM para gerenciar credenciais temporárias para aplicativos que estão sendo executados em uma instância do EC2 e fazendo AWS CLI solicitações de API. AWS É preferível fazer isso a armazenar chaves de acesso na instância do EC2. Para atribuir uma AWS função a uma instância do EC2 e disponibilizá-la para todos os seus aplicativos, você cria um perfil de instância anexado à instância. Um perfil de instância contém o perfil e permite que os programas em execução na instância do EC2 obtenham credenciais temporárias. Para obter mais informações, consulte Usar uma função do IAM para conceder permissões a aplicações em execução nas instâncias do Amazon EC2 no Guia do usuário do IAM.

Para saber se deseja usar os perfis do IAM, consulte Quando criar um perfil do IAM (em vez de um usuário) no Guia do usuário do IAM.

Gerenciamento do acesso usando políticas

Você controla o acesso AWS criando políticas e anexando-as a AWS identidades ou recursos. Uma política é um objeto AWS que, quando associada a uma identidade ou recurso, define suas permissões. AWS avalia essas políticas quando um principal (usuário, usuário raiz ou sessão de função) faz uma solicitação. As permissões nas políticas determinam se a solicitação será permitida ou negada. A maioria das políticas é armazenada AWS como documentos JSON. Para mais informações sobre a estrutura e o conteúdo de documentos de políticas JSON, consulte Visão geral das políticas JSON no Guia do usuário do IAM.

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos e em que condições.

Por padrão, usuários e funções não têm permissões. Para conceder aos usuários permissão para executar ações nos recursos de que eles precisam, um administrador do IAM pode criar políticas do

IAM. O administrador pode então adicionar as políticas do IAM a perfis, e os usuários podem assumir os perfis.

As políticas do IAM definem permissões para uma ação, independentemente do método usado para executar a operação. Por exemplo, suponha que você tenha uma política que permite a ação `iam:GetRole`. Um usuário com essa política pode obter informações de função da AWS Management Console AWS CLI, da ou da AWS API.

Políticas baseadas em identidade

As políticas baseadas em identidade são documentos de políticas de permissões JSON que você pode anexar a uma identidade, como usuário, grupo de usuários ou perfil do IAM. Essas políticas controlam quais ações os usuários e funções podem realizar, em quais recursos e em que condições. Para saber como criar uma política baseada em identidade, consulte Criação de política do IAM no Guia do usuário do IAM.

As políticas baseadas em identidade podem ser categorizadas ainda mais como políticas em linha ou políticas gerenciadas. As políticas em linha são anexadas diretamente a um único usuário, grupo ou perfil. As políticas gerenciadas são políticas autônomas que você pode associar a vários usuários, grupos e funções em seu Conta da AWS. As políticas AWS gerenciadas incluem políticas gerenciadas e políticas gerenciadas pelo cliente. Para saber como escolher entre uma política gerenciada ou uma política em linha, consulte Escolher entre políticas gerenciadas e políticas em linha no Guia do usuário do IAM.

Políticas baseadas em recursos

Políticas baseadas em recurso são documentos de políticas JSON que você anexa a um recurso. São exemplos de políticas baseadas em recursos as políticas de confiança de perfil do IAM e as políticas de bucket do Amazon S3. Em serviços compatíveis com políticas baseadas em recursos, os administradores de serviço podem usá-las para controlar o acesso a um recurso específico. Para o recurso ao qual a política está anexada, a política define quais ações uma entidade principal especificada pode executar nesse recurso e em que condições. Você deve especificar uma entidade principal em uma política baseada em recursos. Os diretores podem incluir contas, usuários, funções, usuários federados ou. Serviços da AWS

Políticas baseadas em recursos são políticas em linha que estão localizadas nesse serviço. Você não pode usar políticas AWS gerenciadas do IAM em uma política baseada em recursos.

Listas de controle de acesso (ACLs)

As listas de controle de acesso (ACLs) controlam quais entidades principais (membros, usuários ou funções da conta) têm permissões para acessar um recurso. As ACLs são semelhantes às políticas baseadas em recursos, embora não usem o formato de documento de política JSON.

O Amazon S3 e o Amazon VPC são exemplos de serviços que oferecem suporte a ACLs. AWS WAF Para saber mais sobre ACLs, consulte Visão geral da lista de controle de acesso (ACL) no Guia do desenvolvedor do Amazon Simple Storage Service.

Outros tipos de política

AWS oferece suporte a tipos de políticas adicionais menos comuns. Esses tipos de política podem definir o máximo de permissões concedidas a você pelos tipos de política mais comuns.

- Limites de permissões: um limite de permissões é um atributo avançado no qual você define o máximo de permissões que uma política baseada em identidade pode conceder a uma entidade do IAM (usuário ou perfil do IAM). É possível definir um limite de permissões para uma entidade. As permissões resultantes são a interseção das políticas baseadas em identidade de uma entidade e dos seus limites de permissões. As políticas baseadas em recurso que especificam o usuário ou a função no campo `Principal` não são limitadas pelo limite de permissões. Uma negação explícita em qualquer uma dessas políticas substitui a permissão. Para mais informações sobre limites de permissões, consulte Limites de permissões para identidades do IAM no Guia do usuário do IAM.
- Políticas de controle de serviço (SCPs) — SCPs são políticas JSON que especificam as permissões máximas para uma organização ou unidade organizacional (OU) em. AWS Organizations AWS Organizations é um serviço para agrupar e gerenciar centralmente vários Contas da AWS que sua empresa possui. Se você habilitar todos os atributos em uma organização, poderá aplicar políticas de controle de serviço (SCPs) a qualquer uma ou a todas as contas. O SCP limita as permissões para entidades nas contas dos membros, incluindo cada uma Usuário raiz da conta da AWS. Para mais informações sobre Organizações e SCPs, consulte Como os SCPs funcionam no AWS Organizations Guia do Usuário.
- Políticas de sessão: são políticas avançadas que você transmite como um parâmetro quando cria de forma programática uma sessão temporária para um perfil ou um usuário federado. As permissões da sessão resultante são a interseção das políticas baseadas em identidade do usuário ou do perfil e das políticas de sessão. As permissões também podem ser provenientes de uma política baseada em recurso. Uma negação explícita em qualquer uma dessas políticas

substitui a permissão. Para mais informações, consulte Políticas de sessão no Guia do usuário do IAM.

Vários tipos de política

Quando vários tipos de política são aplicáveis a uma solicitação, é mais complicado compreender as permissões resultantes. Para saber como AWS determinar se uma solicitação deve ser permitida quando vários tipos de políticas estão envolvidos, consulte Lógica de avaliação de políticas no Guia do usuário do IAM.

Como a criptografia AWS de pagamento funciona com o IAM

Antes de usar o IAM para gerenciar o acesso à criptografia de AWS pagamento, você deve entender quais recursos do IAM estão disponíveis para uso com a criptografia AWS de pagamento. Para ter uma visão geral de como a criptografia de AWS pagamento e outros AWS serviços funcionam com o IAM, consulte AWS Serviços que funcionam com o IAM no Guia do usuário do IAM.

Tópicos

- AWS Criptografia de pagamento Políticas baseadas em identidade
- Autorização baseada em tags do AWS Payment Cryptography

AWS Criptografia de pagamento Políticas baseadas em identidade

Com as políticas baseadas em identidade do IAM, é possível especificar ações ou recursos permitidos ou negados, bem como as condições sob as quais as ações são permitidas ou negadas. AWS A criptografia de pagamento oferece suporte a ações, recursos e chaves de condição específicos. Para conhecer todos os elementos usados em uma política JSON, consulte Referência de elementos de política JSON do IAM no Guia do usuário do IAM.

Ações

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento `Action` de uma política JSON descreve as ações que você pode usar para permitir ou negar acesso em uma política. As ações de política geralmente têm o mesmo nome da operação de

AWS API associada. Existem algumas exceções, como ações somente de permissão, que não têm uma operação de API correspondente. Há também algumas operações que exigem várias ações em uma política. Essas ações adicionais são chamadas de ações dependentes.

Incluem ações em uma política para conceder permissões para executar a operação associada.

As ações de política na criptografia de AWS pagamento usam o seguinte prefixo antes da ação:. `payment-cryptography:` Por exemplo, para conceder permissão a alguém para executar uma operação da `VerifyCardData` API de criptografia de AWS pagamento, você inclui a `payment-cryptography:VerifyCardData` ação na política dessa pessoa. As instruções de política devem incluir um elemento `Action` ou `NotAction`. AWS A criptografia de pagamento define seu próprio conjunto de ações que descrevem as tarefas que você pode realizar com esse serviço.

Para especificar várias ações em uma única instrução, separe-as com vírgulas, como segue:

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

Você também pode especificar várias ações usando caracteres curinga (*). Por exemplo, para especificar todas as ações que começam com a palavra `List` (como `ListKeys` e `ListAliases`), inclua a seguinte ação:

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

Para ver uma lista de ações de criptografia de AWS pagamento, consulte Ações definidas pela criptografia AWS de pagamento no Guia do usuário do IAM.

Recursos

Os administradores podem usar políticas AWS JSON para especificar quem tem acesso ao quê. Ou seja, qual entidade principal pode executar ações em quais recursos, e em que condições.

O elemento de política `Resource` JSON especifica o objeto ou os objetos aos quais a ação se aplica. As instruções devem incluir um elemento `Resource` ou um elemento `NotResource`. Como prática recomendada, especifique um recurso usando seu nome do recurso da Amazon (ARN). Isso pode ser feito para ações que oferecem suporte a um tipo de recurso específico, conhecido como permissões em nível de recurso.

Para ações que não oferecem suporte a permissões em nível de recurso, como operações de listagem, use um asterisco (*) para indicar que a instrução se aplica a todos os recursos.

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

O recurso da chave de Payment Cryptography tem o ARN a seguir:

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

Para obter mais informações sobre o formato dos ARNs, consulte Amazon Resource Names (ARNs) e AWS Service Namespaces.

Por exemplo, para especificar a instância `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h` na instrução, use o seguinte ARN:

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

Para especificar todas as chaves que pertencem a uma conta específica, use o caractere curinga (*):

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

Algumas ações AWS de criptografia de pagamento, como as de criação de chaves, não podem ser executadas em um recurso específico. Nesses casos, você deve utilizar o caractere curinga (*).

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

Para especificar vários recursos em uma única instrução, use uma vírgula como mostrado abaixo:

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

Exemplos

Para ver exemplos de políticas baseadas em identidade de criptografia de AWS pagamento, consulte. AWS Exemplos de políticas baseadas em identidade de criptografia de pagamento

Autorização baseada em tags do AWS Payment Cryptography

AWS Exemplos de políticas baseadas em identidade de criptografia de pagamento

Por padrão, os usuários e funções do IAM não têm permissão para criar ou modificar recursos de AWS Payment Cryptography. Eles também não podem realizar tarefas usando a AWS API AWS Management Console AWS CLI, ou. Um administrador do IAM deve criar políticas do IAM que concedam aos usuários e perfis permissão para executarem operações de API específicas nos recursos especificados de que precisam. O administrador deve anexar essas políticas aos usuários ou grupos do IAM que exigem essas permissões.

Para saber como criar uma política baseada em identidade do IAM usando esses exemplos de documentos de política JSON, consulte Criar políticas na guia JSON no Guia do usuário do IAM.

Tópicos

Melhores práticas de política

As políticas baseadas em identidade determinam se alguém pode criar, acessar ou excluir recursos de criptografia AWS de pagamento em sua conta. Essas ações podem incorrer em custos para a Conta da AWS. Ao criar ou editar políticas baseadas em identidade, siga estas diretrizes e recomendações:

- Comece com as políticas AWS gerenciadas e avance para as permissões de privilégios mínimos — Para começar a conceder permissões aos seus usuários e cargas de trabalho, use as políticas AWS gerenciadas que concedem permissões para muitos casos de uso comuns. Eles estão disponíveis no seu Conta da AWS. Recomendamos que você reduza ainda mais as permissões definindo políticas gerenciadas pelo AWS cliente que sejam específicas para seus casos de uso.

Para mais informações, consulte Políticas gerenciadas pela AWS ou Políticas gerenciadas pela AWS para funções de trabalho no Guia do usuário do IAM.

- Aplique permissões de privilégio mínimo: ao definir permissões com as políticas do IAM, conceda apenas as permissões necessárias para executar uma tarefa. Você faz isso definindo as ações que podem ser executadas em recursos específicos sob condições específicas, também conhecidas como permissões de privilégio mínimo. Para mais informações sobre como usar o IAM para aplicar permissões, consulte Políticas e permissões no IAM no Guia do usuário do IAM.
- Use condições nas políticas do IAM para restringir ainda mais o acesso: você pode adicionar uma condição às políticas para limitar o acesso a ações e recursos. Por exemplo, você pode escrever uma condição de política para especificar que todas as solicitações devem ser enviadas usando SSL. Você também pode usar condições para conceder acesso às ações de serviço se elas forem usadas por meio de uma ação específica AWS service (Serviço da AWS), como AWS CloudFormation. Para obter mais informações, consulte Elementos de política JSON do IAM: Condition no Guia do usuário do IAM.
- Use o IAM Access Analyzer para validar suas políticas do IAM a fim de garantir permissões seguras e funcionais: o IAM Access Analyzer valida as políticas novas e existentes para que elas sigam a linguagem de política do IAM (JSON) e as práticas recomendadas do IAM. O IAM Access Analyzer oferece mais de cem verificações de política e recomendações acionáveis para ajudar você a criar políticas seguras e funcionais. Para mais informações, consulte Validação de políticas do IAM Access Analyzer no Guia do Usuário do IAM.
- Exigir autenticação multifator (MFA) — Se você tiver um cenário que exija usuários do IAM ou um usuário root, ative Conta da AWS a MFA para obter segurança adicional. Para exigir a MFA quando as operações de API forem chamadas, adicione condições de MFA às suas políticas. Para mais informações, consulte Configuração de acesso à API protegido por MFA no Guia do usuário do IAM.

Para mais informações sobre as práticas recomendadas do IAM, consulte Práticas recomendadas de segurança no IAM no Guia do usuário do IAM.

Como usar o console de AWS Payment Cryptography

Para acessar o console AWS de criptografia de pagamento, você deve ter um conjunto mínimo de permissões. Essas permissões devem permitir que você liste e visualize detalhes sobre os recursos AWS de criptografia de pagamento em sua AWS conta. Se você criar uma política baseada em identidade que seja mais restritiva que as permissões mínimas necessárias, o console não funcionará como pretendido para entidades (usuários ou perfis do IAM) com essa política.

Para garantir que essas entidades ainda possam usar o console AWS de criptografia de pagamento, anexe também a seguinte política AWS gerenciada às entidades. Para obter mais informações, consulte Adicionar permissões a um usuário no Guia do usuário do IAM.

Você não precisa permitir permissões mínimas do console para usuários que estão fazendo chamadas somente para a API AWS CLI ou para a AWS API. Em vez disso, permita o acesso somente às ações que correspondem à operação da API que você está tentando executar.

Permitir que os usuários visualizem suas próprias permissões

Este exemplo mostra como você pode criar uma política que permite que os usuários do IAM visualizem as políticas gerenciadas e em linha anexadas a sua identidade de usuário. Essa política inclui permissões para concluir essa ação no console ou programaticamente usando a API AWS CLI ou 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": "*"
        }
    ]
}
```

Capacidade de acessar todos os aspectos da criptografia de AWS pagamento

> **Warning**
>
> Este exemplo fornece permissões amplas e não é recomendado. Em vez disso, considere modelos de acesso menos privilegiados.

Neste exemplo, você quer conceder a um usuário do IAM em sua AWS conta acesso a todas as suas chaves de criptografia de AWS pagamento e a capacidade de chamar todas as APIs de criptografia de AWS pagamento, incluindo ambas ControlPlane e operações. DataPlane

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

Capacidade de chamar APIs usando chaves especificadas

Neste exemplo, você deseja conceder a um usuário do IAM em sua AWS conta acesso a uma de suas chaves de criptografia de AWS pagamento `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h` e, em seguida, usar esse recurso em duas

APIs, e. `GenerateCardData` `VerifyCardData` Por outro lado, o usuário do IAM não terá acesso para usar essa chave em outras operações, como `DeleteKey` ou `ExportKey`

Os recursos podem ser chaves, prefixadas com `key` ou aliases, prefixados com `alias`.

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

Capacidade de negar um recurso específico

> **Warning**
>
> Considere cuidadosamente as implicações da concessão de acesso com caracteres curinga. Em vez disso, considere um modelo de privilégio mínimo.

Neste exemplo, você quer permitir que um usuário do IAM em sua AWS conta acesse qualquer chave de criptografia de AWS pagamento, mas quer negar permissões para uma chave específica. O usuário terá acesso a `VerifyCardData` e `GenerateCardData` com todas as chaves, com exceção da especificada na declaração de negação.

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

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

Solução de problemas AWS de identidade e acesso à criptografia de pagamento

Os tópicos serão adicionados a esta seção à medida que forem identificados problemas relacionados ao IAM que são específicos da criptografia AWS de pagamento. Para obter conteúdo geral de solução de problemas sobre tópicos do IAM, consulte a seção de solução de problemas do Guia do usuário do IAM.

Como monitorar a AWS Payment Cryptography

O monitoramento é uma parte importante para manter a confiabilidade, a disponibilidade e o desempenho da AWS Payment Cryptography e de outras soluções da AWS. A AWS fornece as seguintes ferramentas de monitoramento para observar a AWS Payment Cryptography, relatar quando algo está errado e tomar ações automáticas quando apropriado:

- O Amazon CloudWatch monitora os recursos da AWS e os aplicativos que você executa na AWS em tempo real. É possível coletar e rastrear métricas, criar painéis personalizados e definir alarmes que o notificam ou que realizam ações quando uma métrica especificada atinge um limite definido. Por exemplo, você pode fazer o CloudWatch acompanhar o uso da CPU ou outras métricas das instâncias do Amazon EC2 e iniciar automaticamente novas instâncias quando necessário. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch.
- O Amazon CloudWatch Logs permite monitorar, armazenar e acessar os arquivos de log de instâncias do Amazon EC2, do CloudTrail e de outras fontes. O CloudWatch Logs pode monitorar informações nos arquivos de log e notificar você quando determinados limites forem atingidos. Você também pode arquivar seus dados de log em armazenamento resiliente. Para obter mais informações, consulte o Guia do usuário do Amazon CloudWatch Logs.
- O Amazon EventBridge pode ser usado para automatizar seus serviços da AWS e responder automaticamente a eventos do sistema, como problemas de disponibilidade de aplicações ou alterações de recursos. Os eventos dos serviços da AWS são entregues ao EventBridge quase em tempo real. Você pode escrever regras simples para indicar quais eventos são do seu interesse, e as ações automatizadas a serem tomadas quando um evento corresponder à regra. Para obter mais informações, consulte o Guia do usuário do Amazon EventBridge.
- O AWS CloudTrail captura chamadas de API e eventos relacionados feitos por ou em nome de sua conta da AWS e entrega os arquivos de log a um bucket do Amazon S3 que você especificar. Você pode identificar quais usuários e contas chamaram a AWS, o endereço IP de origem do qual as chamadas foram feitas e quando elas ocorreram. Para obter mais informações, consulte o Guia do usuário do AWS CloudTrail.

> **Note**
>
> Logs AWS CloudTrail são suportados para operações do ambiente de gerenciamento, como CreateKey, mas não para operações do plano de dados, como Generate Card Data (Gerar dados do cartão)

Fazer o log de chamadas da API de AWS Payment Cryptography usando o AWS CloudTrail

A AWS Payment Cryptography está integrada ao AWS CloudTrail, um serviço que fornece um registro das ações realizadas por um usuário, perfil ou um serviço da AWS na AWS Payment Cryptography. O CloudTrail captura todas as chamadas da API de AWS Payment Cryptography como eventos. As chamadas capturadas incluem chamadas do console da AWS Payment Cryptography e chamadas de código para as operações da API de AWS Payment Cryptography. Se você criar uma trilha, poderá habilitar a entrega contínua de eventos do CloudTrail para um bucket do Amazon S3, incluindo eventos para AWS Payment Cryptography. Se você não configurar uma trilha, ainda poderá visualizar os eventos mais recentes no console do CloudTrail em Event history (Histórico de eventos). Usando as informações coletadas pelo CloudTrail, você pode determinar a solicitação que foi feita à AWS Payment Cryptography, o endereço IP do qual a solicitação foi feita, quem fez a solicitação, quando ela foi feita e os detalhes adicionais.

Para saber mais sobre o CloudTrail, consulte o Guia do usuário do AWS CloudTrail.

> **Note**
>
> Atualmente, a integração com o Cloudtrail é suportada somente para operações do ambiente de gerenciamento.

Informações de AWS Payment Cryptography no CloudTrail

O CloudTrail é habilitado em sua conta da AWS quando ela é criada. Quando ocorre atividade na AWS Payment Cryptography, ela é registrada em um evento do CloudTrail, junto com outros eventos de serviço da AWS em Histórico de eventos. Você pode visualizar, pesquisar e baixar eventos recentes em sua conta da AWS. Para obter mais informações, consulte Como visualizar eventos com o histórico de eventos do CloudTrail.

Para um registro contínuo de eventos em sua conta da AWS, incluindo eventos de AWS Payment Cryptography, crie uma trilha. Uma trilha permite que o CloudTrail entregue arquivos de log a um bucket do Amazon S3. Por padrão, quando você cria uma trilha no console, ela é aplicada a todas as regiões da AWS. A trilha registra em log eventos de todas as regiões na partição da AWS e entrega os arquivos de log para o bucket do Amazon S3 especificado por você. Além disso, é possível configurar outros serviços da AWS para analisar mais ainda mais e agir com base nos dados de eventos coletados nos logs do CloudTrail. Para obter mais informações, consulte:

- Visão geral da criação de uma trilha
- Serviços e integrações compatíveis com o CloudTrail
- Configurar notificações do Amazon SNS para o CloudTrail
- Recebimento de arquivos de log do CloudTrail de várias regiões
- Recebimento de arquivos de log do CloudTrail de várias contas

O CloudTrail faz o log de operações de AWS Payment Cryptography, como CreateKey, ImportKey, DeleteKey, ListKeys, TagResource e todas as outras operações do ambiente de gerenciamento.

Cada entrada de log ou evento contém informações sobre quem gerou a solicitação. As informações de identidade ajudam a determinar:

- Se a solicitação foi feita com credenciais de usuário raiz ou do AWS Identity and Access Management (IAM).
- Se a solicitação foi feita com credenciais de segurança temporárias de uma função ou de um usuário federado.
- Se a solicitação foi feita por outro serviço da AWS.

Para obter mais informações, consulte Elemento userIdentity do CloudTrail.

Noções básicas sobre entradas do arquivo de log da AWS Payment Cryptography

Uma trilha é uma configuração que permite a entrega de eventos como arquivos de log a um bucket do Amazon S3 especificado. Os arquivos de log do CloudTrail contêm uma ou mais entradas de log. Um evento representa uma única solicitação de qualquer fonte e inclui informações sobre a ação solicitada, a data e a hora da ação, os parâmetros de solicitação e assim por diante. Os arquivos de log do CloudTrail não são um rastreamento de pilha ordenada de chamadas de API pública. Dessa forma, eles não são exibidos em uma ordem específica.

O exemplo a seguir mostra uma entrada de log do CloudTrail que demonstra a ação `CreateKey` da AWS Payment Cryptography.

```
  {
CloudTrailEvent: {
  tlsDetails= {
```

```
        TlsDetails: {
          cipherSuite=TLS_AES_128_GCM_SHA256,
          tlsVersion=TLSv1.3,
          clientProvidedHostHeader=pdx80.controlplane.paymentcryptography.us-
west-2.amazonaws.com
        }
      },
    requestParameters=CreateKeyInput (
      keyAttributes=KeyAttributes(
        KeyUsage=TR31_B0_BASE_DERIVATION_KEY,
        keyClass=SYMMETRIC_KEY,
        keyAlgorithm=AES_128,
        keyModesOfUse=KeyModesOfUse(
          encrypt=false,
          decrypt=false,
          wrap=false
          unwrap=false,
          generate=false,
          sign=false,
          verify=false,
          deriveKey=true,
          noRestrictions=false)
        ),
      keyCheckValueAlgorithm=null,
      exportable=true,
      enabled=true,
      tags=null),
    eventName=CreateKey,
    userAgent=Coral/Apache-HttpClient5,
    responseElements=CreateKeyOutput(
      key=Key(
        keyArn=arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwsp,
          keyAttributes=KeyAttributes(
            KeyUsage=TR31_B0_BASE_DERIVATION_KEY,
            keyClass=SYMMETRIC_KEY,
            keyAlgorithm=AES_128,
            keyModesOfUse=KeyModesOfUse(
              encrypt=false,
              decrypt=false,
              wrap=false,
              unwrap=false,
              generate=false,
              sign=false,
```

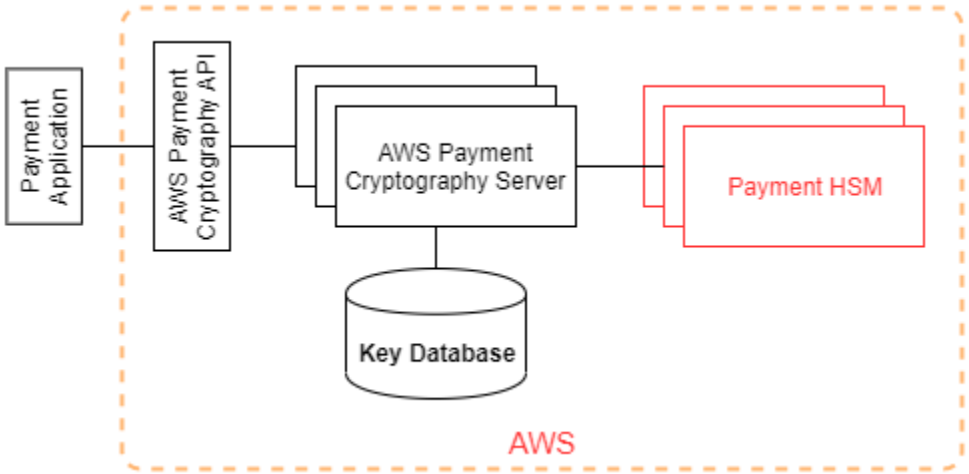
```
              verify=false,
              deriveKey=true,
              noRestrictions=false)
            ),
          keyCheckValue=FE23D3,
          keyCheckValueAlgorithm=ANSI_X9_24,
          enabled=true,
          exportable=true,
          keyState=CREATE_COMPLETE,
          keyOrigin=AWS_PAYMENT_CRYPTOGRAPHY,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

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

Detalhes criptográficos

A AWS Payment Cryptography fornece uma interface da Web para gerar e gerenciar chaves criptográficas para transações de pagamento da AWS. A Payment Cryptography oferece serviços padrão de gerenciamento de chaves, e ferramentas e criptografia de transações de pagamento que você pode usar para gerenciamento e auditoria centralizados. Esta documentação fornece uma descrição detalhada das operações criptográficas que você pode usar na AWS Payment Cryptography para auxiliar você na avaliação dos recursos oferecidos pelo serviço.

A AWS Payment Cryptography contém várias interfaces (incluindo uma API RESTful, por meio de AWS CLI, AWS SDK e AWS Management Console) para solicitar operações criptográficas de uma frota distribuída de módulos de segurança de hardware validados por PCI PTS HSM.



A AWS Payment Cryptography é um serviço hierárquico que consiste em hosts de AWS Payment Cryptography voltados para a Web e uma camada de HSMs. O agrupamento desses hosts hierárquicos forma a pilha de AWS Payment Cryptography. Todas as solicitações para AWS Payment Cryptography devem ser feitas por meio do protocolo Transport Layer Security (TLS) e encerradas em um host de AWS Payment Cryptography. Os hosts de serviços permitem TLS apenas com um pacote de criptografia que fornece perfect forward secrecy. O serviço autentica e autoriza suas solicitações usando os mesmos mecanismos de credenciais e políticas do IAM que estão disponíveis para todas as outras operações de API da AWS.

Os servidores de AWS Payment Cryptography se conectam ao HSM subjacente por meio de uma rede privada e não virtual. As conexões entre os componentes do serviço e o HSM são protegidas com TLS mútuo (mTLS) para autenticação e criptografia.

Objetivos de projeto

A AWS Payment Cryptography foi projetada para atender aos seguintes requisitos:

- Confiável: o uso de chaves é protegido por políticas de controle de acesso que você define e gerencia. Não há mecanismo para exportar chaves da AWS Payment Cryptography em texto simples. A confidencialidade das suas chaves criptográficas é crucial. Vários funcionários da Amazon com acesso específico a controles de acesso baseados em quórum têm a exigência de executar ações administrativas nos HSMs. Nenhum funcionário da Amazon tem acesso às chaves principais (ou mestras) ou backups do HSM. As chaves principais não podem ser sincronizadas com HSMs que não fazem parte de uma região de AWS Payment Cryptography. Todas as outras chaves são protegidas pelas chaves principais do HSM. Portanto, as chaves de AWS Payment Cryptography do cliente não podem ser usadas fora do serviço de AWS Payment Cryptography em operação na conta do cliente.
- Baixa latência e alto throughput: a AWS Payment Cryptography fornece operações criptográficas em nível de latência e throughput adequados para gerenciar chaves criptográficas de pagamento e processar transações de pagamento.
- Durabilidade: a durabilidade das chaves criptográficas foi projetada para ser igual à dos serviços de maior durabilidade na AWS. Uma única chave criptográfica pode ser compartilhada com um terminal de pagamento, cartão com chip EMV ou outro secure cryptographic device (SCD – dispositivo criptográfico seguro) que esteja em uso por muitos anos.
- Regiões independentes: a AWS fornece regiões independentes a clientes que precisam restringir o acesso a dados em diferentes regiões ou precisam atender a requisitos de residência de dados. O uso de chaves pode ser isolado dentro de uma região da AWS.
- Fonte segura de números aleatórios: como a criptografia robusta depende da geração de números aleatórios verdadeiramente imprevisíveis, a AWS Payment Cryptography fornece uma fonte validada e de alta qualidade de números aleatórios. Toda a geração de chaves para a AWS Payment Cryptography usa HSM listado no PCI PTS HSM, operando no modo PCI.
- Auditoria: a AWS Payment Cryptography registra o uso e o gerenciamento de chaves criptográficas em logs do CloudTrail e logs de serviços disponíveis por meio do Amazon CloudWatch. É possível usar logs do CloudTrail para inspecionar o uso de suas chaves criptográficas, incluindo o uso de chaves por contas com as quais você compartilhou chaves. AWS A Payment Cryptography é auditada por avaliadores terceirizados em relação aos padrões aplicáveis de segurança de pagamento PCI, marca do cartão e padrões regionais de segurança de pagamento. Atestados e guias de responsabilidade compartilhada estão disponíveis no AWS Artifact.

- Elasticidade: a AWS Payment Cryptography aumenta a escala horizontalmente e de acordo com sua demanda. Em vez de prever e reservar a capacidade do HSM, a AWS Payment Cryptography fornece criptografia de pagamentos sob demanda. AWS A Payment Cryptography assume a responsabilidade de manter a segurança e a conformidade do HSM para fornecer capacidade suficiente para atender aos picos de demanda do cliente.

Fundamentos

Os tópicos deste capítulo descrevem as primitivas criptográficas da criptografia de AWS pagamento e onde elas são usadas. Eles também apresentam os elementos básicos do serviço.

Tópicos

Primitivas criptográficas

AWS A criptografia de pagamento usa algoritmos criptográficos padrão e parametrizáveis para que os aplicativos possam implementar os algoritmos necessários para seu caso de uso. O conjunto de algoritmos criptográficos é definido pelos padrões PCI, ANSI X9, EMVCo e ISO. Toda a criptografia é executada por HSMs listados no padrão PCI PTS HSM em execução no modo PCI.

Entropia e geração de números aleatórios

AWS A geração da chave de criptografia de pagamento é realizada nos HSMs AWS de criptografia de pagamento. Os HSMs implementam um gerador de números aleatórios que atende aos requisitos do PCI PTS HSM para todos os tipos e parâmetros de chave compatíveis.

Operações de chave simétrica

Algoritmos de chave simétrica e pontos fortes definidos em ANSI X9 TR 31, ANSI X9.24 e Anexo C do PCI PIN são compatíveis:

- Funções de hash: algoritmos da família SHA2 e SHA3 com tamanho de saída maior que 2551. Exceto pela compatibilidade retroativa com terminais pré-PCI PTS POI v3.
- Criptografia e descriptografia: AES com tamanho de chave maior ou igual a 128 bits ou TDEA com tamanho de chave maior ou igual a 112 bits (2 chaves ou 3 chaves).
- Códigos de autenticação de mensagens (MACs) CMAC ou GMAC com AES, bem como HMAC com uma função hash aprovada e um tamanho de chave maior ou igual a 128.

AWS A criptografia de pagamento usa AES 256 para chaves principais do HSM, chaves de proteção de dados e chaves de sessão TLS.

Operações de chave assimétrica

Algoritmos de chave assimétrica e pontos fortes de chave definidos em ANSI X9 TR 31, ANSI X9.24 e Anexo C do PCI PIN são compatíveis:

- Esquemas de estabelecimento de chaves aprovados: conforme descrito em NIST SP800-56A (acordo de chave baseado em ECC/FCC2), NIST SP800-56B (acordo de chave baseado em IFC) e NIST SP800-38F (criptografia/encapsulamento de chave baseado em AES).

AWS Os hosts de criptografia de pagamento só permitem conexões ao serviço usando TLS com um pacote de criptografia que fornece sigilo direto perfeito.

Armazenamento de chaves

AWS As chaves de criptografia de pagamento são protegidas pelas chaves principais HSM AES 256 e armazenadas em blocos de chaves ANSI X9 TR 31 em um banco de dados criptografado. O banco de dados é replicado para um banco de dados na memória em servidores de criptografia AWS de pagamento.

De acordo com o Anexo C do Regulamento de segurança PCI PIN, as chaves AES 256 são tão ou mais fortes que:

- TDEA de 3 chaves

- RSA de 15360 bits
- ECC de 512 bits
- DSA, DH e MQV 15360/512

Importar chaves usando chaves simétricas

AWS A criptografia de pagamento suporta a importação de criptogramas e blocos de chaves com chaves simétricas ou públicas com uma chave de criptografia de chave simétrica (KEK) tão forte ou mais forte que a chave protegida para importação.

Importar chaves usando chaves assimétricas

AWS A criptografia de pagamento suporta a importação de criptogramas e blocos de chaves com chaves simétricas ou públicas protegidas por uma chave de criptografia de chave privada (KEK) tão forte ou mais forte que a chave protegida para importação. A chave pública fornecida para a decodificação deve ter sua autenticidade e integridade garantidas por um certificado de uma autoridade em que o cliente confie.

Os KEK públicos fornecidos pela AWS Payment Cryptography têm a autenticação e a proteção de integridade de uma autoridade de certificação (CA) com conformidade atestada com o PCI PIN Security e o PCI P2PE Anexo A.

Exportação de chaves

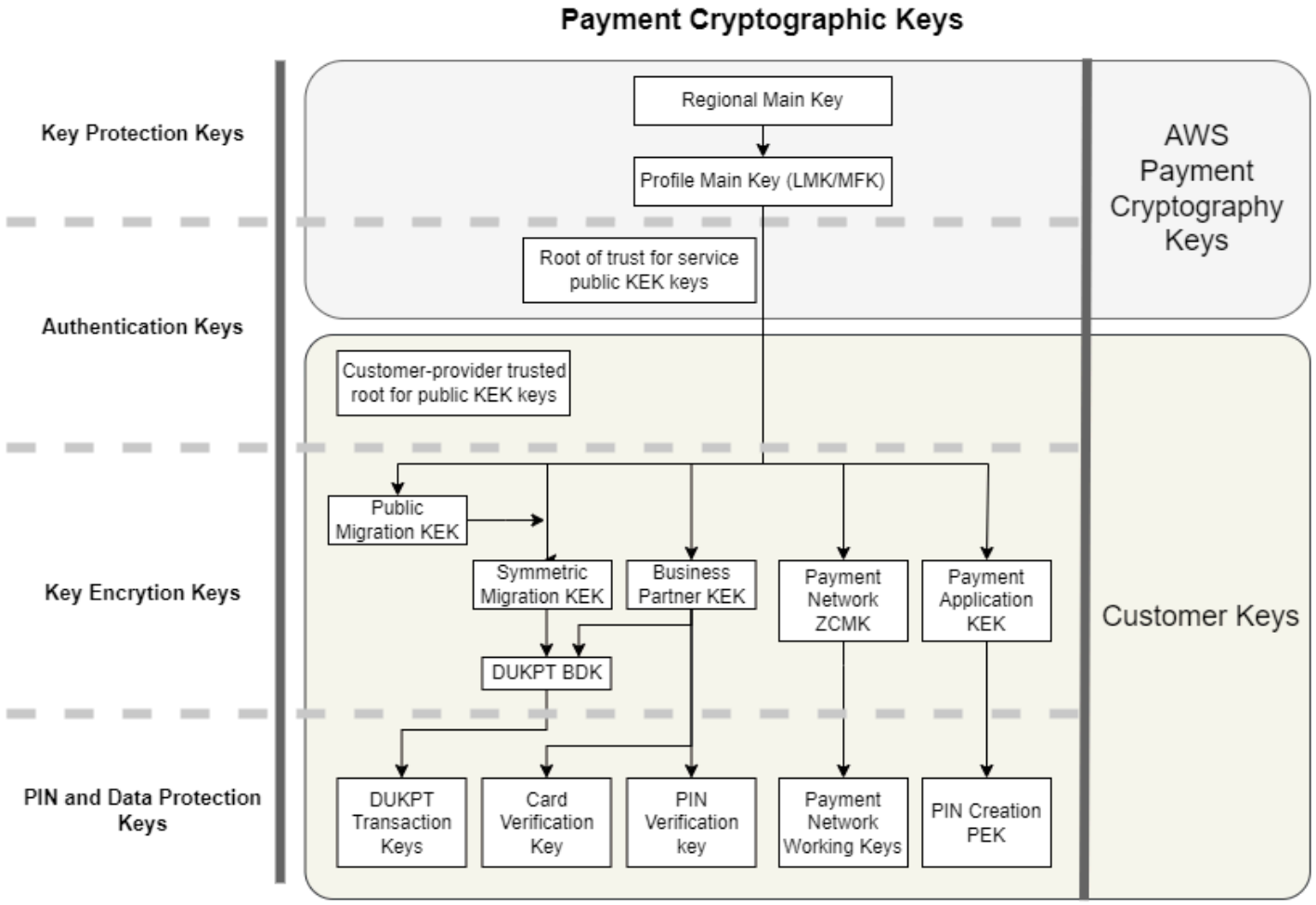
As chaves podem ser exportadas e protegidas por chaves com as chaves apropriadas KeyUsage e que sejam tão fortes ou mais fortes do que a chave a ser exportada.

Protocolo de chave única derivada por transação (DUKPT)

AWS A criptografia de pagamento é compatível com chaves de derivação base (BDK) TDEA e AES, conforme descrito pelo ANSI X9.24-3.

Hierarquia de chaves

A hierarquia de chaves de criptografia de AWS pagamento garante que as chaves sejam sempre protegidas por chaves tão fortes ou mais fortes do que as chaves que elas protegem.



AWS As chaves de criptografia de pagamento são usadas para proteção de chaves dentro do serviço:

Chave	Descrição
Chave principal regional	Protege imagens ou perfis virtuais do HSM usados para processamento criptográfico. Essa chave existe somente no HSM e nos backups seguros.
Chave principal do perfil	Chave de proteção de chave de cliente de alto nível, tradicionalmente chamada de chave mestra local (LMK) ou chave de arquivo mestre (MFK) para chaves de cliente. Essa chave existe somente no HSM e nos backups

Chave	Descrição
	seguros. Os perfis definem configurações de HSM distintas, conforme exigido pelos padrões de segurança para casos de uso de pagamento s.
Raiz de confiança nas chaves de chave de criptografia de chave pública (KEK) da criptografia de AWS pagamento	A chave pública raiz confiável e o certifica do para autenticar e validar chaves públicas fornecidos pela AWS Payment Cryptogra phy para importação e exportação de chaves usando chaves assimétricas.

As chaves do cliente são agrupadas por chaves usadas para proteger outras chaves e chaves que protegem dados relacionados a pagamentos. Estes são exemplos de chaves de cliente dos dois tipos:

Chave	Descrição
Raiz confiável fornecida pelo cliente para chaves KEK públicas	Chave pública e certificado fornecidos por você como raiz de confiança para autenticar e validar chaves públicas que você fornece para importação e exportação de chaves usando chaves assimétricas.
Chaves de criptografia de chaves (KEK)	As KEK são usadas exclusivamente para criptografar outras chaves para troca entre armazenamentos externos de chaves e criptografia de AWS pagamento, parceiros de negócios, redes de pagamento ou aplicativos diferentes em sua organização.
Chave de derivação de base (BDK) da chave única derivada por transação (DUKPT)	As BDKs são usadas para criar chaves exclusivas para cada terminal de pagamento e traduzir transações de vários terminais em uma única chave de trabalho do banco adquirent e ou do adquirente. A prática recomendada,

Chave	Descrição
	exigida pela criptografia ponto a ponto PCI (P2PE), é que diferentes BDKs sejam usadas para diferentes modelos de terminal, injeção de chave ou serviços de inicialização, ou outra segmentação para limitar o impacto do comprometimento de uma BDK.
Chave mestra de controle de zona da rede de pagamento (ZCMK)	As ZCMK, também conhecidas como chaves de zona ou chaves mestras de zona, são fornecidas pelas redes de pagamento para estabelecer as chaves de trabalho iniciais.
Chaves de transação DUKPT	Os terminais de pagamento configurados para DUKPT derivam uma chave única para o terminal e a transação. O HSM que recebe a transação pode determinar a chave a partir do identificador do terminal e do número da sequência da transação.
Chaves de preparação de dados do cartão	As chaves mestras do emissor EMV, as chaves do cartão EMV e os valores de verificação e as chaves de proteção de arquivos de dados de personalização do cartão são usadas para criar dados para cartões individuais para uso por um provedor de personalização de cartões. Essas chaves e dados de validação criptográfica também são usados pelos bancos emissores ou emissores para autenticar os dados do cartão como parte da autorização de transaçõe s.

Chave	Descrição
Chaves de preparação de dados do cartão	As chaves mestras do emissor EMV, as chaves do cartão EMV e os valores de verificação e as chaves de proteção de arquivos de dados de personalização do cartão são usadas para criar dados para cartões individuais para uso por um provedor de personalização de cartões. Essas chaves e dados de validação criptográfica também são usados pelos bancos emissores ou emissores para autenticar os dados do cartão como parte da autorização de transaçõe s.
Chaves de funcionamento da rede de pagamento	Frequentemente chamadas de chave de trabalho do emissor ou chave de trabalho do adquirente, essas são as chaves que criptogra fam as transações enviadas ou recebidas das redes de pagamento. Essas chaves são alternadas com frequência pela rede, geralmente a cada dia ou de hora em hora. Estas são chaves de criptografia de PINs (PEK) para transações com PIN/débito.
Chaves de criptografia (PEK) do número de identificação pessoal (PIN)	Os aplicativos que criam ou descriptografam blocos de PIN usam PEKs para impedir o armazenamento ou a transmissão de PIN em texto não criptografado.

Operações internas

Este tópico descreve os requisitos internos implementados pelo serviço para proteger as chaves do cliente e as operações criptográficas para um serviço de gerenciamento de chaves e criptografia de pagamento escalável e distribuído globalmente.

Especificações e ciclo de vida do HSM

AWS A criptografia de pagamento usa uma frota de HSM disponíveis comercialmente. Os HSMs são validados pelo FIPS 140-2 Nível 3 e também usam versões de firmware e a política de segurança listada na lista de dispositivos PCI PTS aprovados pelo PCI Security Standards Council como uma reclamação PCI HSM v3. O padrão PCI PTS HSM inclui requisitos adicionais para fabricação, envio, implantação, gerenciamento e destruição do hardware HSM, que são importantes para a segurança e conformidade dos pagamentos, mas não são abordados pelo FIPS 140.

Todos os HSMs são operados no modo PCI e configurados com a política de segurança PCI PTS HSM. Somente as funções necessárias para oferecer suporte aos casos de uso da criptografia de AWS pagamento estão ativadas. AWS A criptografia de pagamento não permite a impressão, exibição ou devolução de PINs de texto não criptografado.

Segurança física do dispositivo HSM

Somente HSMs que tenham chaves de dispositivo assinadas por uma autoridade certificadora de criptografia de AWS pagamento (CA) pelo fabricante antes da entrega podem ser usados pelo serviço. A criptografia AWS de pagamento é uma subCA da CA do fabricante que é a raiz da confiança dos certificados de fabricantes e dispositivos do HSM. A CA do fabricante implementa o ANSI TR 34 e atestou a conformidade com o Anexo A de Segurança PCI PIN e o Anexo A. O fabricante verifica se todos os HSM com chaves de dispositivo assinadas pela Autoridade de Criptografia de AWS Pagamentos são enviados para o destinatário designado pela AWS.

Conforme exigido pela segurança PCI PIN, o fabricante fornece uma lista de números de série por meio de um canal de comunicação diferente do da remessa do HSM. Esses números de série são verificados em cada etapa do processo de instalação do HSM nos datacenters da AWS. Por fim, os operadores de criptografia de AWS pagamento validam a lista de HSM instalados em relação à lista do fabricante antes de adicionar o número de série à lista de HSM autorizados a receber AWS chaves de criptografia de pagamento.

Os HSMs estão sempre em armazenamento seguro ou sob controle duplo, o que inclui:

- Remessa do fabricante para uma instalação de montagem de rack da AWS.
- Durante a montagem do rack.
- Envio da instalação de montagem do rack para um datacenter.

- Recebimento e instalação em uma sala de processamento segura do datacenter. Os racks HSM impõem controle duplo com fechaduras controladas por cartão, sensores de porta com alarme e câmeras.
- Durante as operações.
- Durante o descomissionamento e a destruição.

Um completo chain-of-custody, com responsabilidade individual, é mantido e monitorado para cada HSM.

Inicialização do Java

Um HSM só é inicializado como parte da frota de criptografia de AWS pagamento depois que sua identidade e integridade são validadas por números de série, chaves de dispositivo instaladas pelo fabricante e soma de verificação do firmware. Depois que a autenticidade e a integridade de um HSM são validadas, ele é configurado, incluindo a ativação do Modo PCI. Em seguida, as chaves principais da região de criptografia de AWS pagamento e as chaves principais do perfil são estabelecidas e o HSM fica disponível para o serviço.

Serviço e reparo do HSM

O HSM tem componentes que podem ser reparados e não exigem a violação do limite criptográfico do dispositivo. Esses componentes incluem ventiladores de resfriamento, fontes de alimentação e baterias. Se um HSM ou outro dispositivo dentro do rack do HSM precisar de manutenção, o controle duplo será mantido durante todo o período em que o rack estiver aberto.

Descomissionamento do HSM

O descomissionamento ocorre devido end-of-life ou falha de um HSM. Os HSM são logicamente zerados antes de serem removidos do rack e, se funcionarem, são destruídos nas salas de processamento seguras dos datacenters da AWS. Eles nunca são devolvidos ao fabricante para reparo, usados para outra finalidade ou removidos de uma sala de processamento segura antes da destruição.

Atualização de firmware do HSM

As atualizações de firmware do HSM são aplicadas quando necessário para manter o alinhamento com as versões listadas do PCI PTS HSM e do FIPS 140-2 (ou FIPS 140-3), se uma atualização estiver relacionada à segurança ou se for determinado que os clientes podem se beneficiar dos

recursos de uma nova versão. AWS Os HSMs de criptografia de pagamento executam off-the-shelf firmware, correspondendo às versões listadas no PCI PTS HSM. As novas versões de firmware são validadas quanto à integridade com as versões de firmware com certificação PCI ou FIPS e testadas quanto à funcionalidade antes da implantação em todos os HSMs.

Acesso do operador

Os operadores podem ter acesso sem console ao HSM para solução de problemas em casos raros em que as informações coletadas do HSM durante as operações normais são insuficientes para identificar um problema ou planejar uma mudança. As etapas a seguir são executadas:

- As atividades de solução de problemas são desenvolvidas e aprovadas e a sessão sem console é agendada.
- Um HSM é removido do serviço de processamento do cliente.
- As chaves principais são excluídas, sob controle duplo.
- O operador tem permissão para acessar o HSM sem console para realizar atividades de solução de problemas aprovadas, sob controle duplo.
  - Após o término da sessão sem console, o processo de provisionamento inicial é executado no HSM, retornando o firmware e a configuração padrão e, em seguida, sincronizando a chave principal, antes de devolver o HSM para atender aos clientes.
  - Os registros da sessão são registrados no controle de alterações.
  - As informações obtidas na sessão são usadas para planejar mudanças futuras.

Todos os registros de acesso que não sejam do console são revisados quanto à conformidade do processo e possíveis alterações no monitoramento do HSM, no processo non-console-access de gerenciamento ou no treinamento do operador.

Gerenciamento de chaves

Todos os HSMs em uma região são sincronizados com uma chave principal da região. Uma chave principal de região protege pelo menos uma chave principal de perfil. Uma chave principal de perfil protege as chaves do cliente.

Todas as chaves principais são geradas por um HSM e distribuídas por distribuição simétrica de chaves usando técnicas assimétricas, alinhadas com ANSI X9 TR 34 e Anexo A do PCI PIN.

Tópicos

Geração

Chaves principais AES de 256 bits são geradas em um dos HSM provisionados para a frota de HSM de serviço, usando o gerador de números aleatórios PCI PTS HSM.

Sincronização da chave principal da região

As chaves principais da região do HSM são sincronizadas pelo serviço em toda a frota regional com mecanismos definidos pela ANSI X9 TR-34, que incluem:

- Autenticação mútua usando chaves e certificados do host de distribuição de chaves (KDH) e do dispositivo receptor de chaves (KRD) para fornecer autenticação e integridade de chaves públicas.
- Os certificados são assinados por uma autoridade de certificação (CA) que atende aos requisitos do Anexo A2 do PCI PIN, exceto para algoritmos assimétricos e pontos fortes de chave apropriados para proteger chaves AES de 256 bits.
- Identificação e proteção de chaves para chaves simétricas distribuídas consistentes com ANSI X9 TR-34 e Anexo A1 do PCI PIN, exceto para algoritmos assimétricos e pontos fortes de chave apropriados para proteger chaves AES de 256 bits.

As chaves principais de região são estabelecidas para HSMs que foram autenticados e provisionados para uma região por meio de:

- Uma chave principal é gerada em um HSM na região. Esse HSM é designado como o host de distribuição de chaves.

- Todos os HSMs provisionados na região geram o token de autenticação KRD, que contém a chave pública do HSM e informações de autenticação não reproduzíveis.
- Os tokens KRD são adicionados à lista de permissões do KDH depois que o KDH valida a identidade e a permissão do HSM para receber as chaves.
- O KDH produz um token de chave principal autenticável para cada HSM. Os tokens contêm informações de autenticação KDH e uma chave principal criptografada que pode ser carregada somente em um HSM para o qual ela foi criada.
- Cada HSM recebe o token de chave principal criado para ele. Depois de validar as informações de autenticação do próprio HSM e as informações de autenticação do KDH, a chave principal é descriptografada pela chave privada do KRD e carregada na chave principal.

Caso um único HSM precise ser sincronizado novamente com uma região:

- Ele é revalidado e provisionado com firmware e configuração.
- Se for novo na região:
  - O HSM gera um token de autenticação KRD.
  - O KDH adiciona o token à sua lista de permissões.
  - O KDH gera um token de chave principal para o HSM.
  - O HSM carrega a chave principal.
  - O HSM é disponibilizado para o serviço.

Isso garante que:

- Somente o HSM validado para processamento AWS de criptografia de pagamento em uma região pode receber a chave mestra dessa região.
- Somente uma chave mestra de um HSM de criptografia de AWS pagamento pode ser distribuída para um HSM da frota.

Alternância de chaves principais da região

As chaves principais da região são alternadas ao fim do período criptográfico, no caso improvável de uma suspeita de comprometimento da chave ou após alterações no serviço que possam afetar a segurança da chave.

Uma nova chave principal de região é gerada e distribuída, como no provisionamento inicial. As chaves principais do perfil salvas devem ser traduzidas para a nova chave principal da região.

A alternância da chave principal da região não afeta o processamento do cliente.

Sincronização da chave principal do perfil

As chaves principais do perfil são protegidas pelas chaves principais da região. Isso restringe um perfil a uma região específica.

As chaves principais do perfil são provisionadas adequadamente:

- Uma chave principal de perfil é gerada em um HSM que tem a chave principal da região sincronizada.
- A chave principal do perfil é armazenada e criptografada com a configuração do perfil e outros contextos.
- O perfil é usado para funções criptográficas do cliente por qualquer HSM na região com a chave principal da região.

Alternância de chaves principais do perfil

As chaves principais do perfil são alternadas ao fim do período criptográfico, após suspeita de comprometimento da chave ou após alterações no serviço que possam afetar a segurança da chave.

Etapas de alternância:

- Uma nova chave principal de perfil é gerada e distribuída como uma chave principal pendente, assim como no provisionamento inicial.
- Um processo em segundo plano converte o material de chave do cliente da chave principal do perfil estabelecido para a chave principal pendente.
- Quando todas as chaves do cliente tiverem sido criptografadas com a chave pendente, ela será promovida à chave principal do perfil.
- Um processo em segundo plano exclui o material de chave do cliente protegido pela chave expirada.

A alternância da chave principal do perfil não afeta o processamento do cliente.

Proteção

As chaves dependem somente da hierarquia de chaves para a proteção. A proteção das chaves principais é fundamental para evitar a perda ou o comprometimento de todas as chaves do cliente.

As chaves principais da região podem ser restauradas somente do backup para o HSM autenticado e provisionado para o serviço. Essas chaves podem ser armazenadas apenas como tokens de chave principal criptografados e mutuamente autenticáveis de um KDH específico para um HSM específico.

As chaves mestras do perfil são armazenadas com a configuração do perfil e as informações de contexto criptografadas por região.

As chaves do cliente são armazenadas em blocos de chaves, protegidas por uma chave mestra de perfil.

Todas as chaves existem exclusivamente em um HSM ou são armazenadas protegidas por outra chave de força criptográfica igual ou mais forte.

Durabilidade

As chaves do cliente para criptografia de transações e funções de negócios devem estar disponíveis mesmo em situações extremas que normalmente causariam interrupções. AWS A criptografia de pagamento utiliza um modelo de redundância de vários níveis em zonas e regiões de disponibilidade. AWS Os clientes que precisam de maior disponibilidade e durabilidade para operações criptográficas de pagamento do que as fornecidas pelo serviço devem implementar arquiteturas multirregionais.

A autenticação do HSM e os tokens da chave principal são salvos e podem ser usados para restaurar uma chave principal ou sincronizar com uma nova chave principal, caso um HSM precise ser redefinido. Os tokens são arquivados e usados somente sob controle duplo quando necessário.

Segurança de comunicação

Externo

AWS Os endpoints da API de criptografia de pagamento atendem aos padrões de AWS segurança, incluindo TLS na versão 1.2 ou superior e Signature versão 4 para autenticação e integridade das solicitações.

As conexões TLS de entrada são encerradas em balanceadores de carga de rede e encaminhadas para manipuladores de API por meio de conexões TLS internas.

Interno

As comunicações internas entre os componentes do serviço e entre os componentes do serviço e outros serviços da AWS são protegidas por TLS usando criptografia robusta.

Os HSM estão em uma rede privada, não virtual, que pode ser acessada apenas por componentes de serviço. Todas as conexões entre o HSM e os componentes do serviço são protegidas com TLS mútuo (mTLS), igual ou superior ao TLS 1.2. Os certificados internos para TLS e mTLS são gerenciados pelo Amazon Certificate Manager usando uma AWS Private Certificate Authority. As VPCs internas e a rede HSM são monitoradas em busca de atividades e alterações de configuração inesperadas.

Gerenciamento de chaves de clientes

Na AWS, a confiança do cliente é nossa maior prioridade. Você mantém o controle total das chaves que você carrega ou cria no serviço sob sua conta da AWS e é responsável pela configuração do acesso às chaves.

AWS A criptografia de pagamento tem total responsabilidade pela conformidade física do HSM e pelo gerenciamento de chaves das chaves gerenciadas pelo serviço. Isso exige a propriedade e o gerenciamento das chaves principais do HSM e o armazenamento das chaves protegidas do cliente no banco de dados de chaves AWS de criptografia de pagamento.

Separação de espaço entre chaves do cliente

AWS A criptografia de pagamento aplica as principais políticas para todos os usos de chaves, incluindo a limitação dos diretores à conta proprietária da chave, a menos que uma chave seja explicitamente compartilhada com outra conta.

Backup e recuperação

O backup das chaves e das principais informações de uma região é feito em arquivos criptografados pela AWS. Os arquivos exigem controle duplo AWS para serem restaurados.

Blocos de chaves

Todas as chaves são armazenadas em blocos de chaves no formato ANSI X9 TR-31.

As chaves podem ser importadas para o serviço a partir de criptogramas ou outros formatos de bloco de chaves suportados pelo ImportKey. Da mesma forma, as chaves podem ser exportadas, se forem exportáveis, para outros formatos de blocos de chaves ou criptogramas compatíveis com os perfis de exportação de chaves.

Uso de chaves

O uso da chave é restrito ao configurado KeyUsage pelo serviço. O serviço falhará em qualquer solicitação com uso inadequado de chaves, modo de uso ou algoritmo para a operação criptográfica solicitada.

Relações de troca de chaves

O PCI PIN Security e o PCI P2PE exigem que as organizações que compartilham chaves que criptografam PINs, incluindo a KEK usada para compartilhar essas chaves, não compartilhem essas chaves com nenhuma outra organização. É uma prática recomendada que chaves simétricas sejam compartilhadas apenas entre duas partes, inclusive dentro da mesma organização. Isso minimiza o impacto de suspeitas de comprometimento de chaves que forcem a substituição das chaves afetadas.

Mesmo os casos de negócios que exigem o compartilhamento de chaves entre mais de duas partes devem manter um número mínimo de partes.

AWS A criptografia de pagamento fornece etiquetas-chave que podem ser usadas para rastrear e impor o uso de chaves dentro desses requisitos.

Por exemplo, KEK e BDK para diferentes instalações de injeção de chave podem ser identificadas definindo um “KIF”=“POSStation” para todas as chaves compartilhadas com esse provedor de serviços. Outro exemplo seria marcar chaves compartilhadas com redes de pagamento com “Rede” = “PayCard”. As tags permitem que você crie controles de acesso e crie relatórios de auditoria para aplicar e demonstrar suas principais práticas de gerenciamento.

Exclusão de chaves

DeleteKey marca as chaves no banco de dados para exclusão após um período configurável pelo cliente. Após esse período, a chave é excluída irreversivelmente. Esse é um mecanismo de segurança para evitar a exclusão acidental ou maliciosa de uma chave. As teclas marcadas para exclusão não estão disponíveis para nenhuma ação, exceto RestoreKey.

As chaves excluídas permanecem nos backups do serviço por sete dias após a exclusão. Elas não são restauráveis durante esse período.

As chaves pertencentes a contas fechadas da AWS são marcadas para exclusão. Se a conta for reativada antes que o período de exclusão seja atingido, todas as chaves marcadas para exclusão serão restauradas, mas desativadas. Elas devem ser reativadas por você para serem usadas em operações criptográficas.

Compartilhamento de chaves

Chaves podem ser compartilhadas com outras contas dentro ou fora da sua organização usando o AWS Resource Access Manager (https://docs.aws.amazon.com/ARG/index.html). As chaves podem ser agrupadas em um compartilhamento de recursos e depois compartilhadas com uma conta ou com usuários e funções específicos do IAM em uma conta. Você especifica as permissões de uso para cada compartilhamento de recursos. As permissões de compartilhamento são restringidas por uma política de recursos de chaves. Uma chave compartilhada não permitirá uma ação restrita por sua própria política. A permissão de compartilhamento pode ser retirada a qualquer momento.

Registrar em log e monitoramento

Os registros de serviços internos incluem:

- CloudTrail registros de chamadas de serviço da AWS feitas pelo serviço
- CloudWatch registros de ambos os eventos registrados diretamente nos CloudWatch registros ou eventos do HSM
- Arquivos de log do HSM e dos sistemas de serviço
- Arquivos de log

Todas as fontes de log monitoram e filtram informações confidenciais, inclusive sobre chaves. Os logs são revisados sistematicamente para garantir que não contenham informações confidenciais do cliente.

O acesso aos logs é restrito às pessoas necessárias para concluir as funções de trabalho.

Todos os logs são retidos de acordo com as políticas de retenção de registros da AWS.

Operações do cliente

AWS A criptografia de pagamento tem total responsabilidade pela conformidade física do HSM de acordo com os padrões PCI. O serviço também fornece um armazenamento seguro de chaves e garante que as chaves possam ser usadas apenas para os fins permitidos pelos padrões PCI e especificados por você durante a criação ou importação. Você é responsável por configurar os principais atributos e acesso para aproveitar os recursos de segurança e conformidade do serviço.

Tópicos

- Importar chaves
- Exportar chaves
- Excluir chaves
- Alternar chaves do

Gerar chaves

Ao criar chaves, você define os atributos que o serviço usa para impor o uso compatível da chave:

- Algoritmo e comprimento da chave
- Uso
- Disponibilidade e validade

Tags usadas para o controle de acesso por atributo (ABAC) são usadas para limitar as chaves para uso com parceiros ou aplicativos específicos e também devem ser definidas durante a criação. Inclua políticas para limitar as funções permitidas para excluir ou alterar tags.

Você deve garantir que as políticas que determinam as funções que podem usar e gerenciar a chave sejam definidas antes da criação da chave.

> **Note**
>
> As políticas do IAM nos CreateKey comandos podem ser usadas para impor e demonstrar o controle duplo para a geração de chaves.

Importar chaves

Ao importar chaves, os atributos para impor o uso compatível da chave são definidos pelo serviço usando as informações vinculadas criptograficamente no bloco de chaves. O mecanismo para definir o contexto de chave fundamental é usar blocos de chave criados com o HSM de origem e protegidos por uma KEK compartilhada ou assimétrica. Isso se alinha aos requisitos do PCI PIN e preserva o uso, o algoritmo e a força da chave do aplicativo de origem.

Atributos importantes, tags e políticas de controle de acesso devem ser estabelecidos na importação, além das informações no bloco de chaves.

A importação de chaves usando criptogramas não transfere atributos de chave do aplicativo de origem. Você deve definir os atributos adequadamente usando esse mecanismo.

Frequentemente, as chaves são trocadas usando componentes de texto não criptografado, transmitidas pelos guardiões das chaves e, em seguida, carregadas com uma cerimônia que implementa o controle duplo em uma sala segura. Isso não é diretamente suportado pela criptografia AWS de pagamento. A API exportará uma chave pública com um certificado que pode ser importado pelo seu próprio HSM para exportar um bloco de chaves que pode ser importado pelo serviço. Isso permite o uso de seu próprio HSM para carregar componentes de texto não criptografado.

Você deve usar valores de verificação de chave (KCV) para verificar se as chaves importadas correspondem às chaves de origem.

As políticas do IAM na ImportKey API podem ser usadas para impor e demonstrar o controle duplo para a importação de chaves.

Exportar chaves

O compartilhamento de chaves com parceiros ou aplicativos on-premises pode exigir a exportação de chaves. O uso de blocos de chaves para exportações mantém o contexto fundamental da chave com o material de chave criptografado.

Tags de chave podem ser usadas para limitar a exportação de chaves para KEKs que compartilham a mesma tag e valor.

AWS A criptografia de pagamento não fornece nem exibe os principais componentes em texto não criptografado. Isso requer acesso direto dos principais guardiões aos dispositivos criptográficos seguros (SCD) testados pelo PCI PTS HSM ou ISO 13491 para exibição ou impressão. Você pode estabelecer uma KEK assimétrica ou simétrica com seu SCD para conduzir a cerimônia de criação de componentes de chave de texto não criptografado sob controle duplo.

Os valores de verificação de chave (KCV) devem ser usados para verificar se as chaves de origem importadas pelo HSM de destino correspondem às chaves de origem.

Excluir chaves

É possível usar a API DeleteKey para programar a exclusão das chaves após um período configurado por você. Antes disso, as chaves podem ser recuperadas. Depois que as chaves são excluídas, elas são removidas do serviço permanentemente.

As políticas do IAM na DeleteKey API podem ser usadas para impor e demonstrar o controle duplo para a exclusão de chaves.

Alternar chaves do

O efeito da alternância da chave pode ser implementado usando o alias da chave criando ou importando uma nova chave e modificando o alias da chave para se referir à nova chave. A chave antiga seria excluída ou desativada, dependendo de suas práticas de gerenciamento.

Cotas para AWS Payment Cryptography

Sua conta da AWS tem cotas padrão, anteriormente chamadas de limites, para cada serviço da AWS. A menos que especificado de outra forma, cada cota é específica da região. Você pode solicitar o aumento de algumas cotas, porém, algumas delas não podem ser aumentadas.

Nome	Padrão	Ajustá	Descrição
Aliases	Cada região com suporte: 2.000	Sim	O número máximo de aliases que você pode ter nesta conta na região atual.
Taxa combinada de solicitações para ambientes de gerenciamento	Cada região compatível: 5 por segundo	Sim	O número máximo de solicitações para ambientes de gerenciamento por segundo que você pode fazer nesta conta na região atual. Essa cota se aplica a todas as operações combinadas do ambiente de gerenciamento.
Taxa combinada de solicitações de planos de dados (assimétrica)	Cada região compatível: 20 por segundo	Sim	O número máximo de solicitações por segundo para operações de plano de dados com uma chave assimétrica que você pode fazer nessa conta na região atual. Essa cota se aplica a todas as operações do plano de dados combinadas.

Nome	Padrão	Ajustá	Descrição
Taxa combinada de solicitações de planos de dados (simétrica)	Cada região compatível: 500 por segundo	Sim	O número máximo de solicitações por segundo para operações de plano de dados com uma chave simétrica que você pode fazer nessa conta na região atual. Essa cota se aplica a todas as operações do plano de dados combinadas.
Chaves	Cada região compatível: 2.000	Sim	O número máximo de chaves que você pode ter nesta conta na região atual, excluindo chaves excluídas.

Histórico do documento para o Guia do usuário da AWS Payment Cryptography

A tabela a seguir descreve as versões da documentação para AWS Payment Cryptography.

Alteração	Descrição	Data
Lançamento de recursos	Informações adicionadas sobre novos recursos relacionados à importação/exportação de chaves usando RSA e exportação de chaves DUKPT IPEK/IK.	15 de janeiro de 2024
Lançamento inicial	Versão inicial do Guia do usuário da AWS Payment Cryptography	8 de junho de 2023

As traduções são geradas por tradução automática. Em caso de conflito entre o conteúdo da tradução e da versão original em inglês, a versão em inglês prevalecerá.