Guide de l’utilisateur

AWS Cryptographie des paiements

AWS Cryptographie des paiements: Guide de l’utilisateur

Table of Contents

Qu'est-ce queAWSCryptographie des paiements ?

AWSLa cryptographie des paiements est une solution géréeAWSservice qui fournit un accès aux fonctions cryptographiques et à la gestion des clés utilisées dans le traitement des paiements conformément aux normes PCI (Payment Card Industry) sans que vous deviez vous procurer des instances HSM de paiement dédiées.AWS La cryptographie des paiements permet aux clients exécutant des fonctions de paiement, tels que les acquéreurs, les facilitateurs de paiement, les réseaux, les commutateurs, les processeurs et les banques, de rapprocher leurs opérations cryptographiques de paiement des applications dans le cloud et de minimiser la dépendance à l'égard des centres de données auxiliaires ou des installations de colocation contenant des HSM de paiement dédiés.

Le service est conçu pour répondre aux règles applicables du secteur, notamment le code PIN PCI, le PCI P2PE et le PCI DSS, et le service exploite le matériel qu'il estCertifié PCI PTS HSM V3 et FIPS 140-2 niveau 3. Il est conçu pour supporter une faible latence etdes niveaux élevés de disponibilité et de résilience.AWS La cryptographie des paiements est totalement élastique et élimine de nombreuses exigences opérationnelles des HSM sur site, telles que la nécessité de fournir du matériel, de gérer en toute sécurité le matériel clé et de maintenir des sauvegardes d'urgence dans des installations sécurisées.AWS La cryptographie des paiements vous permet également de partager des clés avec vos partenaires par voie électronique, ce qui élimine le besoin de partager des composants papier en texte clair.

Vous pouvez utiliser leAWSAPI du plan de contrôle de la cryptographie des paiementspour créer et gérer des clés.

Vous pouvez utiliser leAWSAPI du plan de données de cryptographie des paiementspour utiliser des clés de chiffrement pour le traitement des transactions liées au paiement et les opérations cryptographiques associées.

AWSLa cryptographie des paiements fournit des fonctionnalités importantes que vous pouvez utiliser pour gérer vos clés :

- Création et gestion de systèmes symétriques et asymétriquesAWSLes clés de cryptographie des paiements, y compris les clés TDES, AES et RSA, et spécifient leur utilisation prévue, par exemple pour la génération de CVV ou la dérivation de clés DUKPT.
- Stockez automatiquement votreAWSClés de cryptographie des paiements sécurisées, protégées par des modules de sécurité matériels (HSM) tout en garantissant la séparation des clés entre les cas d'utilisation.

- Créez, supprimez, listez et mettez à jour des alias, qui sont des « noms conviviaux » qui peuvent être utilisés pour accéder ou contrôler l'accès à votreAWSClés de cryptographie de paiement.
- Étiquetez votreAWSClés de cryptographie des paiements pour l'identification, le regroupement, l'automatisation, le contrôle d'accès et le suivi des coûts.
- Importez et exportez des clés symétriques entreAWSCryptographie des paiements et votre HSM (ou un tiers) à l'aide de clés de chiffrement (KEK) conformes à la norme TR-31 (Interoperable Secure Key Exchange Key Block Specification).
- Importez et exportez des clés de chiffrement à clé symétrique (KEK) entreAWSCryptographie des paiements et autres systèmes utilisant des paires de clés asymétriques utilisant des moyens électroniques tels que le TR-34 (Méthode de distribution de clés symétriques utilisant des techniques asymétriques).

Vous pouvez utiliser votreAWSClés de cryptographie des paiements utilisées dans les opérations cryptographiques, telles que :

- Chiffrez, déchiffrez et chiffrez à nouveau les données de manière symétrique ou asymétriqueAWSClés de cryptographie de paiement.
- Traduisez en toute sécurité les données sensibles (telles que les codes PIN des titulaires de carte) entre les clés de chiffrement sans exposer le texte en clair conformément aux règles PCI PIN.
- Générez ou validez les données du titulaire de la carte telles que CVV, CVV2 ou ARQC.
- Générez et validez les codes PIN du titulaire de la carte.
- Générez ou validez des signatures MAC.

Concepts

Découvrez les termes et concepts de base utilisés dans le AWS domaine de la cryptographie des paiements et découvrez comment vous pouvez les utiliser pour protéger vos données.

Alias

> Nom convivial associé à une clé de chiffrement des AWS paiements. L'alias peut être utilisé de manière interchangeable avec l'ARN de la clé dans de nombreuses opérations de l'API de cryptographie des AWS paiements. Les alias permettent de faire pivoter ou de modifier les clés sans affecter le code de votre application. Le nom d'alias est une chaîne comportant jusqu'à 256 caractères. Il identifie de manière unique une clé de cryptographie AWS de paiement

associée au sein d'un compte et d'une région. Dans AWS Payment Cryptography, les noms d'alias commencent `alias/` toujours par.

Le format d'un nom d'alias est le suivant :

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

Par exemple :

```
alias/sampleAlias2
```

ARN de clé

La clé ARN est le nom de ressource Amazon (ARN) d'une entrée clé dans AWS Payment Cryptography. Il s'agit d'un identifiant unique et complet pour la clé de cryptographie des AWS paiements. Un ARN clé comprend une Compte AWS région et un identifiant généré de manière aléatoire. L'ARN n'est pas lié ou dérivé du matériau clé. Comme elles sont automatiquement attribuées lors des opérations de création ou d'importation, ces valeurs ne sont pas idempotentes. Si vous importez plusieurs fois la même clé, vous obtiendrez plusieurs ARN clés dotés de leur propre cycle de vie.

Le format d'un ARN de clé est le suivant :

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

Voici un exemple d'ARN de clé :

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

Identifiant clé

Un identifiant de clé est une référence à une clé et l'une (ou plusieurs) d'entre elles sont des entrées typiques des opérations de cryptographie des AWS paiements. Les identifiants de clé valides peuvent être un ARN de clé ou un alias de clé.

AWS Clés de chiffrement des paiements

AWS Les clés de cryptographie (clés) de paiement sont utilisées pour toutes les fonctions cryptographiques. Les clés sont soit générées directement par vous à l'aide du raccourci clavier

Create, soit ajoutées au système en appelant key import. L'origine d'une clé peut être déterminée en examinant l'attribut KeyOrigin. AWS La cryptographie des paiements prend également en charge les clés dérivées ou intermédiaires utilisées lors d'opérations cryptographiques telles que celles utilisées par DUKPT.

Ces clés ont des attributs immuables et mutables définis lors de leur création. Les attributs, tels que l'algorithme, la longueur et l'utilisation, sont définis lors de la création et ne peuvent pas être modifiés. D'autres, comme la date d'entrée en vigueur ou la date d'expiration, peuvent être modifiées. Consultez la référence de l'API de cryptographie des AWS paiements pour obtenir la liste complète des attributs des clés de cryptographie des AWS paiements.

AWS Les clés de cryptographie de paiement ont des types de clés, principalement définis par la norme ANSI X9 TR 31, qui limitent leur utilisation aux fins prévues, conformément à l'exigence 19 du PCI PIN v3.1.

Les attributs sont liés aux clés à l'aide de blocs de clés lorsqu'ils sont stockés, partagés avec d'autres comptes ou exportés conformément à l'exigence 18-3 du PCI PIN v3.1.

Les clés sont identifiées sur la plateforme AWS de cryptographie des paiements à l'aide d'une valeur unique appelée clé Amazon Resource Name (ARN).

> **Note**
>
> ARNLa clé est générée lorsqu'une clé est initialement créée ou importée dans le service de cryptographie des AWS paiements. Ainsi, si vous ajoutez le même élément clé plusieurs fois à l'aide de la fonctionnalité de clé d'importation, le même matériau clé sera situé sous plusieurs clés, mais chacune aura un cycle de vie de clé différent.

Terminologie de l'industrie

Rubriques

Types de clés courants

POINÇON

Une clé de travail d'acquéreur (AWK) est une clé généralement utilisée pour échanger des données entre un acquéreur/processeur acquéreur et un réseau (tel que Visa ou Mastercard). Historiquement, AWK utilise 3DES pour le chiffrement et était représenté sous la forme TR31_P0_PIN_ENCRYPTION_KEY.

BDK

Une clé de dérivation de base (BDK) est une clé de travail utilisée pour dériver les clés suivantes. Elle est couramment utilisée dans le cadre des processus PCI PIN et PCI P2PE DUKPT. Il est noté TR31_B0_BASE_DERIVATION_KEY.

CLÉ CMK

Une clé principale de carte (CMK) est une ou plusieurs clés spécifiques à une carte généralement dérivées d'une clé principale d'émetteur, d'un PAN et d'un PSN et sont généralement des clés 3DES. Ces clés sont stockées sur la puce EMV lors de la personnalisation. Les clés AC, SMI et SMC sont des exemples de CMK.

CMK-AC

Une clé de cryptogramme d'application (AC) est utilisée dans le cadre des transactions EMV pour générer le cryptogramme de transaction et constitue un type de clé principale de carte.

CMK-SMI

Une clé d'intégrité de la messagerie sécurisée (SMI) est utilisée dans le cadre de l'EMV pour vérifier l'intégrité des charges utiles envoyées à la carte via MAC, telles que les scripts de mise à jour du code PIN. Il s'agit d'un type de clé principale de carte.

CMK-SMC

Une clé de confidentialité des messages sécurisés (SMC) est utilisée dans le cadre de l'EMV pour chiffrer les données envoyées à la carte, telles que les mises à jour du code PIN. Il s'agit d'un type de clé principale de carte.

CVK

Une clé de vérification de carte (CVK) est une clé utilisée pour générer des valeurs CVV, CVV2 et similaires à l'aide d'un algorithme défini, ainsi que pour valider une entrée. Il est désigné sous le nom de TR31_C0_CARD_VERIFICATION_KEY.

IMK

Une clé principale de l'émetteur (IMK) est une clé principale utilisée dans le cadre de la personnalisation des cartes à puce EMV. Il y aura généralement 3 IMK, une pour chacune des clés AC (cryptogramme), SMI (clé principale de script pour l'intégrité/signature) et SMC (clé principale de script pour la confidentialité/le chiffrement).

cinématique inverse

Une clé initiale (IK) est la première clé utilisée dans le processus DUKPT et dérive de la clé de dérivation de base (BDK). Aucune transaction n'est traitée sur cette clé, mais elle est utilisée pour dériver les futures clés qui seront utilisées pour les transactions. La méthode de dérivation pour créer un IK a été définie dans X9. 24-1:2017. Lorsqu'un TDES BDK est utilisé, X9. 24-1:2009 est la norme applicable et IK est remplacé par la clé IPEK (Initial Pin Encryption Key).

IPEK

Une clé de chiffrement par code PIN initial (IPEK) est la clé initiale utilisée dans le processus DUKPT et dérive de la clé de dérivation de base (BDK). Aucune transaction n'est traitée sur cette clé, mais elle est utilisée pour dériver les futures clés qui seront utilisées pour les transactions. IPEK est un terme impropre car cette clé peut également être utilisée pour dériver le cryptage des données et les clés Mac. La méthode de dérivation pour créer un IPEK a été définie dans X9. 24-1:2009. Lorsqu'un AES BDK est utilisé, X9. 24-1:2017 est la norme applicable et l'IPEK est remplacé par Initial Key (IK).

IWK

Une clé de travail de l'émetteur (IWK) est une clé généralement utilisée pour échanger des données entre un émetteur/un processeur émetteur et un réseau (tel que Visa ou Mastercard). Historiquement, IWK utilise 3DES pour le chiffrement et est représenté sous la forme TR31_P0_PIN_ENCRYPTION_KEY.

KEK

Une clé de chiffrement (KEK) est une clé utilisée pour chiffrer d'autres clés à des fins de transmission ou de stockage. Les clés destinées à protéger d'autres clés ont généralement la valeur KeyUsage TR31_K0_KEY_ENCRYPTION_KEY conformément à la norme. TR-31

PEK

Une clé de cryptage PIN (PEK) est un type de clé fonctionnelle utilisée pour chiffrer des codes PIN à des fins de stockage ou de transmission entre deux parties. IWK et AWK sont

deux exemples d'utilisations spécifiques des clés de chiffrement par code PIN. Ces clés sont représentées sous la forme TR31_P0_PIN_ENCRYPTION_KEY.

PGK

PGK (Pin Generation Key) est un autre nom pour une clé de vérification par code PIN. Il n'est pas réellement utilisé pour générer des épingles (qui sont par défaut des nombres cryptographiquement aléatoires) mais plutôt pour générer des valeurs de vérification telles que le PVV.

PVK

Une clé de vérification du code PIN (PVK) est un type de clé fonctionnelle utilisée pour générer des valeurs de vérification du code PIN telles que le code PVV. Les deux types les plus courants sont le TR31_V1_IBM3624_PIN_VERIFICATION_KEY utilisé pour générer les valeurs de décalage IBM3624 et le TR31_V2_VISA_PIN_VERIFICATION_KEY utilisé pour les valeurs de vérification Visa/ABA. Cela peut également être connu sous le nom de clé de génération de code PIN.

Autres termes

ARQC

Le cryptogramme de demande d'autorisation (ARQC) est un cryptogramme généré au moment de la transaction par une carte à puce standard EMV (ou une implémentation sans contact équivalente). Généralement, un ARQC est généré par une carte à puce et transmis à un émetteur ou à son agent pour vérification au moment de la transaction.

CVV

Une valeur de vérification de carte est une valeur secrète statique qui était traditionnellement intégrée sur une bande magnétique et utilisée pour valider l'authenticité d'une transaction. L'algorithme est également utilisé à d'autres fins telles que iCVV, CAVV, CVV2. Il se peut qu'il ne soit pas intégré de cette manière pour d'autres cas d'utilisation.

CVV2

Une valeur de vérification de carte 2 est une valeur secrète statique qui était traditionnellement imprimée au recto (ou au verso) d'une carte de paiement et utilisée pour vérifier l'authenticité des paiements sans carte (par exemple au téléphone ou en ligne). Il utilise le même algorithme que le CVV mais le code de service est défini sur 000.

iCVV

iCVV est une valeur similaire à CVv2 mais incorporée aux données équivalentes à track2 sur une carte EMV (Chip). Cette valeur est calculée à l'aide d'un code de service 999 et est différente du code CVV1/CVV2 afin d'empêcher que des informations volées ne soient utilisées pour créer de nouveaux identifiants de paiement d'un type différent. Par exemple, si des données de transaction par puce ont été obtenues, il n'est pas possible de les utiliser pour générer une bande magnétique (CVV1) ou pour des achats en ligne (CVV2).

Il utilise une ??? clé

DUKPT

La clé unique dérivée par transaction (DUKPT) est une norme de gestion des clés généralement utilisée pour définir l'utilisation de clés de chiffrement à usage unique sur les POS/POI physiques. Historiquement, DUKPT utilise 3DES pour le chiffrement. La norme industrielle pour le DUKPT est définie dans la norme ANSI X9.24-3-2017.

EMV

EMV (initialement Europay, Mastercard, Visa) est un organisme technique qui travaille avec les parties prenantes du secteur des paiements pour créer des normes et des technologies de paiement interopérables. Un exemple de norme concerne les cartes à puce/sans contact et les terminaux de paiement avec lesquels elles interagissent, y compris la cryptographie utilisée. La dérivation de clés EMV fait référence à des méthodes permettant de générer des clés uniques pour chaque carte de paiement sur la base d'un ensemble initial de clés tel qu'un `IMK`

HSM

Un module de sécurité matériel (HSM) est un dispositif physique qui protège les opérations cryptographiques (par exemple, le chiffrement, le déchiffrement et les signatures numériques) ainsi que les clés sous-jacentes utilisées pour ces opérations.

KCV

La valeur de contrôle des clés (KCV) fait référence à une variété de méthodes de somme de contrôle principalement utilisées pour comparer les clés les unes aux autres sans avoir accès au contenu clé réel. Les KCV ont également été utilisés pour la validation de l'intégrité (en particulier lors de l'échange de clés), bien que ce rôle soit désormais inclus dans les formats de blocs de clés tels que `TR-31`. Pour les clés TDES, le KCV est calculé en chiffrant 8 octets, chacun ayant une valeur de zéro, avec la clé à vérifier et en conservant les 3 octets d'ordre le plus élevé du résultat chiffré. Pour les clés AES, le KCV est calculé à l'aide d'un algorithme CMAC où les

données d'entrée sont de 16 octets de zéro et en conservant les 3 octets d'ordre le plus élevé du résultat chiffré.

KDH

Un hôte de distribution de clés (KDH) est un appareil ou un système qui envoie des clés dans le cadre d'un processus d'échange de clés tel que le TR-34. Lors de l'envoi de clés depuis AWS Payment Cryptography, il est considéré comme le KDH.

KIF

Une installation d'injection de clés (KIF) est une installation sécurisée utilisée pour initialiser les terminaux de paiement, notamment pour les charger avec des clés de chiffrement.

KRD

Un dispositif de réception de clés (KRD) est un appareil qui reçoit des clés dans le cadre d'un processus d'échange de clés tel que le TR-34. Lorsque vous envoyez des clés à AWS Payment Cryptography, celle-ci est considérée comme le KRD.

KSN

Un numéro de série de clé (KSN) est une valeur utilisée comme entrée pour le chiffrement/déchiffrement DUKPT afin de créer des clés de chiffrement uniques par transaction. Le KSN se compose généralement d'un identifiant BDK, d'un identifiant de terminal semi-unique ainsi que d'un compteur de transactions qui s'incrémente à chaque transition traitée sur un terminal de paiement donné.

mPOC

Le mPOC (point de vente mobile sur matériel commercial) est une norme PCI qui répond aux exigences de sécurité des solutions permettant aux commerçants d'accepter les codes PIN des titulaires de carte ou les paiements sans contact à l'aide d'un smartphone ou d'autres appareils mobiles commerciaux off-the-shelf (COTS).

POÊLE

Un numéro de compte principal (PAN) est un identifiant unique pour un compte tel qu'une carte de crédit ou de débit. Généralement de 13 à 19 chiffres. Les 6 à 8 premiers chiffres identifient le réseau et la banque émettrice.

Bloc PIN

Bloc de données contenant un code PIN pendant le traitement ou la transmission ainsi que d'autres éléments de données. Les formats de bloc PIN normalisent le contenu du bloc PIN

et la manière dont il peut être traité pour récupérer le code PIN. La plupart des blocs PIN sont composés du code PIN, de la longueur du code PIN, et contiennent souvent une partie ou la totalité du PAN. AWS La cryptographie des paiements prend en charge les formats ISO 9564-1 0, 1, 3 et 4. Le format 4 est requis pour les clés AES. Lors de la vérification ou de la traduction des codes PIN, il est nécessaire de spécifier le bloc PIN des données entrantes ou sortantes.

POI

Le point d'interaction (POI), également fréquemment utilisé comme synonyme de point de vente (POS), est le périphérique matériel avec lequel le titulaire de la carte interagit pour présenter son identifiant de paiement. Un exemple de POI est le terminal physique situé sur le site d'un commerçant. Pour consulter la liste des terminaux PCI PTS POI certifiés, consultez le site Web PCI.

PSN

Le numéro de séquence PAN (PSN) est une valeur numérique utilisée pour différencier plusieurs cartes émises avec le même PAN.

Clé publique

Lors de l'utilisation de chiffrements asymétriques (RSA), la clé publique est le composant public d'une paire de clés publique-privée. La clé publique peut être partagée et distribuée aux entités qui ont besoin de chiffrer les données pour le propriétaire de la paire de clés publique-privée. Pour les opérations de signature numérique, la clé publique est utilisée pour vérifier la signature.

Clé privée

Lors de l'utilisation de chiffrements asymétriques (RSA), la clé privée est le composant privé d'une paire de clés publique-privée. La clé privée est utilisée pour déchiffrer des données ou créer des signatures numériques. À l'instar des clés de cryptographie AWS de paiement symétriques, les clés privées sont créées de manière sécurisée par les HSM. Ils sont déchiffrés uniquement dans la mémoire volatile du HSM et uniquement pendant le temps nécessaire au traitement de votre demande cryptographique.

PVV

Une valeur de vérification du code PIN (PVV) est un type de sortie cryptographique qui peut être utilisé pour vérifier un code PIN sans enregistrer le code PIN réel. Bien qu'il s'agisse d'un terme générique, dans le contexte de la cryptographie des AWS paiements, PVV fait référence à la méthode PVV Visa ou ABA. Ce PVV est un numéro à quatre chiffres dont les entrées sont le numéro de carte, le numéro de séquence Pan, le PAN lui-même et une clé de vérification du code

PIN. Au cours de la phase de validation, AWS Payment Cryptography recrée en interne le PVV à l'aide des données de transaction et le compare à nouveau à la valeur enregistrée par le client de AWS Payment Cryptography. De cette façon, il est conceptuellement similaire à un hachage cryptographique ou à un MAC.

RSA Wrapper/Unwrap

L'encapsulation RSA utilise une clé asymétrique pour encapsuler une clé symétrique (telle qu'une clé TDES) afin de la transmettre à un autre système. Seul le système possédant la clé privée correspondante peut déchiffrer la charge utile et charger la clé symétrique. À l'inverse, RSA unwrap déchiffre en toute sécurité une clé chiffrée à l'aide de RSA, puis la charge dans le chiffrement des paiements. AWS L'encapsulation RSA est une méthode d'échange de clés de bas niveau qui ne transmet pas les clés sous forme de blocs de clés et n'utilise pas de signature de charge utile par l'expéditeur. Des contrôles alternatifs doivent être envisagés pour vérifier si les attributs clés ne sont pas modifiés.

Le TR-34 utilise également le RSA en interne, mais il s'agit d'un format distinct et n'est pas interopérable.

TR-31

Le TR-31 (officiellement défini comme ANSI X9 TR 31) est un format de bloc clé défini par l'American National Standards Institute (ANSI) pour permettre de définir les attributs clés dans la même structure de données que les données clés elles-mêmes. Le format de bloc de touches TR-31 définit un ensemble d'attributs clés liés à la clé afin qu'ils soient maintenus ensemble. AWS La cryptographie des paiements utilise les termes normalisés TR-31 dans la mesure du possible pour garantir une séparation correcte des clés et l'objectif des clés. Le TR-31 a été remplacé par la norme ANSI X9.143-2022.

TR-34

Le TR-34 est une implémentation de la norme ANSI X9.24-2 qui décrit un protocole permettant de distribuer de manière sécurisée des clés symétriques (telles que 3DES et AES) à l'aide de techniques asymétriques (telles que RSA). AWS La cryptographie des paiements utilise les méthodes TR-34 pour permettre l'importation et l'exportation sécurisées des clés.

Services connexes

AWS Key Management Service

AWSService de gestion des clés (AWSKMS) est un service géré qui vous permet de créer et de contrôler facilement les clés cryptographiques utilisées pour protéger vos données.AWS KMS utilise des modules de sécurité matériels (HSM) pour protéger et valider votreAWSClés KMS.

AWS CloudHSM

AWS CloudHSMfournit aux clients des instances HSM à usage général dédiées dansAWSNuage.AWS CloudHSMpeut fournir diverses fonctions cryptographiques telles que la création de clés, la signature de données ou le chiffrement et le déchiffrement de données.

Pour plus d'informations

- Pour en savoir plus sur les termes et concepts utilisés dansAWSCryptographie des paiements, voirAWSConcepts de cryptographie des paiements.
- Pour plus d'informations surAWSAPI du plan de contrôle de la cryptographie des paiements, voirAWSRéférence de l'API du plan de contrôle de la cryptographie des paiements.
- Pour plus d'informations surAWSAPI Payment Cryptography Data Plane, voirAWSRéférence de l'API du plan de données de cryptographie des paiements.
- Pour des informations techniques détaillées sur la façon dontAWSLa cryptographie des paiements utilise la cryptographie et sécuriseAWSClés de cryptographie de paiement, voirDétails cryptographiques.

Points de terminaison pour AWS Payment Cryptography

Pour vous connecter par programmation AWS Payment Cryptography, vous utilisez un point de terminaison, le URL point d'entrée du service. Les outils de ligne de commande AWS SDKs et les outils de ligne de commande utilisent automatiquement le point de terminaison par défaut du service en Région AWS fonction du contexte régional d'une demande. Il n'est donc généralement pas nécessaire de définir explicitement ces valeurs. Si nécessaire, vous pouvez spécifier un point de terminaison différent pour vos API demandes.

Points de terminaison du plan de contrôle

Nom de la région	Région	Point de terminaison	Protocole
USA Est (Virginie du Nord)	`us-east-1`	plan de contrôle. payment-cryptography.us-east-1.amazonaws.com	HTTPS
USA Est (Ohio)	`us-east-2`	plan de contrôle. payment-cryptography.us-east-2.amazonaws.com	HTTPS
USA Ouest (Oregon)	`us-west-2`	plan de contrôle. payment-cryptography.us-west-2.amazonaws.com	HTTPS
Asie-Pacifique (Singapour)	`ap-southeast-1`	plan de contrôle. payment-cryptography.ap-southeast-1.amazonaws.com	HTTPS
Asie-Pacifique (Tokyo)	`ap-northeast-1`	plan de contrôle. payment-cryptography.ap-northeast-1.amazonaws.com	HTTPS
Europe (Francfort)	`eu-central-1`	plan de contrôle. payment-cryptography.eu-central-1.amazonaws.com	HTTPS
Europe (Irlande)	`eu-west-1`	plan de contrôle. payment-cryptography.eu-west-1.amazonaws.com	HTTPS

Points de terminaison du plan de données

Nom de la région	Région	Point de terminaison	Protocole
USA Est (Virginie du Nord)	`us-east-1`	plan de données. payment-cryptography.us-east-1.amazonaws.com	HTTPS
USA Est (Ohio)	`us-east-2`	plan de données. payment-cryptography.us-east-2.amazonaws.com	HTTPS

Nom de la région	Région	Point de terminaison	Protocole
USA Ouest (Oregon)	`us-west-2`	plan de données. payment-cryptography.us-west-2.amazonaws.com	HTTPS
Asie-Pacifique (Singapour)	`ap-southeast-1`	plan de données. payment-cryptography.ap-southeast-1.amazonaws.com	HTTPS
Asie-Pacifique (Tokyo)	`ap-northeast-1`	plan de données. payment-cryptography.ap-northeast-1.amazonaws.com	HTTPS
Europe (Francfort)	`eu-central-1`	plan de données. payment-cryptography.eu-central-1.amazonaws.com	HTTPS
Europe (Irlande)	`eu-west-1`	plan de données. payment-cryptography.eu-west-1.amazonaws.com	HTTPS

Débuter avec la cryptographie des AWS paiements

Pour commencer à utiliser le chiffrement des AWS paiements, vous devez d'abord créer des clés, puis les utiliser dans diverses opérations cryptographiques. Le didacticiel ci-dessous fournit un cas d'utilisation simple de génération d'une clé à utiliser pour générer/vérifier CVV2 des valeurs. Pour essayer d'autres exemples et explorer les modèles de déploiement qu'ils contiennentAWS, essayez l'atelier de cryptographie des AWS paiements suivant ou explorez notre exemple de projet disponible sur Github

Ce didacticiel explique comment créer une clé unique et effectuer des opérations cryptographiques à l'aide de cette clé. Ensuite, vous supprimez la clé si vous ne la souhaitez plus, ce qui complète le cycle de vie de la clé.

> **Warning**
>
> Les exemples présentés dans ce guide de l'utilisateur peuvent utiliser des valeurs d'échantillon. Nous vous recommandons vivement de ne pas utiliser de valeurs d'échantillon dans un environnement de production, telles que les numéros de série clés.

Rubriques

Prérequis

Avant de commencer, assurez-vous que :

- Vous êtes autorisé à accéder au service. Pour plus d'informations, consultez IAMles politiques.
- Vous l'avez AWS CLIinstallé. Vous pouvez également utiliser AWS SDKsou accéder AWS APIsà la cryptographie des AWS paiements, mais les instructions de ce didacticiel utilisent le AWS CLI.

Étape 1 : Création d'une clé

La première étape consiste à créer une clé. Dans ce didacticiel, vous allez créer une clé 3 DES (2 KEYTDES) CVKdouble longueur pour générer et vérifier CVV2 les valeurs CVV /.

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

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (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"
    }
}
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east- 2:111122223333:key/tqv5yij6wtxx64pi. Vous en aurez besoin à l'étape suivante.

Étape 2 : générer une CVV2 valeur à l'aide de la clé

Dans cette étape, vous générez un CVV2 pour une date donnée PAN et une date d'expiration à l'aide de la clé de l'étape 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"
}
```

Prenez note du `cardDataValue` numéro 144 à 3 chiffres dans ce cas. Vous en aurez besoin à l'étape suivante.

Étape 3 : Vérifiez la valeur générée à l'étape 2

Dans cet exemple, vous validez le formulaire CVV2 de l'étape 2 à l'aide de la clé que vous avez créée à l'étape 1.

Exécutez la commande suivante pour valider leCVV2.

```
$  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"
}
```

Le service renvoie une HTTP réponse de 200 pour indiquer qu'il a validé leCVV2.

Étape 4 : Réaliser un test négatif

Au cours de cette étape, vous créez un test négatif dans lequel le résultat n'CVV2est pas correct et n'est pas validé. Vous essayez de valider une erreur à l'CVV2aide de la clé que vous avez créée à l'étape 1. Il s'agit d'une opération attendue, par exemple si le titulaire de la carte a mal saisi CVV2 la commande.

```
$  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.
```

Le service renvoie une HTTP réponse de 400 avec le message « Échec de la vérification des données de validation de la carte » et un motif lNVALID _ VALIDATION _DATA.

Étape 5 : (Facultatif) Nettoyer

Vous pouvez maintenant supprimer la clé que vous avez créée à l'étape 1. Pour minimiser les modifications irrécupérables, la période de suppression des clés par défaut est de sept jours.

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

```
{
    "Key": {
```

```
        "CreateTimestamp": "2022-10-27T08:27:51.795000-07:00",
        "DeletePendingTimestamp": "2022-11-03T13:37:12.114000-07:00",
        "Enabled": true,
        "Exportable": true,
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
        "KeyAttributes": {
            "KeyAlgorithm": "TDES_3KEY",
            "KeyClass": "SYMMETRIC_KEY",
            "KeyModesOfUse": {
                "Decrypt": true,
                "DeriveKey": false,
                "Encrypt": true,
                "Generate": false,
                "NoRestrictions": false,
                "Sign": false,
                "Unwrap": true,
                "Verify": false,
                "Wrap": true
            },
            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY"
        },
        "KeyCheckValue": "CADDA1",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "DELETE_PENDING",
        "UsageStartTimestamp": "2022-10-27T08:27:51.753000-07:00"
    }
}
```


Prenez note de deux champs dans le résultat. La valeur `deletePendingTimestamp` est fixée par défaut à sept jours dans le futur. Le keyState est réglé sur`DELETE_PENDING`. Vous pouvez annuler cette suppression à tout moment avant l'heure de suppression prévue en appelant `restore-key`.

Gestion de clés

Pour commencer à utiliser la cryptographie des AWS paiements, vous devez créer une clé de cryptographie des AWS paiements.

Les rubriques de cette section expliquent comment créer et gérer différents types de clés de chiffrement des AWS paiements, de leur création à leur suppression. Il inclut des rubriques sur la création, la modification et l'affichage des clés, le balisage des clés, la création d'alias de clés, ainsi que l'activation et la désactivation des clés.

Rubriques

Génération de clés

Vous pouvez créer des clés AWS de cryptographie de paiement à l'aide de cette CreateKey API opération. Au cours de ce processus, vous allez spécifier différents attributs de la clé ou de la sortie résultante, tels que l'algorithme clé (par exemple, TDES _3KEY), le KeyUsage (par exemple TR31 _P0_ _ PIN ENCRYPTION _KEY), les opérations autorisées (par exemple, chiffrer, signer) et si elle est exportable. Vous ne pouvez pas modifier ces propriétés une fois la clé de chiffrement des AWS paiements créée.

Exemples

- Création d'une clé asymétrique (RSA)
- Génération d'une clé PIN de valeur de vérification (PVV)

Génération d'une KEY TDES clé 2

Example

Cette commande génère une KEY TDES touche 2 dans le but de générer et de vérifier CVV2 les valeurs CVV /. La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'un KCV (Key Check Value).

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


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

```
        "KeyCheckValue": "B72F",
        "KeyCheckValueAlgorithm": "ANSI_X9_24",
        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
        "KeyState": "CREATE_COMPLETE",
        "UsageStartTimestamp": "2022-10-26T16:04:11.559000-07:00"
    }
}
```

Génération d'une clé de chiffrement par code PIN

Example Génération d'une clé de chiffrement par code PIN (PEK)

Cette commande génère une KEY TDES clé à 3 dans le but de chiffrer les PIN valeurs (connue sous le nom de clé de chiffrement par épingle). Cette clé peut être utilisée pour sécuriser le stockage PINs ou pour le déchiffrement PINs fourni lors d'une tentative de vérification, par exemple lors d'une transaction. La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'un KCV (Key Check Value).

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

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

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

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

Création d'une clé asymétrique (RSA)

Example

Dans cet exemple, nous allons générer une nouvelle paire de clés asymétriques de RSA 2048 bits. Une nouvelle clé privée sera générée ainsi que la clé publique correspondante. La clé publique peut être récupérée à l'aide du 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"
    }
}
```

Génération d'une clé PIN de valeur de vérification (PVV)

Example

Cette commande génère une KEY TDES touche 3 dans le but de générer des PVV valeurs (connue sous le nom de valeur de vérification du code PIN). Vous pouvez utiliser cette clé pour générer une PVV valeur qui peut être comparée à un calcul ultérieurPVV. La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'un KCV (Key Check Value).

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

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

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

Clés de liste

List Keys présente une liste des clés accessibles à l'appelant dans ce compte et cette région.

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"
    }
    ]
}
```


Activation et désactivation des clés

Vous pouvez désactiver et réactiver les clés AWS de chiffrement des paiements. Lorsque vous créez une clé, elle est activée par défaut. Si vous désactivez une clé, elle ne peut être utilisée dans aucune opération cryptographique tant que vous ne l'avez pas réactivée. Les commandes de démarrage/arrêt de l'utilisation prennent effet immédiatement. Il est donc recommandé de vérifier l'utilisation avant de procéder à une telle modification. Vous pouvez également définir une modification (démarrer ou arrêter l'utilisation) pour qu'elle prenne effet dans le futur à l'aide du `timestamp` paramètre optionnel.

Comme elle est temporaire et facile à annuler, la désactivation d'une clé de chiffrement de AWS paiement constitue une alternative plus sûre à la suppression d'une clé de cryptographie de AWS paiement, une action destructrice et irréversible. Si vous envisagez de supprimer une clé de chiffrement des AWS paiements, désactivez-la d'abord et assurez-vous que vous n'aurez pas besoin de l'utiliser pour chiffrer ou déchiffrer des données à l'avenir.

Rubriques

Démarrer l'utilisation des clés

L'utilisation des clés doit être activée afin d'utiliser une clé pour les opérations cryptographiques. Si une clé n'est pas activée, vous pouvez utiliser cette opération pour la rendre utilisable. Le champ `UsageStartTimestamp` indiquera quand la clé est devenue ou deviendra active. Ce sera le cas dans le passé pour un jeton activé, et dans le futur s'il est en attente d'activation.

Example

Dans cet exemple, il est demandé qu'une clé soit activée pour son utilisation. La réponse inclut les informations clés et l'indicateur d'activation est passé à vrai. Cela se reflétera également dans l'objet de réponse 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"
  }
}
```

Arrêter l'utilisation des clés

Si vous n'avez plus l'intention d'utiliser une clé, vous pouvez arrêter de l'utiliser pour empêcher de nouvelles opérations cryptographiques. Cette opération n'est pas permanente, vous pouvez donc l'inverser en utilisant la clé de départ. Vous pouvez également définir une clé pour qu'elle soit désactivée à l'avenir. Le champ `UsageStopTimestamp` indiquera quand la clé est devenue ou sera désactivée.

Example

Dans cet exemple, il est demandé d'arrêter l'utilisation des clés à l'avenir. Après exécution, cette clé ne peut pas être utilisée pour des opérations cryptographiques à moins d'être réactivée via l'utilisation de la clé de démarrage. La réponse inclut les informations relatives à la clé et l'indicateur d'activation est passé à faux. Cela se reflétera également dans l'objet de réponse 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"
  }
}
```


Suppression des clés

La suppression d'une clé de chiffrement de AWS paiement supprime le contenu de la clé et toutes les métadonnées associées à la clé et est irréversible, sauf si une copie de la clé est disponible en dehors de la cryptographie des AWS paiements. Une fois qu'une clé est supprimée, vous ne pouvez plus déchiffrer les données chiffrées sous cette clé, ce qui signifie que les données peuvent devenir irrécupérables. Vous ne devez supprimer une clé que lorsque vous êtes certain de ne plus avoir besoin de l'utiliser et qu'aucune autre personne n'utilise cette clé. En cas de doute, pensez à désactiver la clé au lieu de la supprimer. Vous pouvez réactiver une clé désactivée si vous devez la réutiliser ultérieurement, mais vous ne pouvez pas récupérer une clé de chiffrement des AWS paiements supprimée à moins de pouvoir la réimporter depuis une autre source.

Avant de supprimer une clé, assurez-vous que vous n'en avez plus besoin. AWS La cryptographie des paiements ne stocke pas les résultats des opérations cryptographiques CVV2 et n'est pas en mesure de déterminer si une clé est nécessaire pour tout matériel cryptographique persistant.

AWS La cryptographie des paiements ne supprime jamais les clés appartenant à AWS des comptes actifs, sauf si vous planifiez explicitement leur suppression et que le délai d'attente obligatoire expire.

Toutefois, vous pouvez choisir de supprimer une clé AWS de chiffrement des paiements pour une ou plusieurs des raisons suivantes :

- Pour terminer le cycle de vie d'une clé dont vous n'avez plus besoin

- Pour éviter les frais de gestion associés à la gestion des clés de chiffrement des AWS paiements non utilisées

> **Note**
>
> Si vous fermez ou supprimez votre Compte AWS, votre clé AWS de cryptographie de paiement devient inaccessible. Il n'est pas nécessaire de planifier la suppression de votre clé de chiffrement des AWS paiements indépendamment de la fermeture du compte.

AWS Payment Cryptography enregistre une entrée dans votre AWS CloudTrailjournal lorsque vous planifiez la suppression de la clé cryptographique des AWS paiements et lorsque la clé de cryptographie des AWS paiements est effectivement supprimée.

À propos de la période d'attente

La suppression d'une clé étant irréversible, AWS Payment Cryptography vous oblige à définir un délai d'attente compris entre 3 et 180 jours. Le délai d'attente par défaut est de sept jours.

Cependant, la période d'attente réelle peut être jusqu'à 24 heures plus longue celle que vous avez planifiée. Pour obtenir la date et l'heure réelles auxquelles la clé AWS de cryptographie de paiement sera supprimée, utilisez les GetKey opérations. Assurez-vous de noter le fuseau horaire.

Pendant la période d'attente, le statut de la clé AWS de chiffrement des paiements et l'état de la clé sont En attente de suppression.

> **Note**
>
> Une clé AWS de chiffrement de paiement en attente de suppression ne peut être utilisée dans aucune opération cryptographique.

Une fois la période d'attente terminée, AWS Payment Cryptography supprime la clé de cryptographie des AWS paiements, ses alias et toutes les métadonnées associées à AWS la cryptographie des paiements.

Utilisez le délai d'attente pour vous assurer que vous n'avez pas besoin de la clé AWS de chiffrement des paiements, ni maintenant ni dans le futur. Si vous constatez que vous avez besoin de la clé pendant la période d'attente, vous pouvez annuler la suppression de la clé avant la fin de la période

d'attente. Une fois la période d'attente terminée, vous ne pouvez pas annuler la suppression de la clé, et le service supprime la clé.

Example

Dans cet exemple, il est demandé de supprimer une clé. Outre les informations clés de base, deux champs pertinents indiquent que l'état de la clé a été changé en DELETE _ PENDING et indique deletePendingTimestamp le moment où la suppression de la clé est actuellement planifiée.

```
$ 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

Dans cet exemple, une suppression en attente est annulée. Une fois l'opération terminée avec succès, aucune clé ne sera supprimée conformément au calendrier précédent. La réponse contient les informations clés de base ; en outre, deux champs pertinents ont été modifiés : `KeyState` et`deletePendingTimestamp`. `KeyState`est renvoyé à une valeur de CREATE _COMPLETE, tandis qu'`DeletePendingTimestamp`il est supprimé.

```
$ 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"
    }
}
```

Clés d'importation et d'exportation

AWS Les clés de chiffrement des paiements peuvent être importées depuis d'autres solutions ou exportées vers d'autres solutions (telles que d'autresHSMs). Il s'agit d'un cas d'utilisation courant pour échanger des clés avec des fournisseurs de services à l'aide des fonctionnalités d'importation et d'exportation. En tant que service cloud, AWS Payment Cryptography adopte une approche électronique moderne de la gestion des clés tout en vous aidant à maintenir la conformité et les contrôles applicables. L'objectif à long terme est d'abandonner les composants clés sur support papier au profit de moyens électroniques normalisés d'échange de clés.

Échange de clé de chiffrement (KEK)

AWS La cryptographie des paiements encourage l'utilisation de la cryptographie à clé publique (RSA) pour l'échange initial de clés en utilisant la norme bien établie ANSIX9.24 TR-34. Les noms courants de ce type de clé initial incluent Key Encryption Key (KEK), Zone Master Key (ZMK) et Zone Control Master Key (ZCMK). Si vos systèmes ou partenaires ne sont pas encore en mesure de prendre en charge le TR-34, vous pouvez également envisager d'utiliser RSA Wrap/Unwrap.

Si vous devez continuer à traiter les composants clés en papier jusqu'à ce que tous les partenaires acceptent l'échange électronique de clés, vous pouvez envisager de le conserver hors ligne HSM à cette fin.

> **Note**
>
> Si vous souhaitez importer vos propres clés de test, veuillez consulter l'exemple de projet sur Github. Pour obtenir des instructions sur la façon d'importer/exporter des clés depuis d'autres plateformes, veuillez consulter le guide de l'utilisateur de ces plateformes.

Échange de clés de travail (WK)

AWS La cryptographie des paiements utilise la norme industrielle pertinente (ANSIX9.24 TR 31-2018) pour échanger des clés de travail. Le TR-31 suppose que KEK a déjà été échangé. Cela

est conforme PCI PIN à l'exigence de lier cryptographiquement le matériel clé à son type de clé et à son utilisation à tout moment. Les clés de travail ont différents noms, notamment les clés de travail de l'acquéreur, les clés de travail de l'émetteur BDKIPEK, etc.

Rubriques

Clés d'importation

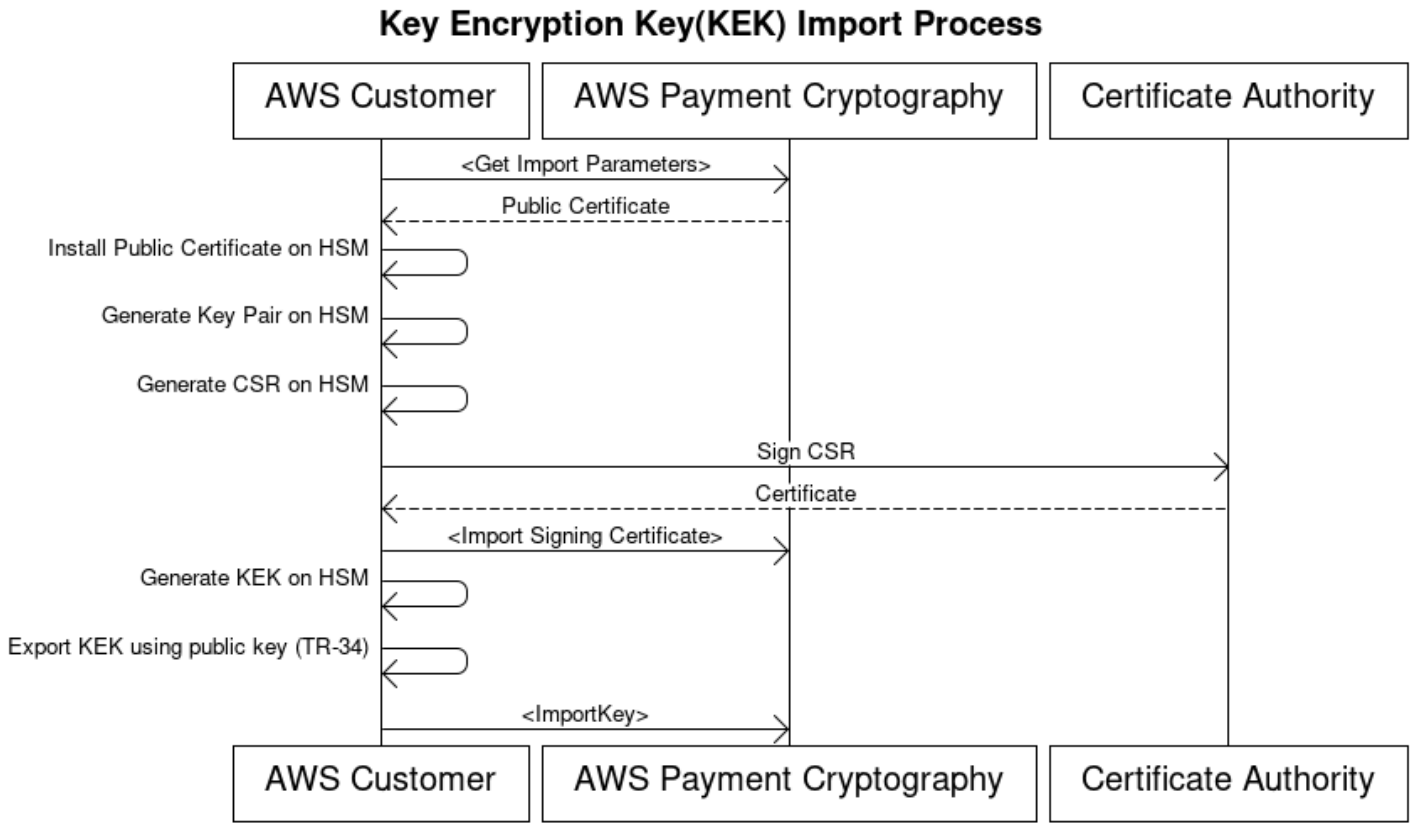
> **Important**
>
> Les exemples peuvent nécessiter la dernière version de la AWS CLI V2. Avant de commencer, assurez-vous d'avoir effectué la mise à niveau vers la dernière version.

Rubriques

Importation de clés symétriques

Rubriques

Importation de clés à l'aide de techniques asymétriques (TR-34)



Vue d'ensemble : Le TR-34 utilise la cryptographie RSA asymétrique pour chiffrer les clés symétriques à des fins d'échange et pour garantir la source des données (signature). Cela garantit à la fois la confidentialité (cryptage) et l'intégrité (signature) de la clé encapsulée.

Si vous souhaitez importer vos propres clés, veuillez consulter l'exemple de projet sur Github. Pour obtenir des instructions sur la façon d'importer/exporter des clés depuis d'autres plateformes, veuillez consulter le guide de l'utilisateur de ces plateformes.

1. Appelez la commande d'initialisation de l'importation

   Appelez `get-parameters-for-import` pour initialiser le processus d'importation. Cela API générera une paire de clés à des fins d'importation de clés, signera la clé et renverra le certificat et la racine du certificat. En fin de compte, la clé à exporter doit être chiffrée à l'aide de cette clé. Dans la terminologie du TR-34, cela s'appelle le KRD Cert. Notez que ces certificats sont de courte durée et ne sont destinés qu'à cette fin.

2. Installer le certificat public sur le système source clé

   Dans de nombreux HSMs cas, vous devrez peut-être installer/charger/approuver le certificat public généré à l'étape 1 afin d'exporter les clés à l'aide de celui-ci.

3. Générez une clé publique et fournissez la racine du certificat à AWS Payment Cryptography

   Pour garantir l'intégrité de la charge utile transmise, celle-ci est signée par l'expéditeur (connu sous le nom d'hôte de distribution des clés ouKDH). L'expéditeur souhaitera générer une clé publique à cette fin, puis créer un certificat de clé publique (X509) qui pourra être renvoyé à AWS Payment Cryptography. AWS Private CA est une option pour générer des certificats, mais il n'existe aucune restriction quant à l'autorité de certification utilisée.

   Une fois que vous avez obtenu le certificat, vous devez charger le certificat racine dans AWS Payment Cryptography à l'aide des `importKey` commandes KeyMaterialType de `ROOT_PUBLIC_KEY_CERTIFICATE` et KeyUsageType de`TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`.

4. Exporter la clé depuis le système source

   De nombreux HSMs systèmes connexes permettent d'exporter des clés en utilisant la norme TR-34. Vous devez spécifier la clé publique de l'étape 1 en tant que certificat KRD (de chiffrement) et la clé de l'étape 3 en tant que certificat KDH (de signature). Pour importer dans AWS Payment Cryptography, vous devez spécifier le format TR-34.2012 sans CMS deux passes, également appelé format TR-34 Diebold.

5. Clé d'importation d'appels

   À la dernière étape, vous appellerez le importKey API avec un KeyMaterialType de`TR34_KEY_BLOCK`. Il `certificate-authority-public-key-identifier` s'agira de la clé ARN de l'autorité de certification racine importée à l'étape 3, du matériel clé de l'étape 4 `key-material` sera encapsulé et `signing-key-certificate` du certificat original de l'étape 3. Vous devrez également fournir le jeton d'importation indiqué à l'étape 1.

6. Utiliser la clé importée pour les opérations cryptographiques ou les importations ultérieures

   Si la clé importée KeyUsage était TR31 _K0_ _ KEY ENCRYPTION _KEY, cette clé peut être utilisée pour les importations de clés suivantes à l'aide de la TR-31. Si le type de clé était un autre type (tel que TR31 _D0_ _ _ SYMMETRIC DATA ENCRYPTION _KEY), la clé peut être directement utilisée pour des opérations cryptographiques.

Importer des clés à l'aide de techniques asymétriques (RSAUnwrap)

Vue d'ensemble : La cryptographie des AWS paiements prend en charge l'RSAencapsulation et le déballage pour l'échange de clés lorsque le TR-34 n'est pas possible. Semblable à la TR-34, cette technique utilise la cryptographie RSA asymétrique pour chiffrer des clés symétriques à des fins d'échange. Cependant, contrairement à la TR-34, cette méthode ne fait pas signer la charge utile par l'expéditeur. De plus, cette technique d'RSAencapsulation ne préserve pas l'intégrité des métadonnées clés pendant le transfert car les blocs clés ne sont pas inclus.

> **Note**
>
> RSAwrap peut être utilisé pour importer ou exporter TDES et AES -128 clés.

1. Appelez la commande d'initialisation de l'importation

   Appelez `get-parameters-for-import` pour initialiser le processus d'importation avec un type de matériau clé KEY _CRYPTOGRAM. WrappingKeyAlgorithm peut être de RSA _2048 lors de l'échange TDES de clés. RSA_3072 ou RSA _4096 peuvent être utilisés lors de l'échange TDES de clés ou -128. AES Cela API générera une paire de clés à des fins d'importation de clés, signera la clé à l'aide d'une racine de certificat et renverra à la fois le certificat et la racine du certificat. En fin de compte, la clé à exporter doit être chiffrée à l'aide de cette clé. Notez que ces certificats sont de courte durée et ne sont destinés qu'à cette fin.

```
$ 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. Installer le certificat public sur le système source clé

   Dans de nombreux HSMs cas, vous devrez peut-être installer/charger/faire confiance au certificat public (et/ou à sa racine) généré à l'étape 1 afin d'exporter les clés à l'aide de celui-ci.

3. Exporter la clé depuis le système source

   De nombreux HSMs systèmes connexes permettent d'exporter des clés à l'aide de RSA Wrap. Vous devez spécifier la clé publique de l'étape 1 en tant que certificat (de cryptageWrappingKeyCertificate). Si vous avez besoin de la chaîne de confiance, elle est contenue dans le champ WrappingKeyCertificateChain de réponse de l'étape #1. Lorsque vous exportez la clé depuis votreHSM, vous devez spécifier le format à utiliserRSA, Mode de remplissage = PKCS #1 v2.2 OAEP (avec SHA 256 ou SHA 512).

4. Clé d'importation d'appels

   À la dernière étape, vous appellerez le importKey API avec un KeyMaterialType de`KeyMaterial`. Vous aurez besoin du jeton d'importation de l'étape 1 et du `key-material` (matériel clé enveloppé) de l'étape 3. Vous devrez fournir les paramètres clés (tels que l'utilisation des clés) car le RSA wrap n'utilise pas de blocs clés.

```
$ 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"
                    }
                }
            }
```

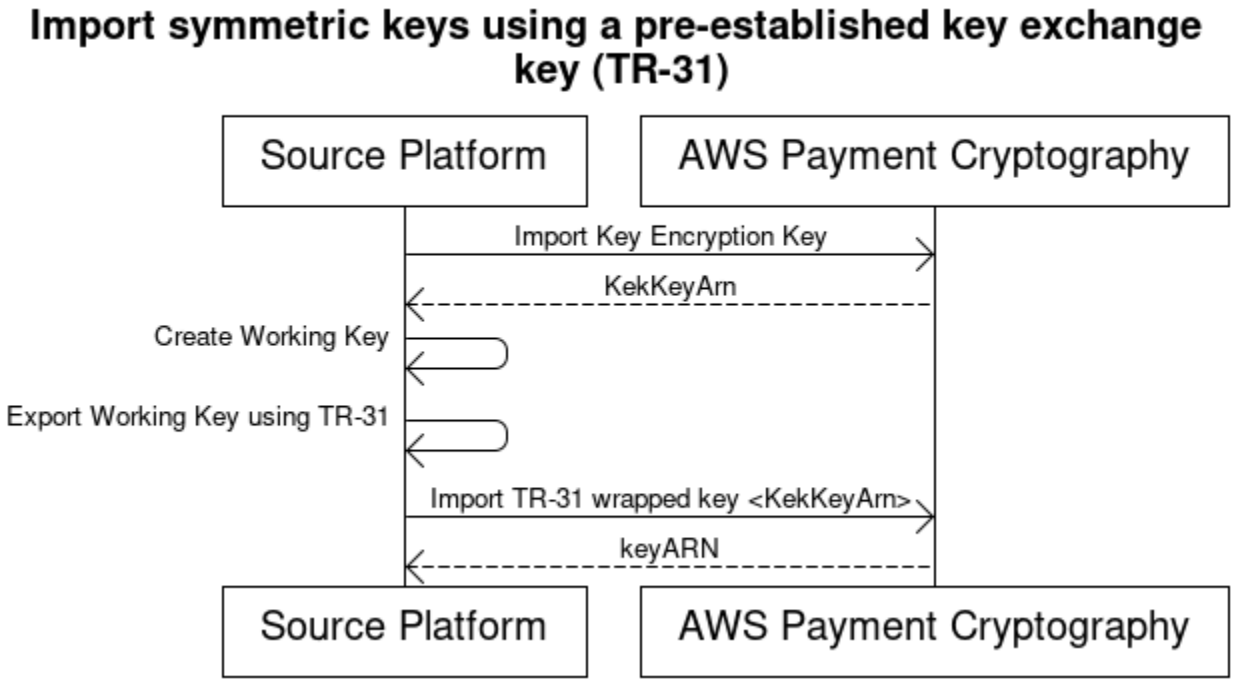
```
$ aws payment-cryptography import-key --cli-input-json file://import-key-
cryptogram.json
```

```
{
      "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. Utiliser la clé importée pour les opérations cryptographiques ou les importations ultérieures

   Si la clé importée KeyUsage était TR31 _K0_ _ KEY ENCRYPTION _KEY, cette clé peut être utilisée pour les importations de clés suivantes à l'aide de la TR-31. Si le type de clé était un autre type (tel que TR31 _D0_ _ _ SYMMETRIC DATA ENCRYPTION _KEY), la clé peut être directement utilisée pour des opérations cryptographiques.

Importation de clés symétriques à l'aide d'une clé d'échange de clés préétablie (TR-31)



Lorsque les partenaires échangent plusieurs clés (ou pour faciliter la rotation des clés), il est courant d'échanger d'abord une clé de chiffrement initiale (KEK) en utilisant des techniques telles que les composants de clés en papier ou, dans le cas de la cryptographie des AWS paiements, à l'aide du TR-34.

Une fois que a KEK est établi, vous pouvez utiliser cette clé pour transporter les clés suivantes (y compris les autresKEKs). AWS La cryptographie des paiements prend en charge ce type d'échange de clés à l'aide du ANSI TR-31, qui est largement utilisé et largement soutenu par les fournisseurs. HSM

1. Clé d'importation Clé de chiffrement (KEK)

   On suppose que vous avez déjà importé votre clé KEK et que vous avez la clé ARN (oukeyAlias) à votre disposition.

2. Créer une clé sur la plateforme source

   Si la clé n'existe pas déjà, créez-la sur la plateforme source. À l'inverse, vous pouvez créer la clé sur AWS Payment Cryptography et utiliser la `export` commande à la place.

3. Exporter la clé depuis la plateforme source

   Lors de l'exportation, assurez-vous de spécifier le format d'exportation TR-31. La plateforme source vous demandera également la clé à exporter et la clé de chiffrement à utiliser.

4. Importation dans la cryptographie des AWS paiements

   Lorsque vous appelez la importKey commande, WrappingKeyIdentifier il doit s'agir de la clé ARN (ou alias) de votre clé de chiffrement et WrappedKeyBlock de la sortie de la plate-forme source.

   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"
        }
    }
```

Importation de clés asymétriques (RSA)

Importation de clés RSA publiques

AWS La cryptographie des paiements prend en charge l'importation de RSA clés publiques sous forme de certificats X.509. Pour importer un certificat, vous devez d'abord importer son certificat racine. Tous les certificats ne doivent pas être expirés au moment de l'importation. Le certificat doit être au PEM format et doit être codé en base64.

1. Importer dans le certificat racine dans la cryptographie des AWS paiements

   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. Importer un certificat à clé publique dans la cryptographie des AWS paiements

   Vous pouvez désormais importer une clé publique. Il existe deux options pour importer des clés publiques. `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`peut être utilisé si le but de la clé est de vérifier les signatures (par exemple lors d'une importation à l'aide du TR-34). `TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTIO`Npeut être utilisé lors du chiffrement de données destinées à être utilisées avec un autre système.

   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"
  }
}
```

Clés d'exportation

Rubriques

- Exportation de clés symétriques
- Exportation de clés asymétriques (RSA)

Exportation de clés symétriques

> **Important**
>
> Les exemples peuvent nécessiter la dernière version de la AWS CLI V2. Avant de commencer, assurez-vous d'avoir effectué la mise à niveau vers la dernière version.

Rubriques

Exporter des clés à l'aide de techniques asymétriques (TR-34)

Vue d'ensemble : Le TR-34 utilise la cryptographie RSA asymétrique pour chiffrer les clés symétriques à des fins d'échange et pour garantir la source des données (signature). Cela garantit à la fois la confidentialité (cryptage) et l'intégrité (signature) de la clé encapsulée. Lors de l'exportation, AWS Payment Cryptography devient l'hôte de distribution des clés (KDH) et le système cible devient le dispositif de réception des clés (KRD).

1. Appelez la commande d'initialisation de l'exportation

   Appelez `get-parameters-for-export` pour initialiser le processus d'exportation. Cela API générera une paire de clés à des fins d'exportation de clés, signera la clé et renverra le certificat et la racine du certificat. En fin de compte, la clé privée générée par cette commande a été utilisée pour signer la charge utile d'exportation. Dans la terminologie du TR-34, cela s'appelle le certificat de KDH signature. Notez que ces certificats sont de courte durée et ne sont destinés qu'à cette fin. Le paramètre ParametersValidUntilTimestamp indique leur durée.

   NOTE: Tous les certificats sont renvoyés dans un format codé en 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. Importer AWS le certificat de cryptographie des paiements dans le système de réception

   Importez la chaîne de certificats fournie à l'étape 1 dans votre système de réception si nécessaire.

3. Générez une paire de clés, créez un certificat public et fournissez la racine du certificat à AWS Payment Cryptography

   Pour garantir la confidentialité de la charge utile transmise, celle-ci est cryptée par l'expéditeur (connu sous le nom d'hôte de distribution de clés ouKDH). La partie destinataire (généralement la vôtre HSM ou celle de vos partenairesHSM) souhaitera générer une clé publique à cette fin, puis créer un certificat de clé publique (x.509) qui pourra être renvoyé à AWS Payment Cryptography. AWS Private CA est une option pour générer des certificats, mais il n'existe aucune restriction quant à l'autorité de certification utilisée.

   Une fois que vous avez obtenu le certificat, vous devez charger le certificat racine dans AWS Payment Cryptography à l'aide des `ImportKey` commandes KeyMaterialType de `ROOT_PUBLIC_KEY_CERTIFICATE` et KeyUsageType de`TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`.

   Ce certificat est TR31 _S0_ _ _ ASYMMETRIC _ KEY FOR DIGITAL _ SIGNATURE car il s'agit de la clé racine utilisée pour signer le certificat feuille. KeyUsageType Les certificats Leaf destinés à l'importation/exportation ne sont pas importés dans AWS Payment Cryptography mais sont transmis en ligne.

   > **Note**
   >
   > Si le certificat racine a déjà été importé, cette étape peut être ignorée.

4. Clé d'exportation des appels

   À la dernière étape, vous appellerez le ExportKey API avec un KeyMaterialType de`TR34_KEY_BLOCK`. Il `certificate-authority-public-key-identifier` s'agira de la clé ARN de l'importation de l'autorité de certification racine à l'étape 3, `WrappingKeyCertificate` du certificat feuille de l'étape 3 et `export-key-identifier` de la clé ARN (ou alias) à exporter. Vous devrez également fournir le jeton d'exportation indiqué à l'étape 1.

Exporter des clés à l'aide de techniques asymétriques (RSAWrap)

Vue d'ensemble : La cryptographie des AWS paiements prend en charge l'RSAencapsulation et le déballage pour l'échange de clés lorsque le TR-34 n'est pas une option disponible par le partenaire. Semblable à la TR-34, cette technique utilise la cryptographie RSA asymétrique pour chiffrer des clés symétriques à des fins d'échange. Cependant, contrairement à la TR-34, cette méthode ne fait pas signer la charge utile par l'expéditeur. De plus, cette technique d'RSAencapsulation n'inclut pas les blocs clés utilisés pour préserver l'intégrité des métadonnées clés pendant le transport.

> **Note**
>
> RSAwrap peut être utilisé pour exporter TDES et AES -128 clés.

1. Générer une RSA clé et un certificat sur le système de réception

   Créez (ou identifiez) une RSA clé qui sera utilisée pour recevoir la clé encapsulée. AWS La cryptographie des paiements exige des clés au format de certificat X.509. Le certificat doit être signé par un certificat racine importé (ou pouvant être importé) dans AWS Payment Cryptography.

2. Installer le certificat public root sur la cryptographie des AWS paiements

```
$ 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. Clé d'exportation des appels

   Vous devez ensuite demander à AWS Payment Cryptography d'exporter votre clé à l'aide de votre certificat Leaf. Vous allez spécifier le ARN certificat racine précédemment importé, le certificat feuille à utiliser pour l'exportation et la clé symétrique à exporter. La sortie sera une version encapsulée (cryptée) binaire codée en hexadécimal de votre clé symétrique.

```
$ 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. Importer la clé dans le système de réception

   De nombreux HSMs systèmes connexes permettent d'importer des clés à l'aide de l'RSAunwrap (y compris la cryptographie des AWS paiements). Pour ce faire, spécifiez la clé publique de l'étape 1 comme certificat (de cryptage) et le format doit être spécifié comme RSA suit : Mode de remplissage = PKCS #1 v2.2 OAEP (avec SHA 256). La terminologie exacte peut varier selonHSM.

   > **Note**
   >
   > AWS La cryptographie des paiements génère la clé encapsulée. hexBinary Vous devrez peut-être convertir le format avant de procéder à l'importation si votre système nécessite une représentation binaire différente, telle que base64.

Exporter des clés symétriques à l'aide d'une clé d'échange de clés préétablie (TR-31)

Lorsque les partenaires échangent plusieurs clés (ou pour faciliter la rotation des clés), il est courant d'échanger d'abord une clé de chiffrement initiale (KEK) en utilisant des techniques telles que les composants de clés en papier ou, dans le cas de la cryptographie des AWS paiements, à l'aide du TR-34. Une fois que a KEK est établi, vous pouvez utiliser cette clé pour transporter les clés suivantes (y compris les autresKEK). AWS La cryptographie des paiements prend en charge ce type d'échange de clés à l'aide du ANSI TR-31, qui est largement utilisé et largement soutenu par les fournisseurs. HSM

1. Clé de chiffrement Exchange Key (KEK)

   On suppose que vous avez déjà échangé votre clé KEK et que vous avez la clé ARN (oukeyAlias) à votre disposition.

2. Créer une clé sur la cryptographie des AWS paiements

   Si la clé n'existe pas déjà, créez-la. Inversement, vous pouvez créer la clé sur l'autre système et utiliser la commande d'importation à la place.

3. Clé d'exportation depuis AWS Payment Cryptography

   Lors de l'exportation, le format sera TR-31. Lorsque vous appelez leAPI, vous spécifiez la clé à exporter et la clé d'encapsulage à utiliser.

```
$ 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. Importer dans votre système

   Vous ou votre partenaire utiliserez l'implémentation de la clé d'importation sur votre système pour importer la clé.

Exporter les clés DUKPT initiales (IPEK/IK)

Lors de l'utilisation DUKPT, une seule clé de dérivation de base (BDK) peut être générée pour un parc de terminaux. Cependant, les terminaux n'ont jamais accès à cet original, BDK mais reçoivent chacun une clé de terminal initiale unique connue sous IPEK le nom de clé initiale (IK). Chacune IPEK est une clé dérivée du BDK et est destinée à être unique par terminal, mais elle est dérivée de l'originalBDK. Les données de dérivation utilisées pour ce calcul sont connues sous le nom de numéro de série de la clé (KSN). Selon X9.24, TDES les 10 octets se composent KSN généralement de 24 bits pour l'identifiant du jeu de clés, de 19 bits pour l'identifiant du terminal et de 21 bits pour le compteur de transactions. En AES effet, les 12 octets se composent KSN généralement de 32 bits pour l'BDKidentifiant, de 32 bits pour l'identifiant de dérivation (ID) et de 32 bits pour le compteur de transactions.

AWS La cryptographie des paiements fournit un mécanisme permettant de générer et d'exporter ces clés initiales. Une fois générées, ces clés peuvent être exportées à l'aide des méthodes TR-31,

TR-34 et wrap. RSA IPEKles clés ne sont pas conservées et ne peuvent pas être utilisées pour des opérations ultérieures sur la cryptographie des AWS paiements

AWS La cryptographie des paiements n'impose pas la division entre les deux premières parties duKSN. Si vous souhaitez enregistrer l'identifiant de dérivation avec leBDK, vous pouvez utiliser la fonction de AWS balises à cette fin.

> **Note**
>
> La partie compteur du KSN (32 bits pour AESDUKPT) n'est pas utilisée pour la dérivation IPEK /IK. Par conséquent, une entrée 12345678901234560001 et 12345678901234569999 produira le même résultat. 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"
    }
}
```

Spécification des en-têtes des blocs-clés lors de l'exportation

AWS La cryptographie des paiements permet de modifier ou d'ajouter des informations sur les blocs de clés lors de l'exportation aux formats ASC TR-31 ou TR-34. Le tableau ci-dessous décrit le format du bloc de touches TR-31 et les éléments qui peuvent être modifiés lors de l'exportation.

Attribut du bloc clé	Objectif	Modifier ou ajouter lors de l'exportation ?	Remarques
ID de version	Représente la méthode utilisée pour protéger le matériau clé sous-jacent. La norme définit les versions A et C (variante clé), version B (dérivation utilisant TDES) et version D (dérivation clé utilisant AES).	Non modifiable	AWS La cryptographie des paiements utilisera la version B lorsque la clé d'encapsulation est TDES et la version D lorsque la clé d'encapsulation l'estAES. Les versions A et C sont obsolètes et ne sont prises en charge que pour les opérations d'importation.
Longueur du bloc clé	Représente la longueur du message restant	Non modifiable	Cette valeur est automatiquement calculée par le service. Notez que le service peut utiliser un rembourrage des touches comme l'exige la spécification, de sorte que la longueur peut ne pas apparaître correcte avant le déchiffrement de la charge utile.
Utilisation de la clé	Représente les objectifs autorisés de la clé, tels que C0 (vérification de la	Non modifiable	

Attribut du bloc clé	Objectif	Modifier ou ajouter lors de l'exportation ?	Remarques
	carte) ou B0 (clé de dérivation de base)		
Algorithm	Représente l'algorithme de la clé sous-jacente. La norme prend en charge plusieurs types de clés tels que T (TDES), A (AES) et E (ECC). AWS La cryptographie des paiements prend actuellement en charge les valeurs T, H et A.	Non modifiable	Ceci est exporté depuis AWS Payment Cryptography tel quel.
Utilisation de la clé	Représente les types d'opérations pour lesquels il peut être utilisé. Les exemples incluent Generate and Verify (C) et Encrypter /Décrypter/Wrap/Unwrap (B)	Modifier*	
Version clé	Représente le numéro de version de la clé si celle-ci est remplacée /pivotée. Il s'agit d'une valeur numérique dont la valeur par défaut est 00 si elle n'est pas spécifiée.	Ajout	

Attribut du bloc clé	Objectif	Modifier ou ajouter lors de l'exportation ?	Remarques
Exportabilité clé	Indique si la clé peut être exportée. N signifie aucune exportabilité, E signifie exportation selon X9.24 (blocs clés), S signifie exportation sous un format de bloc clé ou de bloc non clé.	Modifier*	
Blocs clés facultatifs	Ajout	Les blocs de clé facultatifs sont une série de paires nom/valeur qui peuvent être liées cryptique ment à la clé. L' KeySetidentifiant des DUKPT clés en est un exemple courant. Le format actuel inclut le nombre de blocs optionnels, la longueur de chacun et un bloc de remplissa ge (PB), mais AWS Payment Cryptogra phy les calculera automatiquement en fonction de la paire nom/valeur saisie.	

*Lors de la modification, la nouvelle valeur doit être plus restrictive que la valeur actuelle dans AWS Payment Cryptography. Par exemple, si le mode d'utilisation des clés actuel est Generate=True, Verify=True, lors de l'exportation, vous pouvez le remplacer par Generate=True, Verify=False. De même, si la clé est déjà définie comme non exportable, vous ne pouvez pas la changer en exportable.

Lors de l'exportation de clés, AWS Payment Cryptography applique automatiquement les valeurs actuelles de la clé exportée sans modification. Toutefois, dans certains cas, vous souhaiterez peut-être modifier ou ajouter ces valeurs avant de les envoyer au système de réception, même si cela est autorisé par AWS Payment Cryptography. Par exemple, lors de l'exportation d'une clé vers un terminal de paiement, il est courant de limiter son exportabilité, `Not Exportable` car les terminaux ne sont généralement destinés qu'à importer des clés et ne doivent donc pas exporter de clés par la suite.

Autre exemple : lorsque vous souhaitez transmettre les métadonnées clés associées au système récepteur. Avant le TR-31, la pratique courante consistait à créer une charge utile personnalisée pour envoyer de telles informations, mais avec le TR-31, vous pouvez les lier cryptographiquement à la clé dans un format spécifique. La version clé peut être définie à l'aide du `KeyVersion` champ. Le TR-31/X9.143 spécifie certains en-têtes communs, mais il n'y a aucune restriction quant à l'utilisation d'autres en-têtes tant que cela correspond aux paramètres de cryptographie des AWS paiements et que le système de réception est en mesure de les accepter. Pour plus d'informations sur la spécification des blocs clés lors de l'exportation, veuillez consulter les en-têtes des blocs clés du API guide.

Dans cet exemple, nous exportons une BDK clé (par exemple vers aKIF). Nous spécifions la version de la clé comme 02, en attribuant la valeur KeyExportability à NON _ EXPORTABLE et en fournissant une valeur facultative pour l' KeySetID 00, ABCDEFAB ce qui signifie qu'il s'agit d'une TDES clé (00) et que la clé initiale estABCDEFABCD. Comme les modes d'utilisation des clés ne sont pas spécifiés, cette clé héritera du mode d'utilisation de arn:aws:payment-cryptography:us-east-2:111122223333:key/5rplquuwozodpwsp qui est `DeriveKey = true`

> **Note**
>
> Même si l'exportabilité est définie sur Non exportable dans cet exemple, il KIFsera toujours possible de dériver des clés telles que le IPEK/IK utilisé dans DUKPT et ces clés dérivées seront exportables afin d'être installées sur les appareils. Cela est spécifiquement autorisé par les normes.

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

```
{
        "WrappedKey": {
            "WrappedKeyMaterialFormat": "TR31_KEY_BLOCK",
            "KeyMaterial":
 "D0128B0TX02N0100BI1000ABCDEFABCD1A2C0EADE244321640E94FE3A3C9D33800D47CE64238D9327DDBFE25B9023
            "KeyCheckValue": "A4C9B3",
            "KeyCheckValueAlgorithm": "ANSI_X9_24"
        }
  }
```

Exportation de clés asymétriques (RSA)

Appelez `get-public-key-certificate` pour exporter une clé publique sous forme de certificat. Cela API exportera le certificat ainsi que son certificat racine codés au format base64.

NOTE: Ce n'APIest pas idempotent : les appels suivants peuvent donner lieu à des certificats différents même si la clé sous-jacente est la même.

Example

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

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

Utilisation des alias

Un alias est un nom convivial pour une clé AWS de cryptographie de paiement. Par exemple, un alias vous permet de faire référence à une clé comme `alias/test-key` au lieu de`arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h`.

Vous pouvez utiliser un alias pour identifier une clé dans la plupart des opérations de gestion des clés (plan de contrôle) et dans les opérations cryptographiques (plan de données).

Vous pouvez également autoriser et refuser l'accès à la clé AWS de chiffrement des paiements en fonction de leurs alias sans modifier les politiques ni gérer les subventions. Cette fonctionnalité fait partie de la prise en charge par le service du contrôle d'accès basé sur les attributs ()ABAC.

La puissance des alias provient en grande partie de votre capacité à modifier la clé associée à un alias à tout moment. Les alias peuvent faciliter l'écriture et la maintenance de votre code. Supposons, par exemple, que vous utilisiez un alias pour faire référence à une clé de cryptographie de AWS paiement particulière et que vous souhaitiez modifier la clé de cryptographie AWS de paiement. Dans ce cas, associez simplement l'alias à une autre clé. Il n'est pas nécessaire de modifier le code ou la configuration de l'application.

Les alias facilitent également la réutilisation du même code dans différentes Régions AWS. Créez des alias portant le même nom dans plusieurs régions et associez chaque alias à une clé de cryptographie de AWS paiement dans sa région. Lorsque le code s'exécute dans chaque région, l'alias fait référence à la clé de cryptographie de AWS paiement associée dans cette région.

Vous pouvez créer un alias pour une clé AWS de cryptographie de paiement en utilisant le `CreateAlias`API.

La cryptographie API des AWS paiements permet de contrôler totalement les alias de chaque compte et de chaque région. APICela inclut les opérations permettant de créer un alias (CreateAlias), d'afficher les noms des alias et la clé liée ARN (list-aliases), de modifier la clé de chiffrement des AWS paiements associée à un alias (update-alias) et de supprimer un alias (delete-alias).

Rubriques

- À propos des alias
- Utilisation d'alias dans vos applications
- Connexe APIs

À propos des alias

Découvrez comment fonctionnent les alias dans la cryptographie des AWS paiements.

Un alias est une AWS ressource indépendante

Un alias n'est pas une propriété d'une clé de cryptographie de AWS paiement. Les actions que vous effectuez sur l'alias n'affectent pas la clé qui lui est associée. Vous pouvez créer un alias pour une clé de cryptographie de AWS paiement, puis mettre à jour l'alias afin qu'il soit associé à une autre clé de cryptographie AWS de paiement. Vous pouvez même supprimer l'alias sans aucun effet sur la clé de cryptographie AWS de paiement associée. Si vous supprimez une clé AWS de chiffrement des paiements, tous les alias associés à cette clé ne seront plus attribués.

Si vous spécifiez un alias comme ressource dans une IAM politique, celle-ci fait référence à l'alias, et non à la clé de cryptographie AWS de paiement associée.

Chaque alias possède un nom convivial

Lorsque vous créez un alias, vous spécifiez le nom de l'alias préfixé par`alias/`. Par exemple `alias/test_1234`

Chaque alias est associé à une clé AWS de cryptographie de paiement à la fois

L'alias et sa clé AWS de cryptographie de paiement doivent se trouver dans le même compte et dans la même région.

Une clé AWS de cryptographie de paiement peut être associée à plusieurs alias simultanément, mais chaque alias ne peut être associé qu'à une seule clé

Par exemple, cette `list-aliases` sortie indique que l'`alias/sampleAlias1`alias est associé à une seule clé de cryptographie de AWS paiement cible, qui est représentée par la `KeyArn` propriété.

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

```
{
  "Aliases": [
      {
          "AliasName": "alias/sampleAlias1",
```

```
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h"
        }
    ]
}
```

Plusieurs alias peuvent être associés à la même clé de cryptographie AWS de paiement

Par exemple, vous pouvez associer les `alias/sampleAlias2` alias `alias/sampleAlias1;` et à la même clé.

```
$ 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"
            }
        ]
    }
```

Un alias doit être unique pour un compte et une région donnés

Par exemple, vous ne pouvez avoir qu'un seul alias `alias/sampleAlias1` dans chaque compte et région. Les alias font la distinction majuscules/majuscules, mais nous vous déconseillons d'utiliser des alias dont les majuscules ne diffèrent que par leur mise en majuscules, car ils peuvent être sujets à des erreurs. Vous ne pouvez pas modifier un nom d'alias. Toutefois, vous pouvez supprimer l'alias et créer un alias avec le nom souhaité.

Vous pouvez créer des alias avec le même nom dans des régions différentes.

Par exemple, vous pouvez avoir un alias `alias/sampleAlias2` dans l'est des États-Unis (Virginie du Nord) et un alias `alias/sampleAlias2` dans l'ouest des États-Unis (Oregon).

Chaque alias serait associé à une clé de cryptographie de AWS paiement dans sa région. Si votre code fait référence à un nom d'alias comme `alias/finance-key`, vous pouvez l'exécuter dans plusieurs régions. Dans chaque région, il utilise un alias différent/2sampleAlias. Pour plus de détails, consultez Utilisation d'alias dans vos applications.

Vous pouvez modifier la clé AWS de chiffrement des paiements associée à un alias

Vous pouvez utiliser cette `UpdateAlias` opération pour associer un alias à une autre clé de cryptographie de AWS paiement. Par exemple, si l'`alias/sampleAlias2`alias est associé à la clé de chiffrement des `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h` AWS paiements, vous pouvez le mettre à jour afin qu'il soit associé à la `arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi` clé.

> **Warning**
>
> AWS La cryptographie des paiements ne permet pas de vérifier que les anciennes et les nouvelles clés possèdent toutes les mêmes attributs, tels que l'utilisation des clés. La mise à jour avec un autre type de clé peut entraîner des problèmes dans votre application.

Certaines clés n'ont pas d'alias

Un alias est une fonctionnalité facultative et toutes les clés n'auront pas d'alias, sauf si vous choisissez de gérer votre environnement de cette manière. Les clés peuvent être associées à des alias à l'aide de la `create-alias` commande. Vous pouvez également utiliser l'opération update-alias pour modifier la clé de chiffrement des AWS paiements associée à un alias et l'opération delete-alias pour supprimer un alias. Par conséquent, certaines clés de chiffrement des AWS paiements peuvent avoir plusieurs alias, tandis que d'autres n'en ont aucun.

Associer une clé à un alias

Vous pouvez mapper une clé (représentée par unARN) à un ou plusieurs alias à l'aide de la `create-alias` commande. Cette commande n'est pas idempotente. Pour mettre à jour un alias, utilisez la commande 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"
    }
}
```

Utilisation d'alias dans vos applications

Vous pouvez utiliser un alias pour représenter une clé de chiffrement des AWS paiements dans le code de votre application. Le `key-identifier` paramètre utilisé dans les opérations relatives aux données de cryptographie des AWS paiements, ainsi que dans d'autres opérations telles que les clés de liste, accepte un alias ou un aliasARN.

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

Lorsque vous utilisez un aliasARN, n'oubliez pas que le mappage d'alias vers une clé de cryptographie de AWS paiement est défini dans le compte qui possède la clé de cryptographie de AWS paiement et qu'il peut différer d'une région à l'autre.

L'une des utilisations les plus performantes des alias est au niveau des applications qui s'exécutent dans plusieurs Régions AWS.

Vous pouvez créer une version différente de votre application dans chaque région ou utiliser un dictionnaire, une configuration ou une instruction switch pour sélectionner la clé de cryptographie de AWS paiement adaptée à chaque région. Mais il peut être plus facile de créer un alias portant le même nom d'alias dans chaque région. Rappelez-vous que ne nom de l'alias est sensible à la casse.

Connexe APIs

Balises

Les balises sont des paires clé/valeur qui agissent comme des métadonnées pour organiser vos clés AWS de cryptographie de paiement. Ils peuvent être utilisés pour identifier les clés de manière flexible ou pour regrouper une ou plusieurs clés.

Obtenez des clés

Une clé AWS de cryptographie de paiement représente une unité unique de matériel cryptographique et ne peut être utilisée que pour les opérations cryptographiques de ce service. Le GetKeys API prend une KeyIdentifier entrée et renvoie les attributs immuables et mutables de la clé mais ne contient aucun matériel cryptographique.

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"
  }
}
```

Obtenir la clé publique/le certificat associé à une paire de clés

Get Public Key/Certificate renvoie la clé publique indiquée par le. `KeyArn` Il peut s'agir de la partie clé publique d'une paire de clés générée par AWS Payment Cryptography ou d'une clé publique précédemment importée. Le cas d'utilisation le plus courant consiste à fournir la clé publique à un service externe qui cryptera les données. Ces données peuvent ensuite être transmises à une application utilisant la cryptographie des AWS paiements et les données peuvent être déchiffrées à l'aide de la clé privée sécurisée dans la cryptographie des paiements. AWS

Le service renvoie les clés publiques sous forme de certificat public. Le API résultat contient l'autorité de certification et le certificat de clé publique. Les deux éléments de données sont codés en base64.

> **Note**
>
> Le certificat public renvoyé est destiné à être de courte durée et n'est pas destiné à être idempotent. Vous pouvez recevoir un certificat différent à chaque API appel, même si la clé publique elle-même reste inchangée.

Example

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

```
{
  "KeyCertificate":
 "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ0lSQUo1OWd2VkpDd3dlYldMNldYZEpYY
  "KeyCertificateChain":
 "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUY0VENDQThtZ0F3SUJBZ0lSQUtlN2piaHFKZjJPd3FGUWI5c3VuO
}
```

Clés de balisage

Dans la cryptographie des AWS paiements, vous pouvez ajouter des balises à une clé de chiffrement des AWS paiements lorsque vous créez une clé, et étiqueter ou décocher les clés existantes, sauf si elles sont en attente de suppression. Les balises sont facultatives, mais peuvent être très utiles.

Pour obtenir des informations générales sur les balises, notamment les meilleures pratiques, les stratégies de balisage, ainsi que le format et la syntaxe des balises, consultez les AWS ressources de balisage dans le. Référence générale d'Amazon Web Services

Rubriques

À propos des balises dans la cryptographie des AWS paiements

Une balise est une étiquette de métadonnées facultative que vous pouvez attribuer (ou AWS attribuer) à une AWS ressource. Chaque balise est constituée d'une clé de balise et d'un valeur de balise, qui sont toutes deux des chaînes sensibles à la casse. La valeur de balise peut être une chaîne vide (nulle). Chaque balise d'une ressource doit avoir une clé de balise différente, mais vous pouvez ajouter la même balise à plusieurs AWS ressources. Chaque ressource peut avoir jusqu'à 50 balises créées par l'utilisateur.

N'incluez pas d'informations confidentielles ou sensibles dans la clé de balise ou la valeur de balise. Les tags sont accessibles à de nombreuses personnes Services AWS, y compris pour la facturation.

Dans AWS Payment Cryptography, vous pouvez ajouter des balises à une clé lorsque vous créez la clé, et étiqueter ou décocher les clés existantes, sauf si elles sont en attente de suppression. Vous ne pouvez pas baliser les alias. Les balises sont facultatives, mais peuvent être très utiles.

Par exemple, vous pouvez ajouter une `"Project"="Alpha"` balise à toutes les clés de chiffrement des AWS paiements et à tous les compartiments Amazon S3 que vous utilisez pour le projet Alpha. Un autre exemple consiste à ajouter une `"BIN"="20130622"` étiquette à toutes les clés associées à un numéro d'identification bancaire spécifique (BIN).

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

Pour obtenir des informations générales sur les balises, notamment leur format et leur syntaxe, consultez la section AWS Ressources de balisage dans le Référence générale d'Amazon Web Services.

Les balises vous permettent d'effectuer les actions suivantes :

- Identifiez et organisez vos AWS ressources. De nombreux AWS services prennent en charge le balisage. Vous pouvez donc attribuer le même tag aux ressources de différents services pour indiquer que les ressources sont liées. Par exemple, vous pouvez attribuer la même balise à une clé de cryptographie de AWS paiement et à un volume ou à un AWS Secrets Manager secret Amazon Elastic Block Store (AmazonEBS). Vous pouvez également utiliser des balises pour identifier les clés d'automatisation.
- Suivez vos AWS coûts. Lorsque vous ajoutez des balises à vos AWS ressources, AWS génère un rapport de répartition des coûts avec l'utilisation et les coûts agrégés par balises. Vous pouvez utiliser cette fonctionnalité pour suivre les coûts AWS de cryptographie des paiements pour un projet, une application ou un centre de coûts.

  Pour en savoir plus sur l'utilisation des balises pour la répartition des coûts, veuillez consulter Utilisation des balises de répartition des coûts dans le Guide de l'utilisateur AWS Billing . Pour obtenir des informations sur les règles des clés et valeurs de balise, veuillez consulter Restrictions encadrant les balises définies par l'utilisateur dans le Guide de l'utilisateur AWS Billing .

- Contrôlez l'accès à vos AWS ressources. L'autorisation et le refus d'accès aux clés en fonction de leurs balises font partie de la prise en charge de la cryptographie des AWS paiements pour le contrôle d'accès basé sur les attributs (). ABAC Pour plus d'informations sur le contrôle de l'accès à la cryptographie des AWS paiements en fonction de leurs balises, consultezAutorisation basée sur les balises AWS de cryptographie des paiements. Pour des informations plus générales sur l'utilisation de balises pour contrôler l'accès aux AWS ressources, consultez la section Contrôle de l'accès aux AWS ressources à l'aide de balises de ressources dans le guide de IAM l'utilisateur.

AWS La cryptographie des paiements écrit une entrée dans votre AWS CloudTrail journal lorsque vous utilisez les ListTagsForResource opérations TagResource UntagResource, ou.

Afficher les tags clés dans la console

Pour afficher les balises dans la console, vous devez disposer d'une autorisation de balisage sur la clé, conformément à une IAM politique qui inclut la clé. Vous avez besoin de ces autorisations en plus des autorisations pour afficher les clés dans la console.

Gestion des tags clés avec des API opérations

Vous pouvez utiliser le chiffrement des AWS paiements API pour ajouter, supprimer et répertorier les balises des clés que vous gérez. Ces exemples utilisent l'AWS Command Line Interface (AWS CLI), mais vous pouvez utiliser n'importe quel langage de programmation pris en charge. Vous ne pouvez pas étiqueter Clés gérées par AWS.

Pour ajouter, modifier, afficher et supprimer des balises pour une clé, vous devez disposer des autorisations requises. Pour plus de détails, consultez Contrôle de l'accès aux balises.

Rubriques

- CreateKey: Ajouter des balises à une nouvelle clé
- TagResource: ajouter ou modifier les balises d'une clé
- ListResourceTags: Récupère les tags d'une clé
- UntagResource: Supprimer les tags d'une clé

CreateKey: Ajouter des balises à une nouvelle clé

Vous pouvez ajouter des balises lorsque vous créez une clé. Pour définir les balises, utilisez le `Tags` paramètre de l'CreateKeyopération.

Pour ajouter des balises lors de la création d'une clé, l'appelant doit être `payment-cryptography:TagResource` autorisé dans une IAM politique. L'autorisation doit au minimum couvrir toutes les clés du compte et de la région. Pour plus de détails, consultez Contrôle de l'accès aux balises.

La valeur du paramètre `Tags` de `CreateKey` est une collection de paires de clés et de valeurs de balises sensibles à la casse. Chaque étiquette d'une clé doit porter un nom de balise différent. La valeur de balise peut être une chaîne vide ou nulle.

Par exemple, la AWS CLI commande suivante crée une clé de chiffrement symétrique avec une `Project:Alpha` balise. Lorsque vous spécifiez plusieurs paires clé-valeur, utilisez un espace pour séparer chaque paire.

```
$ 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"}]'
```

Lorsque cette commande aboutit, elle renvoie un `Key` objet contenant des informations sur la nouvelle clé. Cependant, l'objet `Key` n'inclut pas les balises. Pour obtenir les balises, utilisez l'ListResourceTagsopération.

TagResource: ajouter ou modifier les balises d'une clé

L'TagResourceopération ajoute une ou plusieurs balises à une clé. Vous ne pouvez pas utiliser cette opération pour ajouter ou modifier des balises dans un autre Compte AWS.

Pour ajouter une balise, spécifiez de nouvelles clé et valeur de balises. Pour modifier une balise, spécifiez une clé de balise existante et une nouvelle valeur de balise. Chaque étiquette d'une clé doit avoir une clé de balise différente. La valeur de balise peut être une chaîne vide ou nulle.

Par exemple, la commande suivante ajoute **`UseCase`** des **`BIN`** balises à un exemple de clé.

```
$ 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"}]'
```

Lorsque cette commande aboutit, elle ne renvoie pas de sortie. Pour afficher les balises d'une clé, utilisez l'ListResourceTagsopération.

Vous pouvez également utiliser TagResource pour modifier la valeur de balise d'une balise existante. Pour remplacer une valeur de balise, spécifiez la même clé de balise avec une valeur différente. Les balises non répertoriées dans une commande de modification ne sont ni modifiées ni supprimées.

Par exemple, cette commande modifie la valeur de la balise `Project` de `Alpha` en `Noe`.

La commande renverra http/200 sans contenu. Pour voir vos modifications, utilisez `ListTagsForResource`

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

ListResourceTags: Récupère les tags d'une clé

L'ListResourceTagsopération obtient les balises d'une clé. Le paramètre `ResourceArn` (keyArn oukeyAlias) est obligatoire. Vous ne pouvez pas utiliser cette opération pour afficher les balises des touches d'une autre manière Compte AWS.

Par exemple, la commande suivante permet d'obtenir les balises d'un exemple de clé.

```
$ 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: Supprimer les tags d'une clé

L'UntagResourceopération supprime les balises d'une clé. Pour identifier les balises à supprimer, spécifiez les clés de balise. Vous ne pouvez pas utiliser cette opération pour supprimer des balises d'une autre clé Compte AWS.

Lorsque l'opération `UntagResource` réussit, elle ne renvoie aucune sortie. De plus, si la clé de balise spécifiée n'est pas trouvée sur la clé, elle ne lance pas d'exception ni ne renvoie de réponse. Pour vérifier que l'opération a fonctionné, ListResourceTagsutilisez-la.

Par exemple, cette commande supprime le **`Purpose`** tag et sa valeur de la clé spécifiée.

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

Contrôle de l'accès aux balises

Pour ajouter, afficher et supprimer des balises à l'aide deAPI, les directeurs doivent disposer d'autorisations de balisage dans IAM les politiques.

Vous pouvez également limiter ces autorisations en utilisant des clés de condition AWS globales pour les balises. Dans le AWS domaine de la cryptographie des paiements, ces conditions peuvent contrôler l'accès aux opérations de marquage, telles que TagResourceet. UntagResource

Pour des exemples de politiques et de plus amples informations, consultez la section Contrôle de l'accès en fonction des clés de balise dans le guide de IAM l'utilisateur.

Les autorisations de création et de gestion de balises fonctionnent comme suit.

cryptographie des paiements : TagResource

Autorise les principaux à ajouter ou à modifier des balises. Pour ajouter des balises lors de la création d'une clé, le principal doit disposer d'une autorisation dans le cadre d'une IAM politique qui n'est pas limitée à des clés spécifiques.

cryptographie des paiements : ListTagsForResource

Permet aux principaux d'afficher les balises sur les clés.

cryptographie des paiements : UntagResource

Permet aux principaux de supprimer les balises des clés.

Autorisations de balises dans les politiques

Vous pouvez fournir des autorisations de balisage dans une politique ou IAM une politique clé. Par exemple, l'exemple de politique de clé suivant donne à certains utilisateurs l'autorisation de taguer la

clé. Il accorde à tous les utilisateurs qui peuvent endosser les rôles Administrateur ou Développeur d'exemple l'autorisation d'afficher les balises.

```
{
  "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": "*"
    }
  ]
}
```

Pour autoriser les directeurs à baliser plusieurs clés, vous pouvez utiliser une IAM politique. Pour que cette politique soit efficace, la politique clé pour chaque clé doit autoriser le compte à utiliser des IAM politiques pour contrôler l'accès à la clé.

Par exemple, la IAM politique suivante permet aux principaux de créer des clés. Cela leur permet également de créer et de gérer des balises sur toutes les clés du compte spécifié. Cette combinaison permet aux principaux d'utiliser le paramètre tags de l'CreateKeyopération pour ajouter des balises à une clé lors de sa création.

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

Limitation des autorisations de balises

Vous pouvez limiter les autorisations d'étiquetage en utilisant des conditions de politique. Les conditions de politique suivantes peuvent être appliquées aux autorisations `payment-cryptography:TagResource` et `payment-cryptography:UntagResource`. Par exemple, vous pouvez utiliser la condition `aws:RequestTag/`*`tag-key`* pour permettre à un principal d'ajouter uniquement des balises particulières, ou empêcher un principal d'ajouter des balises avec des clés de balise particulières.

- lois : RequestTag

- aws :ResourceTag/tag-key (IAMpolitiques uniquement)
- lois : TagKeys

Lorsque vous utilisez des balises pour contrôler l'accès aux clés, il est recommandé d'utiliser la touche de `aws:TagKeys` condition `aws:RequestTag/`*`tag-key`* ou pour déterminer quelles balises (ou clés de balise) sont autorisées.

Par exemple, la IAM politique suivante est similaire à la précédente. Toutefois, cette politique permet aux principaux de créer des balises (`TagResource`) et de supprimer des balises `UntagResource` uniquement pour les balises avec une clé de balise `Project`.

Étant donné que `TagResource` les `UntagResource` demandes peuvent inclure plusieurs balises, vous devez spécifier un opérateur `ForAllValues` ou un `ForAnyValue` ensemble avec la TagKeys condition aws :. L'opérateur `ForAnyValue` exige qu'au moins l'une des clés de balise dans la demande corresponde à l'une des clés de balise dans la politique. L'opérateur `ForAllValues` exige que toutes les clés de balise dans la demande correspondent à l'une des clés de balise dans la politique. L'`ForAllValues`opérateur renvoie également `true` si la demande ne contient aucune balise, mais TagResource UntagResource échoue si aucune balise n'est spécifiée. Pour plus de détails sur les opérateurs définis, voir Utiliser plusieurs clés et valeurs dans le Guide de IAM l'utilisateur.

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

Utilisation de balises pour contrôler l'accès aux clés

Vous pouvez contrôler l'accès à la cryptographie des AWS paiements en fonction des balises figurant sur la clé. Par exemple, vous pouvez rédiger une IAM politique permettant aux principaux d'activer et de désactiver uniquement les clés dotées d'une balise particulière. Vous pouvez également utiliser une IAM politique pour empêcher les principaux d'utiliser des clés dans des opérations cryptographiques à moins que la clé ne comporte une balise particulière.

Cette fonctionnalité fait partie de la prise en charge de la cryptographie des AWS paiements pour le contrôle d'accès basé sur les attributs (). ABAC Pour plus d'informations sur l'utilisation de balises pour contrôler l'accès aux AWS ressources, voir À quoi ABAC ça sert AWS ? et le contrôle de l'accès aux AWS ressources à l'aide de balises de ressources dans le guide de IAM l'utilisateur.

> **Note**

AWS La cryptographie des paiements prend en charge la clé contextuelle de condition globale aws :ResourceTag/tag-key, qui vous permet de contrôler l'accès aux clés en fonction des balises figurant sur la clé. Comme plusieurs clés peuvent avoir le même tag, cette fonctionnalité vous permet d'appliquer l'autorisation à un ensemble de clés sélectionné. Vous pouvez également modifier facilement les clés de l'ensemble en modifiant leurs balises.

Dans la cryptographie des AWS paiements, la clé de `aws:ResourceTag/`*`tag-key`* condition n'est prise en charge que dans IAM les politiques. Il n'est pas pris en charge dans les politiques clés, qui ne s'appliquent qu'à une seule touche, ni dans les opérations qui n'utilisent pas de clé particulière, telles que les ListAliasesopérations ListKeysor.

Le contrôle de l'accès à l'aide de balises offre un moyen simple, évolutif et flexible de gérer les autorisations. Toutefois, s'il n'est pas correctement conçu et géré, il peut autoriser ou refuser l'accès à vos clés par inadvertance. Si vous utilisez des balises pour contrôler l'accès, tenez compte des pratiques suivantes.

- Utilisez des balises pour renforcer la bonne pratique de l'accès le moins privilégié. IAMAccordez aux directeurs uniquement les autorisations dont ils ont besoin sur les clés qu'ils doivent utiliser ou gérer. Par exemple, utilisez des balises pour étiqueter les clés utilisées pour un projet. Donnez ensuite à l'équipe du projet l'autorisation d'utiliser uniquement les clés portant le tag du projet.
- Soyez prudent lorsque vous donnez aux principaux les autorisations `payment-cryptography:TagResource` et `payment-cryptography:UntagResource` qui leur permettent d'ajouter, de modifier et de supprimer des balises. Lorsque vous utilisez des balises pour contrôler l'accès aux clés, la modification d'une balise peut autoriser les principaux à utiliser des clés qu'ils n'étaient pas autorisés à utiliser autrement. Il peut également refuser l'accès aux clés dont les autres directeurs ont besoin pour faire leur travail. Les administrateurs clés qui ne sont pas autorisés à modifier les politiques clés ou à créer des autorisations peuvent contrôler l'accès aux clés s'ils sont autorisés à gérer les balises.

  Dans la mesure du possible, utilisez une condition de politique, telle que `aws:RequestTag/`*`tag-key`* ou `aws:TagKeys` pour limiter les autorisations de balisage d'un principal à des balises ou à des modèles de balises spécifiques sur des clés spécifiques.
- Passez en revue les principes Compte AWS qui disposent actuellement d'autorisations de balisage et de débalisage et ajustez-les si nécessaire. IAMles politiques peuvent autoriser les autorisations de balisage et de débalisage sur toutes les clés. Par exemple, la politique gérée par l'administrateur permet aux principaux de baliser, de débaliser et de répertorier les balises sur toutes les clés.
- Avant de définir une politique qui dépend d'une balise, passez en revue les balises figurant sur les clés de votre Compte AWS. Assurez-vous que votre politique s'applique uniquement aux balises que vous avez l'intention d'inclure. Utilisez CloudTrail les journaux et les CloudWatch alarmes pour vous avertir des modifications de balises susceptibles d'affecter l'accès à vos clés.
- Les conditions de politique de balise utilisent la correspondance de modèles ; elles ne sont pas liées à une instance particulière d'une balise. Une politique qui utilise des clés de condition basées sur des balises affecte toutes les balises nouvelles et existantes qui correspondent au modèle. Si vous supprimez et recréez une balise qui correspond à une condition de politique, la condition s'applique à la nouvelle balise, comme elle l'a fait pour l'ancienne.

Par exemple, considérez la IAM politique suivante. Cela permet aux principaux d'appeler les opérations de déchiffrement uniquement sur les clés de votre compte correspondant à la région des États-Unis Est (Virginie du Nord) et dotées d'une `"Project"="Alpha"` étiquette. Vous pouvez attacher cette politique à des rôles dans l'exemple de projet 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"
        }
      }
    }
  ]
}
```

L'exemple de IAM politique suivant permet aux principaux d'utiliser n'importe quelle clé du compte pour certaines opérations cryptographiques. Mais il interdit aux donneurs d'ordre d'utiliser ces opérations cryptographiques sur des clés comportant ou non `"Type"` une `"Type"="Reserved"` étiquette.

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

Comprendre les principaux attributs de la clé AWS de cryptographie des paiements

L'un des principes d'une bonne gestion des clés est que les clés ont une portée appropriée et ne peuvent être utilisées que pour des opérations autorisées. Ainsi, certaines touches ne peuvent être créées qu'avec certains modes d'utilisation des touches. Dans la mesure du possible, cela correspond aux modes d'utilisation disponibles tels que définis par le TR-31.

Bien que la cryptographie des AWS paiements vous empêche de créer des clés non valides, des combinaisons valides sont fournies ici pour votre commodité.

Clés symétriques

- TR31BASE_B0_ _ _ DERIVATION KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31_C0_ _ _ CARD VERIFICATION KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_D0_ _ _ _ SYMMETRIC DATA ENCRYPTION KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée de modes d'utilisation clés : {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_E0_ _ _ _ EMV MKEY APP CRYPTOGRAMS
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31EMV_E1_ _ _ MKEY CONFIDENTIALITY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31EMV_E2_ _ _ MKEY INTEGRITY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31_E4_ _ _ _ EMV MKEY DYNAMIC NUMBERS
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES

  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31_E5_ _ _ _ EMV MKEY CARD PERSONALIZATION
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31EMV_E6_ _ _ MKEY OTHER
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31KEY_K0_ _ _ ENCRYPTION KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée de modes d'utilisation clés : {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_K1_ _ _ _ KEY BLOCK PROTECTION KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée de modes d'utilisation clés : {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_M1_ ISO _9797_1_ _ MAC KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3 KEY
  - Combinaison autorisée des principaux modes d'utilisation : {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_M3_ ISO _9797_3_ _ MAC KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3 KEY
  - Combinaison autorisée des principaux modes d'utilisation : {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_M6_ ISO _9797_5_ _ CMAC KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}

- TR31_M7_ _ HMAC KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31PIN_P0_ _ _ ENCRYPTION KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée de modes d'utilisation clés : {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_V1_ _ _ _ IBM3624 PIN VERIFICATION KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31_V2_ _ _ _ VISA PIN VERIFICATION KEY
  - Algorithmes clés autorisés : TDES _2KEY, TDES _3, AES _128KEY, _192, AES _256 AES
  - Combinaison autorisée des principaux modes d'utilisation : {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}

Clés asymétriques

- TR31_D1_ _ _ _ ASYMMETRIC _ KEY FOR DATA ENCRYPTION
  - Algorithmes clés autorisés : RSA _2048, _3072, RSA _4096 RSA
  - Combinaison autorisée de modes d'utilisation clés : {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}
  - NOTE: : {Encrypt = true, Wrap = true} est la seule option valide lors de l'importation d'une clé publique destinée à chiffrer des données ou à encapsuler une clé
- TR31_S0_ _ _ _ ASYMMETRIC _ KEY FOR DIGITAL SIGNATURE
  - Algorithmes clés autorisés : RSA _2048, _3072, RSA _4096 RSA
  - Combinaison autorisée des principaux modes d'utilisation : {Sign = true}, {Verify = true}
  - NOTE: : {Verify = true} est la seule option valide lors de l'importation d'une clé destinée à la signature, telle qu'un certificat racine, un certificat intermédiaire ou des certificats de signature pour le TR-34.

Opérations relatives aux données

Une fois que vous avez défini une clé AWS de chiffrement des paiements, celle-ci peut être utilisée pour effectuer des opérations cryptographiques. Les différentes opérations exécutent différents types d'activités, allant du chiffrement au hachage, en passant par des algorithmes spécifiques au domaine tels que la génération de CVV2.

Les données cryptées ne peuvent pas être déchiffrées sans la clé de déchiffrement correspondante (clé symétrique ou clé privée selon le type de chiffrement). De même, les algorithmes de hachage et spécifiques à un domaine ne peuvent pas être vérifiés sans la clé symétrique ou la clé publique.

Pour plus d'informations sur les types de clés valides pour des opérations spécifiques, reportez-vous à la section Clés valides pour les opérations cryptographiques

> **Note**
>
> Nous recommandons d'utiliser les données de test dans un environnement hors production. L'utilisation de clés et de données de production (PAN, ID BDK, etc.) dans un environnement hors production peut avoir un impact sur votre périmètre de conformité, par exemple pour les normes PCI DSS et PCI P2PE.

Rubriques

Chiffrer, déchiffrer et rechiffrer les données

Les méthodes de chiffrement et de déchiffrement peuvent être utilisées pour chiffrer ou déchiffrer des données à l'aide de diverses techniques symétriques et asymétriques, notamment TDES, AES et RSA. Ces méthodes prennent également en charge les clés dérivées à l'aide des techniques DUKPT

et EMV. Dans les cas d'utilisation où vous souhaitez sécuriser les données sous une nouvelle clé sans exposer les données sous-jacentes, la ReEncrypt commande peut également être utilisée.

> **Note**
>
> Lorsque vous utilisez les fonctions de chiffrement/déchiffrement, toutes les entrées sont supposées être en hexadécimal. Par exemple, une valeur de 1 sera saisie sous la forme 31 (hexadécimal) et un t minuscule est représenté par 74 (hexadécimal). Toutes les sorties sont également en HexBinary.

Pour plus de détails sur toutes les options disponibles, veuillez consulter le guide de l'API pour le chiffrement, le déchiffrement et le rechiffrement.

Rubriques

- Chiffrer des données
- Déchiffrer des données

Chiffrer des données

L'`Encrypt DataAPI` est utilisée pour chiffrer les données à l'aide de clés de chiffrement symétriques et asymétriques ainsi que de clés dérivées du DUKPT et de l'EMV. Différents algorithmes et variantes sont pris en charge`TDES`, notamment, `RSA` et`AES`.

Les entrées principales sont la clé de chiffrement utilisée pour chiffrer les données, les données en texte brut au format HexBinary à chiffrer et les attributs de chiffrement tels que le vecteur d'initialisation et le mode pour les chiffrements par blocs tels que TDES. Les données en texte brut doivent être exprimées en multiples de 8 octets pour`TDES`, 16 octets pour `AES` et de la longueur de la clé dans le cas de`RSA`. Les entrées clés symétriques (TDES, AES, DUKPT, EMV) doivent être complétées dans les cas où les données d'entrée ne répondent pas à ces exigences. Le tableau suivant indique la longueur maximale du texte brut pour chaque type de clé et le type de remplissage que vous définissez `EncryptionAttributes` pour les clés RSA.

Type de rembourrage	RSA_2048	RSA_3072	RSA_4096
`OAEP SHA1`	428	684	940

Type de rembourrage	RSA_2048	RSA_3072	RSA_4096
OAEP SHA256	380	636	892
OAEP SHA512	252	508	764
PKCS1	488	744	1 000
None	488	744	1 000

Les sorties principales incluent les données chiffrées sous forme de texte chiffré au format HexBinary et la valeur de la somme de contrôle pour la clé de chiffrement. Pour plus de détails sur toutes les options disponibles, veuillez consulter le guide de l'API pour Encrypt.

Exemples

- Chiffrer les données à l'aide d'une clé symétrique AES
- Chiffrer les données à l'aide de la clé DUKPT
- Chiffrer les données à l'aide d'une clé symétrique dérivée de l'EMV
- Chiffrer les données à l'aide d'une clé RSA

Chiffrer les données à l'aide d'une clé symétrique AES

> **Note**
>
> Tous les exemples supposent que la clé correspondante existe déjà. Les clés peuvent être créées à l'aide de l'CreateKeyopération ou importées à l'aide de l'ImportKeyopération.

Example

Dans cet exemple, nous chiffrerons les données en texte brut à l'aide d'une clé symétrique créée à l'aide de l'CreateKeyopération ou importée à l'aide de l'opération. ImportKey Pour cette opération, la clé doit être KeyModesOfUse réglée sur `Encrypt` et KeyUsage définie sur`TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`. Consultez la section Clés pour les opérations cryptographiques pour plus d'options.

```
$ aws payment-cryptography-data encrypt-data --key-identifier arn:aws:payment-
cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi --plain-text
 313233343132333431323334313233343132333431323334 --encryption-attributes 'Symmetric={Mode=CBC}'
```

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

Chiffrer les données à l'aide de la clé DUKPT

Example

Dans cet exemple, nous allons chiffrer les données en texte brut à l'aide d'une clé DUKPT. AWS Supports de cryptographie des paiements TDES et clés AES DUKPT. Pour cette opération, la clé doit être KeyModesOfUse réglée sur `DeriveKey` et KeyUsage définie sur`TR31_B0_BASE_DERIVATION_KEY`. Consultez la section Clés pour les opérations cryptographiques pour plus d'options.

```
$ aws payment-cryptography-data encrypt-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi
 --plain-text 313233343132333431323334313233343132333431323334 --encryption-attributes
 'Dukpt={KeySerialNumber=FFFF9876543210E00001}'
```

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

Chiffrer les données à l'aide d'une clé symétrique dérivée de l'EMV

Example

Dans cet exemple, nous allons chiffrer des données en texte clair à l'aide d'une clé symétrique dérivée de l'EMV qui a déjà été créée. Vous pouvez utiliser une commande comme celle-ci pour envoyer des données vers une carte EMV. Pour cette opération, la clé doit être KeyModesOfUse définie sur `Derive` et KeyUsage définie sur `TR31_E1_EMV_MKEY_CONFIDENTIALITY` ou`TR31_E6_EMV_MKEY_OTHER`. Reportez-vous à la section Clés pour les opérations cryptographiques pour plus de détails.

```
$ 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"
}
```

Chiffrer les données à l'aide d'une clé RSA

Example

Dans cet exemple, nous allons chiffrer les données en texte brut à l'aide d'une clé publique RSA importée à l'aide de l'opération. ImportKey Pour cette opération, la clé doit être KeyModesOfUse réglée sur `Encrypt` et KeyUsage définie sur`TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION`. Consultez la section Clés pour les opérations cryptographiques pour plus d'options.

Pour le PKCS #7 ou d'autres systèmes de rembourrage non pris en charge actuellement, veuillez en faire la demande avant d'appeler le service et sélectionner aucun rembourrage en omettant l'indicateur de rembourrage « 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-1:111122223333:key/5dza7xqd6soanjtb",
 "KeyCheckValue": "FF9DE9CE"
}
```

Déchiffrer des données

L'`Decrypt Data`API est utilisée pour déchiffrer les données à l'aide de clés de chiffrement symétriques et asymétriques ainsi que de clés dérivées de DUKPT et EMV. Différents algorithmes et variantes sont pris en charge`TDES`, notamment, `RSA` et`AES`.

Les entrées principales sont la clé de déchiffrement utilisée pour déchiffrer les données, les données chiffrées au format HexBinary à déchiffrer et les attributs de déchiffrement tels que le vecteur d'initialisation, le mode sous forme de blocs, etc. Les sorties principales incluent les données déchiffrées sous forme de texte brut au format HexBinary et la valeur de la somme de contrôle pour la clé de déchiffrement. Pour plus de détails sur toutes les options disponibles, veuillez consulter le guide de l'API pour le déchiffrement.

Exemples

- Déchiffrer les données à l'aide d'une clé symétrique AES
- Déchiffrer les données à l'aide de la clé DUKPT
- Déchiffrer les données à l'aide d'une clé symétrique dérivée d'EMV
- Déchiffrer les données à l'aide d'une clé RSA

Déchiffrer les données à l'aide d'une clé symétrique AES

Example

Dans cet exemple, nous allons déchiffrer les données chiffrées à l'aide d'une clé symétrique. Cet exemple montre une `AES` clé, mais elle `TDES_2KEY` est également prise en charge. `TDES_3KEY` Pour cette opération, la clé doit être KeyModesOfUse réglée sur `Decrypt` et KeyUsage définie sur`TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`. Consultez la section Clés pour les opérations cryptographiques pour plus d'options.

```
$ 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"
}
```

Déchiffrer les données à l'aide de la clé DUKPT

> **Note**
>
> L'utilisation de données de déchiffrement avec DUKPT pour les transactions P2PE peut renvoyer à votre application les données PAN et autres données du titulaire de la carte, qui devront être prises en compte lors de la détermination de son champ d'application PCI DSS.

Example

Dans cet exemple, nous allons déchiffrer les données chiffrées à l'aide d'une clé DUKPT créée à l'aide de l'CreateKeyopération ou importée à l'aide de l'opération. ImportKey Pour cette opération, la clé doit être KeyModesOfUse réglée sur `DeriveKey` et KeyUsage définie sur`TR31_B0_BASE_DERIVATION_KEY`. Consultez la section Clés pour les opérations cryptographiques pour plus d'options. Lorsque vous utilisez`DUKPT`, pour l'`TDES`algorithme, la longueur des données du texte chiffré doit être un multiple de 16 octets. Pour `AES` l'algorithme, la longueur des données du texte chiffré doit être un multiple de 32 octets.

```
$ 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"
}
```

Déchiffrer les données à l'aide d'une clé symétrique dérivée d'EMV

Example

Dans cet exemple, nous allons déchiffrer les données de texte chiffré à l'aide d'une clé symétrique dérivée de l'EMV créée à l'aide de l'opération ou importée à l'aide de l'CreateKeyopération. ImportKey Pour cette opération, la clé doit être KeyModesOfUse définie sur `Derive` et KeyUsage définie sur `TR31_E1_EMV_MKEY_CONFIDENTIALITY` ou`TR31_E6_EMV_MKEY_OTHER`. Reportez-vous à la section Clés pour les opérations cryptographiques pour plus de détails.

```
$ 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"
}
```

Déchiffrer les données à l'aide d'une clé RSA

Example

Dans cet exemple, nous allons déchiffrer les données de texte chiffré à l'aide d'une paire de clés RSA créée à l'aide de l'opération. CreateKey Pour cette opération, la clé doit être KeyModesOfUse réglée sur Activé `Decrypt` et KeyUsage définie sur`TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION`. Consultez la section Clés pour les opérations cryptographiques pour plus d'options.

Pour le PKCS #7 ou d'autres systèmes de rembourrage non pris en charge actuellement, veuillez ne sélectionner aucun rembourrage en omettant l'indicateur de rembourrage « Asymmetric= {} » et supprimez le rembourrage après avoir appelé le service.

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

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-
east-1:111122223333:key/5dza7xqd6soanjtb",
    "KeyCheckValue": "FF9DE9CE",
    "PlainText": "31323334313233343132333431323334"
}
```

Générer et vérifier les données de la carte

La génération et la vérification des données de carte intègrent des données dérivées des données de carte, par exemple CVV, CVV2, CVC et DCVV.

Rubriques

Générer des données de carte

L'`Generate Card Data`API est utilisée pour générer des données de carte à l'aide d'algorithmes tels que CVV, CVV2 ou Dynamic CVV2. Pour savoir quelles clés peuvent être utilisées pour cette commande, consultez la section Clés valides pour les opérations cryptographiques.

De nombreuses valeurs cryptographiques telles que CVV, CVV2, iCVV, CAVV V7 utilisent le même algorithme cryptographique mais font varier les valeurs d'entrée. Par exemple, CardVerificationValue1 contient les entrées « numéro de ServiceCode carte » et « date d'expiration ». Bien que CardVerificationValue2 ne possède que deux de ces entrées, cela est dû au fait que pour CVV2/CVC2, la valeur ServiceCode est fixée à 000. De même, pour iCVV, ServiceCode il est fixé à 999. Certains algorithmes peuvent réutiliser les champs existants, tels que CAVV V8, auquel cas vous devrez consulter le manuel de votre fournisseur pour obtenir les valeurs d'entrée correctes.

> **Note**
>
> La date d'expiration doit être saisie dans le même format (tel que MMYY ou YYMM) pour la génération et la validation afin de produire des résultats corrects.

Générer un CVV2

Example

Dans cet exemple, nous allons générer un CVV2 pour un PAN donné avec les entrées de `PAN` et de date d'expiration de la carte. Cela suppose qu'une clé de vérification de carte a été générée.

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

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

Générer iCVV

Example

Dans cet exemple, nous allons générer un iCVV pour un PAN donné avec les entrées suivantes : un code de PAN service 999 et une date d'expiration de la carte. Cela suppose qu'une clé de vérification de carte a été générée.

Pour tous les paramètres disponibles, voir CardVerificationValue1 dans le guide de référence de l'API.

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

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

Vérifier les données de la carte

`Verify Card Data`est utilisé pour vérifier les données créées à l'aide d'algorithmes de paiement basés sur des principes de cryptage tels que`DISCOVER_DYNAMIC_CARD_VERIFICATION_CODE`.

Les valeurs d'entrée sont généralement fournies dans le cadre d'une transaction entrante à un émetteur ou à un partenaire de plateforme de support. Pour vérifier un cryptogramme ARQC (utilisé pour les cartes à puce EMV), veuillez consulter Vérifier l'ARQC.

Pour plus d'informations, consultez VerifyCardValidationDatale guide de l'API.

Si la valeur est vérifiée, l'API renverra http/200. Si la valeur n'est pas vérifiée, elle renverra http/400.

Vérifiez le CVV2

Example

Dans cet exemple, nous allons valider un CVV/CVV2 pour un PAN donné. Le CVV2 est généralement fourni par le titulaire de la carte ou l'utilisateur au moment de la transaction pour validation. Afin de valider leur saisie, les valeurs suivantes seront fournies lors de l'exécution : clé à utiliser pour la validation (CVK)PAN, date d'expiration de la carte et code CVV2 saisi. Le format d'expiration de la carte doit correspondre à celui utilisé lors de la génération de valeur initiale.

Pour tous les paramètres disponibles, voir CardVerificationValue2 dans le guide de référence de l'API.

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

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

Vérifiez iCVV

Example

Dans cet exemple, nous allons vérifier un iCVV pour un PAN donné en saisissant la clé à utiliser pour la validation (CVK), le code de service 999PAN, la date d'expiration de la carte et l'iCVV fourni par la transaction à valider.

iCVV n'est pas une valeur saisie par l'utilisateur (comme CVV2) mais intégrée sur une carte EMV. Il convient de se demander s'il doit toujours être validé lorsqu'il est fourni.

Pour tous les paramètres disponibles, voir CardVerificationValue1 dans le guide de référence de l'API.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi
 --primary-account-number=171234567890123 --verification-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=999} --validation-data 801
```

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

Générez, traduisez et vérifiez les données PIN

Les fonctions de données PIN vous permettent de générer des codes PIN aléatoires, des valeurs de vérification des codes PIN (PVV) et de valider les codes PIN chiffrés entrants par rapport au PVV ou aux décalages de code PIN.

La traduction par épingle vous permet de traduire une épingle d'une touche fonctionnelle à une autre sans exposer l'épingle en texte clair, conformément à l'exigence 1 du code PIN PCI.

> **Note**
>
> Étant donné que la génération et la validation du code PIN sont généralement des fonctions de l'émetteur et que la traduction du code PIN est une fonction typique de l'acquéreur, nous

vous recommandons de prendre en compte l'accès le moins privilégié et de définir des politiques adaptées au cas d'utilisation de vos systèmes.

Rubriques

- Translate code PIN
- Générer des données PIN
- Vérifier les données PIN

Translate code PIN

Les fonctions Translate PIN data sont utilisées pour traduire les données PIN cryptées d'un jeu de clés à un autre sans que les données chiffrées ne quittent le HSM. Il est utilisé pour le chiffrement P2PE où les clés de travail doivent changer mais où le système de traitement n'a pas besoin de déchiffrer les données ou n'est pas autorisé à le faire. Les entrées principales sont les données cryptées, la clé de chiffrement utilisée pour chiffrer les données, les paramètres utilisés pour générer les valeurs d'entrée. L'autre ensemble d'entrées est constitué des paramètres de sortie demandés, tels que la clé à utiliser pour chiffrer la sortie et les paramètres utilisés pour créer cette sortie. Les principaux résultats sont un ensemble de données nouvellement crypté ainsi que les paramètres utilisés pour le générer.

> **Note**
>
> Les types de clés AES ne prennent en charge que les blocs à 4 broches au format ISO.

Rubriques

- Code PIN de PEK à DUKPT
- Code PIN de DUKPT à AWK

Code PIN de PEK à DUKPT

Example

Dans cet exemple, nous allons convertir un code PIN issu du chiffrement PEK TDES utilisant un bloc PIN ISO 0 en un bloc PIN AES ISO 4 à l'aide de l'algorithme DUKPT. Cela peut généralement être

fait dans le sens inverse, lorsqu'un terminal de paiement chiffre un code PIN dans la norme ISO 4, puis il peut être retraduit en TDES pour un traitement en aval.

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

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

Code PIN de DUKPT à AWK

Example

Dans cet exemple, nous allons traduire un code PIN crypté AES DUKPT en code PIN crypté sous un AWK. C'est fonctionnellement l'inverse de l'exemple précédent.

```
$ 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"
        }
```

Générer des données PIN

Les fonctions de génération de données PIN sont utilisées pour générer des valeurs liées au code PIN, telles que le PVV et les décalages par blocs de code utilisés pour valider la saisie du code PIN par les utilisateurs pendant la transaction ou le délai d'autorisation. Cette API peut également générer une nouvelle épingle aléatoire à l'aide de divers algorithmes.

Générer un visa PVV pour une épingle

Example

Dans cet exemple, nous allons générer une nouvelle épingle (aléatoire) dont les sorties seront chiffrées `PIN block` (PinData. PinBlock) et a PVV (PinData.offset). Les entrées clés sontPAN, le`Pin Verification Key`, le `Pin Encryption Key` et le`PIN block format`.

Cette commande nécessite que la clé soit de type`TR31_V2_VISA_PIN_VERIFICATION_KEY`.

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

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

```
            }
        }
```

Générer un décalage de broche IBM3624 pour une broche

IBM 3624 PIN Offset est aussi parfois appelée méthode IBM. Cette méthode génère un code PIN naturel/intermédiaire à l'aide des données de validation (généralement le PAN) et d'une clé PIN (PVK). Les épingles naturelles sont en fait une valeur dérivée et le caractère déterministe est très efficace à gérer pour un émetteur, car aucune donnée PIN n'a besoin d'être stockée au niveau du titulaire de la carte. L'inconvénient le plus évident est que ce système ne prend pas en compte les épingles sélectionnables ou aléatoires par le titulaire de la carte. Pour autoriser ces types de broches, un algorithme de décalage a été ajouté au schéma. Le décalage représente la différence entre l'épingle sélectionnée (ou aléatoire) par l'utilisateur et la touche naturelle. La valeur de décalage est enregistrée par l'émetteur ou le processeur de la carte. Au moment de la transaction, le service AWS de cryptographie des paiements recalcule en interne le code PIN naturel et applique le décalage pour trouver le code PIN. Il compare ensuite cette valeur à la valeur fournie par l'autorisation de transaction.

Plusieurs options existent pour IBM3624 :

- `Ibm3624NaturalPin`affichera le code PIN naturel et un bloc de code PIN crypté
- `Ibm3624PinFromOffset`générera un bloc PIN crypté avec un décalage
- `Ibm3624RandomPin`générera un code PIN aléatoire, puis le décalage correspondant et le bloc de code crypté.
- `Ibm3624PinOffset`génère le décalage de broche en fonction d'une épingle sélectionnée par l'utilisateur.

En interne à la cryptographie des AWS paiements, les étapes suivantes sont effectuées :

- Ajoutez le plan fourni à 16 caractères. Si <16 est indiqué, tapez sur le côté droit en utilisant le caractère de remplissage fourni.
- Chiffre les données de validation à l'aide de la clé de génération du code PIN.
- Décimalisez les données chiffrées à l'aide de la table de décimalisation. Cela fait correspondre des chiffres hexadécimaux à des chiffres décimaux, par exemple « A » peut correspondre à 9 et 1 peut correspondre à 1.
- Obtenez les 4 premiers chiffres à partir d'une représentation hexadécimale de la sortie. C'est l'épingle naturelle.

- Si un code PIN a été sélectionné par l'utilisateur ou généré au hasard, modulo soustrait le code naturel du code PIN du client. Le résultat est le décalage de la broche.

Exemples

- Exemple : générer un décalage de broche IBM3624 pour une épingle

Exemple : générer un décalage de broche IBM3624 pour une épingle

Dans cet exemple, nous allons générer une nouvelle épingle (aléatoire) dont les sorties seront chiffrées `PIN block` (PinData. PinBlock) et une valeur de `IBM3624` décalage (Pindata.offset). Les entrées sont `PAN` les données de validation (généralement le pan), le caractère de remplissage, le`Pin Verification Key`, le `Pin Encryption Key` et le`PIN block format`.

Cette commande nécessite que la clé de génération du code PIN soit de type `TR31_V1_IBM3624_PIN_VERIFICATION_KEY` et que la clé de chiffrement soit de type `TR31_P0_PIN_ENCRYPTION_KEY`

Example

L'exemple suivant montre la génération d'une épingle aléatoire puis la sortie du bloc de broches crypté et de la valeur de décalage IBM3624 à l'aide de Ibm3624. RandomPin

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

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

```
            }
        }
```

Vérifier les données PIN

Les fonctions de vérification des données PIN sont utilisées pour vérifier si un code PIN est correct. Cela implique généralement de comparer le code PIN précédemment enregistré à celui saisi par le titulaire de la carte lors d'un POI. Ces fonctions comparent deux valeurs sans exposer la valeur sous-jacente de l'une ou l'autre des sources.

Valider le code PIN crypté en utilisant la méthode PVV

Example

Dans cet exemple, nous allons valider un code PIN pour un PAN donné. Le code PIN est généralement fourni par le titulaire de la carte ou l'utilisateur au moment de la transaction à des fins de validation et est comparé à la valeur enregistrée (l'entrée du titulaire de la carte est fournie sous forme de valeur cryptée par le terminal ou un autre fournisseur en amont). Afin de valider cette entrée, les valeurs suivantes seront également fournies lors de l'exécution : la clé utilisée pour chiffrer le code PIN d'entrée (il s'agit souvent d'un`IWK`) PAN et la valeur par rapport à laquelle vérifier (un PVV ou`PIN offset`).

Si AWS Payment Cryptography parvient à valider le code PIN, un http/200 est renvoyé. Si le code PIN n'est pas validé, il renverra un http/400.

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

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

```
            "EncryptionKeyCheckValue": "7CC9E2",

        }
```

Valider un code PIN par rapport au décalage de broche IBM3624 précédemment enregistré

Dans cet exemple, nous validerons le code PIN fourni par le titulaire de la carte par rapport à l'offset du code enregistré auprès de l'émetteur/du processeur de la carte. Les entrées sont similaires ??? à l'ajout du code PIN crypté fourni par le terminal de paiement (ou un autre fournisseur en amont tel que le réseau de cartes). Si le code PIN correspond, l'API renverra http 200, où les sorties seront cryptées `PIN block` (PinData. PinBlock) et une valeur de `IBM3624` décalage (Pindata.offset).

Cette commande nécessite que la clé de génération du code PIN soit de type `TR31_V1_IBM3624_PIN_VERIFICATION_KEY` et que la clé de chiffrement soit de type `TR31_P0_PIN_ENCRYPTION_KEY`

Example

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

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

Cryptogramme de demande d'authentification (ARQC)

L'API de cryptogramme de demande d'authentification de vérification est utilisée pour vérifier l'ARQC. La génération de l'ARQC n'entre pas dans le cadre de la cryptographie des AWS paiements et est généralement effectuée sur une carte à puce EMV (ou un équivalent numérique tel qu'un portefeuille mobile) pendant le temps d'autorisation de la transaction. Un ARQC est unique à chaque transaction et est destiné à montrer de manière cryptographique à la fois la validité de la carte et à garantir que les données de transaction correspondent exactement à la transaction en cours (attendue).

AWS La cryptographie des paiements fournit diverses options pour valider l'ARQC et générer des valeurs ARPC facultatives, notamment celles définies dans le livre 2 de l'EMV 4.4 et d'autres schémas utilisés par Visa et Mastercard. Pour une liste complète de toutes les options disponibles, consultez la VerifyCardValidationData section du guide de l'API.

Les cryptogrammes ARQC nécessitent généralement les entrées suivantes (bien que cela puisse varier en fonction de l'implémentation) :

- PAN - Spécifié dans le PrimaryAccountNumber champ
- Numéro de séquence PAN (PSN) - spécifié dans le champ PanSequenceNumber
- Méthode de dérivation des clés, telle que la clé de session commune (CSK), spécifiée dans le SessionKeyDerivationAttributes
- Mode de dérivation de la clé principale (tel que l'option A EMV) - Spécifié dans le MajorKeyDerivationMode
- Données de transaction - une chaîne de diverses données de transaction, de terminal et de carte telles que le montant et la date - spécifiées dans le TransactionData champ
- Clé principale de l'émetteur : clé principale utilisée pour dériver la clé cryptographique (AC) utilisée pour protéger les transactions individuelles et spécifiée dans le champ KeyIdentifier

Rubriques

- Création de données de transaction
- Rembourrage des données de transaction
- Exemples

Création de données de transaction

Le contenu exact (et l'ordre) du champ de données de transaction varie en fonction de l'implémentation et du schéma de réseau, mais les champs minimaux recommandés (et la séquence de concaténation) sont définis dans le livre 2 de l'EMV 4.4, section 8.1.1 - Sélection des données. Si les trois premiers champs sont le montant (17,00), l'autre montant (0,00) et le pays d'achat, les données de transaction commenceront comme suit :

- 000000001700 - montant - 12 positions décimales implicites à deux chiffres
- 000000000000 - autre montant - 12 positions décimales implicites à deux chiffres
- 0124 - code de pays à quatre chiffres
- Données de transaction (partielles) de sortie - 00000000170000000000000000124

Rembourrage des données de transaction

Les données de transaction doivent être complétées avant d'être envoyées au service. La plupart des schémas utilisent le remplissage de la méthode 2 ISO 9797, où une chaîne hexadécimale est ajoutée par 80 puis par 00 jusqu'à ce que le champ soit un multiple de la taille du bloc de chiffrement ; 8 octets ou 16 caractères pour TDES et 16 octets ou 32 caractères pour AES. L'alternative (méthode 1) n'est pas aussi courante mais utilise uniquement 00 comme caractères de remplissage.

Rembourrage ISO 9797 Méthode 1

Non rembourré :
00000000000000000000000840008000800008401605170000000093800000B03011203 (74 caractères ou 37 octets)

Rembourré :
00000000000000000000000840008000800008401605170000000093800000B03011203 000000 (80 caractères ou 40 octets)

Rembourrage ISO 9797 Méthode 2

Non rembourré :
00000000000000000000000840008000800008401605170000000093800000B1F220103000000 (80 caractères ou 40 octets)

Rembourré :
0000000000000000000000084000800080000840160517000000000938000008B1F220103000000
8000000000000000 (88 caractères ou 44 octets)

Exemples

Visa CVN 10

Example

Dans cet exemple, nous allons valider un ARQC généré à l'aide de Visa CVN10.

Si AWS Payment Cryptography est en mesure de valider l'ARQC, un http/200 est renvoyé. Si l'arqc n'est pas validé, il renverra une réponse http/400.

```
$ aws payment-cryptography-data verify-auth-request-cryptogram --auth-request-
cryptogram D791093C8A921769 \
--key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk \
--major-key-derivation-mode EMV_OPTION_A \
--transaction-data
 00000000170000000000000008400080008000084016051700000000093800000B03011203000000 \
--session-key-derivation-attributes='{"Visa":{"PanSequenceNumber":"01" \
,"PrimaryAccountNumber":"9137631040001422"}}'
```

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

Visa CVN18 et Visa CVN22

Example

Dans cet exemple, nous allons valider un ARQC généré à l'aide de Visa CVN18 ou CVN22. Les opérations cryptographiques sont les mêmes entre le CVN18 et le CVN22, mais les données contenues dans les données de transaction varient. Par rapport au CVN10, un cryptogramme complètement différent est généré même avec les mêmes entrées.

Si AWS Payment Cryptography est en mesure de valider l'ARQC, un http/200 est renvoyé. Si l'arqc n'est pas validé, il renverra un http/400.

```
$ aws payment-cryptography-data verify-auth-request-cryptogram \
--auth-request-cryptogram 61EDCC708B4C97B4
--key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk \
--major-key-derivation-mode EMV_OPTION_A
--transaction-data
 00000000170000000000000084000800080000840160517000000000938000 0B1F22010300000000000
 \
00000000000000000000000000000000000000000800000000000000
--session-key-derivation-attributes='{"EmvCommon":
{"ApplicationTransactionCounter":"000B", \
"PanSequenceNumber":"01","PrimaryAccountNumber":"9137631040001422"}}'
```

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

Générer et vérifier MAC

Les codes d'authentification de message (MAC) sont généralement utilisés pour authentifier l'intégrité d'un message (s'il a été modifié). Les hachages cryptographiques tels que le HMAC (code d'authentification de message basé sur le hachage), le CBC-MAC et le CMAC (code d'authentification de message basé sur le chiffrement) fournissent également une assurance supplémentaire à l'expéditeur du MAC en utilisant la cryptographie. HMAC est basé sur des fonctions de hachage tandis que CMAC est basé sur des chiffrements par blocs.

Tous les algorithmes MAC de ce service combinent une fonction de hachage cryptographique et une clé secrète partagée. Ils prennent un message et une clé secrète, tels que le contenu clé d'une clé, et renvoient un tag ou un mac unique. Si un seul caractère du message change, ou si la clé secrète change, le tag obtenu est totalement différent. En exigeant une clé secrète, les MAC cryptographiques garantissent également l'authenticité ; il est impossible de générer un mac identique sans la clé secrète. Les MAC cryptographiques sont parfois appelés signatures symétriques, car ils fonctionnent comme des signatures numériques, mais utilisent une clé unique pour la signature et la vérification.

AWSLa cryptographie des paiements prend en charge plusieurs types de MAC :

ALGORITHME ISO9797 1

Désigné par ou `KeyUsage` ISO9797_ALGORITHM1

ALGORITHME ISO9797 3 (MAC de détail)

Désigné par ou `KeyUsage` ISO9797_ALGORITHM3

ALGORITHME ISO9797 5 (CMAC)

Désigné par ou `KeyUsage` TR31_M6_ISO_9797_5_CMAC_KEY

HMAC

Désigné par `KeyUsage` TR31_M7_HMAC_KEY, y compris HMAC_SHA224, HMAC_SHA256, HMAC_SHA384 et HMAC_SHA512

Rubriques

Générer un MAC

L'API Generate MAC est utilisée pour authentifier les données relatives aux cartes, telles que les données de suivi provenant d'une bande magnétique de carte, en utilisant des valeurs de données connues pour générer un code d'authentification de message (MAC) pour la validation des données entre les parties émettrices et réceptrices. Les données utilisées pour générer le MAC incluent des données de message, une clé de cryptage MAC secrète et un algorithme MAC pour générer une valeur MAC unique pour la transmission. Le destinataire du MAC utilisera les mêmes données de message MAC, la même clé de chiffrement MAC et le même algorithme pour reproduire une autre valeur MAC à des fins de comparaison et d'authentification des données. Même si un caractère du message change ou si la clé MAC utilisée pour la vérification n'est pas identique, la valeur MAC obtenue est différente. L'API prend en charge les clés de chiffrement DUPKT MAC, HMAC et EMV MAC pour cette opération.

La valeur d'entrée pour `message-data` doit être des données HexBinary.

Dans cet exemple, nous allons générer un code HMAC (Hash-Based Message Authentication Code) pour l'authentification des données de carte à l'aide de l'algorithme HMAC `HMAC_SHA256` et de la clé de cryptage HMAC. La clé doit être KeyUsage réglée sur `TR31_M7_HMAC_KEY` et

KeyModesOfUse sur`Generate`. La clé MAC peut être créée avec AWS Payment Cryptography en appelant CreateKeyou importée en appelant 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"
}
```

Vérifiez le MAC

L'API Verify MAC est utilisée pour vérifier le MAC (code d'authentification des messages) pour l'authentification des données liées à la carte. Il doit utiliser la même clé de chiffrement utilisée lors de la génération du MAC pour reproduire la valeur MAC à des fins d'authentification. La clé de chiffrement MAC peut être créée avec AWS Payment Cryptography en appelant CreateKeyou importée par appel ImportKey. L'API prend en charge les clés de chiffrement DUPKT MAC, HMAC et EMV MAC pour cette opération.

Si la valeur est vérifiée, le paramètre de réponse `MacDataVerificationSuccessful` sera renvoyé`Http/200`, sinon `Http/400` avec un message l'indiquant`Mac verification failed`.

Dans cet exemple, nous allons vérifier un code HMAC (Hash-Based Message Authentication Code) pour l'authentification des données de carte à l'aide de l'algorithme HMAC `HMAC_SHA256` et de la clé de cryptage HMAC. La clé doit être KeyUsage réglée sur `TR31_M7_HMAC_KEY` et KeyModesOfUse sur`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",
}
```

Clés valides pour les opérations cryptographiques

Certaines touches ne peuvent être utilisées que pour certaines opérations. En outre, certaines opérations peuvent limiter les modes d'utilisation des touches. Consultez le tableau suivant pour connaître les combinaisons autorisées.

> **Note**
>
> Certaines combinaisons, bien qu'autorisées, peuvent créer des situations inutilisables, telles que la génération de codes CVV (`generate`) mais l'impossibilité de les vérifier. (`verify`)

Rubriques

- GenerateCardDonnées
- VerifyCardDonnées
- GeneratePinData (pour les programmes VISA/ABA)
- GeneratePinData (pourIBM3624)
- VerifyPinData (pour les programmes VISA/ABA)
- VerifyPinData (pourIBM3624)
- Déchiffrer des données
- Encrypt Data
- Translate Pin Data

- Générer/vérifier un MAC
- VerifyAuthRequestCryptogram
- Clé d'Import/Export
- Types de clés non utilisés

GenerateCardDonnées

Point de terminaison API	Opération ou algorithme cryptographique	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
GenerateCardDonnées	• AMEX_CARD_SECURITY_CODE_VERSION_1 • AMEX_CARD_SECURITY_CODE_VERSION_2	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KEY • TDES_3KEY	{Générer = vrai}, {Générer = vrai, Vérifier = vrai}
GenerateCardDonnées	• VALEUR_VÉRIFICATION_1 DE LA CARTE_1 • VALEUR_VÉRIFICATION_2 DE LA CARTE_2	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KEY	{Générer = vrai}, {Générer = vrai, Vérifier = vrai}
GenerateCardDonnées	• VALEUR_AUTHENTIFICATION_VÉRIFICATION_DU_TITUL	TR31_E6_EMV_MKEY_AUTRE	• TDES_2KEY	{ DeriveKey = vrai}

Point de terminaison API	Opération ou algorithme cryptographique	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
	AIRE DE LA CARTE			
GenerateCardDonnées	• CODE DE VÉRIFICATION DE LA CARTE DYNAMIQUE	TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS	• TDES_2KEY	{ DeriveKey = vrai}
GenerateCardDonnées	• VALEUR_VÉRIFICATION_DYNAMIQUE DE LA CARTE	TR31_E6_EMV_MKEY_AUTRE	• TDES_2KEY	{ DeriveKey = vrai}

VerifyCardDonnées

Opération ou algorithme cryptographique	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
• AMEX_CARD_SECURITY_CODE_VERSION_1 • AMEX_CARD_SECURITY_CODE_VERSION_2	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KEY • TDES_3KEY	{Générer = vrai}, {Générer = vrai, Vérifier = vrai}

Opération ou algorithme cryptographique	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
• VALEUR_VÉRIFICATION_1 DE LA CARTE_1 • VALEUR_VÉRIFICATION_2 DE LA CARTE_2	TR31_C0_CARD_VERIFICATION_KEY	• TDES_2KEY	{Générer = vrai}, {Générer = vrai, Vérifier = vrai}
• VALEUR_AUTHENTIFICATION_VÉRIFICATION_DU_TITULAIRE DE LA CARTE	TR31_E6_EMV_MKEY_AUTRE	• TDES_2KEY	{ DeriveKey = vrai}
• CODE DE VÉRIFICATION DE LA CARTE DYNAMIQUE	TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS	• TDES_2KEY	{ DeriveKey = vrai}
• VALEUR_VÉRIFICATION_DYNAMIQUE DE LA CARTE	TR31_E6_EMV_MKEY_AUTRE	• TDES_2KEY	{ DeriveKey = vrai}

GeneratePinData (pour les programmes VISA/ABA)

`VISA_PIN or VISA_PIN_VERIFICATION_VALUE`

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
Clé de cryptage PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY	• {Chiffrer = vrai, envelopper = vrai} • {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai} • { NoRestrictions = vrai}
Clé de génération de code PIN	TR31_V2_VISA_PIN_VERIFICATION_KEY	• TDES_3KEY	• {Générer = vrai} • {Générer = vrai, vérifier = vrai}

GeneratePinData (pour**IBM3624**)

`IBM3624_PIN_OFFSET,IBM3624_NATURAL_PIN,IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET)`

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
Clé de cryptage PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY	Pour IBM3624_NATURAL_PIN, IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
			• {Chiffrer = vrai, envelopper = vrai} • {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai} • { NoRestrictions = vrai} Pour IBM3624_PIN_OFFSET • {Chiffrer = vrai, Décompresser = vrai} • {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai} • { NoRestrictions = vrai}
Clé de génération de code PIN	TR31_V1_IBM3624_PIN_VERIFICATION_KEY	• TDES_3KEY	• {Générer = vrai} • {Générer = vrai, vérifier = vrai}

VerifyPinData (pour les programmes VISA/ABA)

`VISA_PIN`

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
Clé de cryptage PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY	• {Déchiffrer = vrai, Décompresser = vrai} • {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai} • { NoRestrictions = vrai}
Clé de génération de code PIN	TR31_V2_VISA_PIN_VERIFICATION_KEY	• TDES_3KEY	• {Vérifier = vrai} • {Générer = vrai, vérifier = vrai}

VerifyPinData (pour**IBM3624**)

`IBM3624_PIN_OFFSET,IBM3624_NATURAL_PIN,IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET)`

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
Clé de cryptage PIN	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY	Pour IBM3624_NATURAL_PIN, IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
			• {Déchiffrer = vrai, Décompresser = vrai} • {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai} • { NoRestrictions = vrai}
Clé de vérification du code PIN	TR31_V1_IBM3624_PIN_VERIFICATION_KEY	• TDES_3KEY	• {Vérifier = vrai} • {Générer = vrai, vérifier = vrai}

Déchiffrer des données

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = vrai} • { NoRestrictions = vrai}
EMV	TR31_E1_EMV_MKEY_CONFIDENTIALITY	• TDES_2KEY	• { DeriveKey = vrai}

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
	TR31_E6_E MV_MKEY_AUTRE		
RSA	TR31_D1_C LÉ_ASYMÉT RIQUE POUR LE CHIFFREMENT DES DONNÉES	• RSA_2048 • RSA_3072 • RSA_4096	• {Déchiffrer = vrai, unwrap=vrai} • {Encrypt=True, Wrap=True, Decrypt = true, Unwrap=True}
Clés symétriques	TR31_D0_S YMMETRIC_ DATA_ENCR YPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• {Déchiffrer = vrai, unwrap=vrai} • {Encrypt=True, Wrap=True, Decrypt = true, Unwrap=True} • { NoRestrictions = vrai}

Encrypt Data

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
DUKPT	TR31_B0_B ASE_DERIV ATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = vrai} • { NoRestrictions = vrai}

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
EMV	TR31_E1_EMV_MKEY_CONFIDENTIALITY TR31_E6_EMV_MKEY_AUTRE	• TDES_2KEY	• { DeriveKey = vrai}
RSA	TR31_D1_CLÉ_ASYMÉTRIQUE POUR LE CHIFFREMENT DES DONNÉES	• RSA_2048 • RSA_3072 • RSA_4096	• {Chiffrer = vrai, wrap=vrai} • {Encrypt=True, Wrap=True, Decrypt = true, Unwrap=True}
Clés symétriques	TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• {Chiffrer = vrai, wrap=vrai} • {Encrypt=True, Wrap=True, Decrypt = true, Unwrap=True} • { NoRestrictions = vrai}

Translate Pin Data

Direction	Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
Source de données entrante	DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = vrai} • { NoRestrictions = vrai}
Source de données entrante	Non dopé (PEK, AWK, IWK, etc.)	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• {Déchiffrer = vrai, Décompresser = vrai} • {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai} • { NoRestrictions = vrai}
Cible de données sortantes	DUKPT	TR31_B0_BASE_DERIVATION_KEY	• TDES_2KEY • AES_128 • AES_192 • AES_256	• { DeriveKey = vrai} • { NoRestrictions = vrai}
Cible de données sortantes	Non dopé (PEK, IWK, AWK, etc.)	TR31_P0_PIN_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192	• {Chiffrer = vrai, envelopper = vrai}

Direction	Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
			• AES_256	• {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai} • { NoRestrictions = vrai}

Générer/vérifier un MAC

Les clés MAC sont utilisées pour créer des hachages cryptographiques d'un message/corps de données. Il n'est pas recommandé de créer une clé avec des modes d'utilisation limités car vous ne pourrez pas effectuer l'opération correspondante. Toutefois, vous pouvez importer/exporter une clé en une seule opération si l'autre système est censé effectuer l'autre moitié de la paire d'opérations.

Utilisation autorisée des clés	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
Clé MAC	TR31_M1_ISO_9797_1_MAC_KEY	• TDES_2KEY • TDES_3KEY	• {Générer = vrai} • {Générer = vrai, vérifier = vrai} • {Vérifier = vrai} • {Générer = vrai}
Clé MAC (MAC pour le commerce de détail)	TR31_M1_ISO_9797_3_MAC_KEY	• TDES_2KEY • TDES_3KEY	• {Générer = vrai}

Utilisation autorisée des clés	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
			• {Générer = vrai, vérifier = vrai} • {Vérifier = vrai} • {Générer = vrai}
Clé MAC (CMAC)	TR31_M6_ISO_9797_5_CMAC_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• {Générer = vrai} • {Générer = vrai, vérifier = vrai} • {Vérifier = vrai} • {Générer = vrai}
Clé MAC (HMAC)	TR31_M7_HMAC_KEY	• TDES_2KEY • TDES_3KEY • AES_128 • AES_192 • AES_256	• {Générer = vrai} • {Générer = vrai, vérifier = vrai} • {Vérifier = vrai} • {Générer = vrai}

VerifyAuthRequestCryptogram

Utilisation autorisée des clés	Option EMV	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
• OPTION A • OPTION B	TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS	• TDES_2KEY	• { DeriveKey = vrai}

Clé d'Import/Export

Type d'opération	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
Clé d'emballage TR-31	TR31_K1_KEY_BLOCK_PROTECTION_KEY TR31_K0_KEY_ENCRYPTION_KEY	• TDES_2KEY • TDES_3KEY • AES_128	• {Encrypt = true, Wrap = true} (exportation uniquement) • {Decrypt = true, Unwrap = true} (importation uniquement) • {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai}
Importation d'une autorité de certification approuvée	TR31_S0_CLÉ_ASYMÉTRIQUE_POUR_SIGNATURE NUMÉRIQUE	• RSA_2048 • RSA_3072 • RSA_4096	• {Vérifier = vrai}
Importation d'un certificat à clé publique pour le chiffrement asymétrique	TR31_D1_CLÉ_ASYMÉTRIQUE POUR LE CHIFFREMENT DES DONNÉES	• RSA_2048 • RSA_3072 • RSA_4096	• {Encrypt=TRUE, WRAP=TRUE}

Types de clés non utilisés

Les types de clés suivants ne sont pas actuellement utilisés par AWS Payment Cryptography :

- TR31_P1_PIN_GENERATION_KEY
- TR31_K3_KEY_ASYMETRIC_FOR_KEY_AGREEMENT

Cas d'utilisation courants

AWS La cryptographie des paiements prend en charge de nombreuses opérations cryptographiques de paiement classiques. Les rubriques suivantes servent de guide sur l'utilisation de ces opérations pour les cas d'utilisation courants typiques. Pour obtenir la liste de toutes les commandes, veuillez consulter l'API AWS Payment Cryptography.

Rubriques

Émetteurs et processeurs d'émetteurs

Les cas d'utilisation par les émetteurs se composent généralement de quelques parties. Cette section est organisée par fonction (utilisation d'épingles, par exemple). Dans un système de production, les clés sont généralement limitées à un compartiment de cartes donné et sont créées lors de la configuration du compartiment plutôt qu'en ligne, comme indiqué ici.

Rubriques

Fonctions générales

Rubriques

Générez une épingle aléatoire et le code associéPVV, puis vérifiez la valeur

Rubriques

Créez la ou les clés

Pour générer un code PIN aléatoire et le PVV, vous aurez besoin de deux clés, une clé de vérification du code PIN (PVK) pour générer le code PIN PVV et une clé de cryptage du code PIN pour chiffrer le code PIN. Le code PIN lui-même est généré de manière aléatoire en toute sécurité dans le service et n'est lié cryptographiquement à aucune des clés.

PGKIl doit s'agir d'une clé de l'algorithme TDES _2 KEY basée sur l'PVValgorithme lui-même. A PEK peut être TDES _2KEY, TDES _3 KEY ou AES _128. Dans ce cas, étant donné que le PEK est destiné à un usage interne au sein de votre système, AES _128 serait un bon choix. Si a PEK est utilisé pour échanger avec d'autres systèmes (par exemple, des réseaux de cartes, des acquéreursATMs) ou s'il est déplacé dans le cadre d'une migration, TDES _2 KEY peut être le choix le plus approprié pour des raisons de compatibilité.

Créez le PEK

```
$ aws payment-cryptography create-key \
               --exportable
               --key-attributes
 KeyAlgorithm=AES_128,KeyUsage=TR31_P0_PIN_ENCRYPTION_KEY,\
                   KeyClass=SYMMETRIC_KEY,\
                   KeyModesOfUse='{Encrypt=true,Decrypt=true,Wrap=true,Unwrap=true}' --
tags='[{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
               "Key": {
                   "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt",
                   "KeyAttributes": {
```

```
                    "KeyUsage": "TR31_P0_PIN_ENCRYPTION_KEY",
                    "KeyClass": "SYMMETRIC_KEY",
                    "KeyAlgorithm": "AES_128",
                    "KeyModesOfUse": {
                        "Encrypt": false,
                        "Decrypt": false,
                        "Wrap": false,
                        "Unwrap": false,
                        "Generate": true,
                        "Sign": false,
                        "Verify": true,
                        "DeriveKey": false,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "7CC9E2",
                "KeyCheckValueAlgorithm": "CMAC",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2023-06-05T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2023-06-05T06:41:46.626000-07:00"
            }
        }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/ivi5ksfsuplneuyt. Vous en aurez besoin à l'étape suivante.

Créez le PVK

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_V2_VISA_PIN_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyMo
  --tags='[{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
                "Key": {
                    "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/ov6icy4ryas4zcza",
```

```
                "KeyAttributes": {
                    "KeyUsage": "TR31_V2_VISA_PIN_VERIFICATION_KEY",
                    "KeyClass": "SYMMETRIC_KEY",
                    "KeyAlgorithm": "TDES_2KEY",
                    "KeyModesOfUse": {
                        "Encrypt": false,
                        "Decrypt": false,
                        "Wrap": false,
                        "Unwrap": false,
                        "Generate": true,
                        "Sign": false,
                        "Verify": true,
                        "DeriveKey": false,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "51A200",
                "KeyCheckValueAlgorithm": "ANSI_X9_24",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2023-06-05T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2023-06-05T06:41:46.626000-07:00"
            }
        }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/ov6icy4ryas4zcza. Vous en aurez besoin à l'étape suivante.

Générez un code PIN aléatoire, générez PVV et renvoyez le code crypté PIN et PVV

Example

Dans cet exemple, nous allons générer un nouveau code PIN (aléatoire) à 4 chiffres dont les sorties seront chiffrées `PIN block` (PinData. PinBlock) et un PVV (pinData. VerificationValue). Les entrées principales sont PAN le format `Pin Verification Key` (également connu sous le nom de clé de génération de code PIN) `Pin Encryption Key` et le format PINBlock.

Cette commande nécessite que la clé soit de type`TR31_V2_VISA_PIN_VERIFICATION_KEY`.

```
$ aws payment-cryptography-data generate-pin-data --generation-key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2 --encryption-
```

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

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

Valider le chiffrement PIN en utilisant PVV la méthode

Example

Dans cet exemple, nous allons valider un PIN pour une donnéePAN. PINIl est généralement fourni par le titulaire de la carte ou l'utilisateur au moment de la transaction à des fins de validation et est comparé à la valeur enregistrée (l'entrée du titulaire de la carte est fournie sous forme de valeur cryptée par le terminal ou un autre fournisseur en amont). Afin de valider cette entrée, les valeurs suivantes seront également fournies lors de l'exécution : le code PIN crypté, la clé utilisée pour chiffrer le code PIN d'entrée (souvent appelé code PIN IWK) PAN et la valeur par rapport à laquelle vérifier (a PVV ou`PIN offset`).

Si AWS Payment Cryptography parvient à valider le code PIN, un http/200 est renvoyé. Si le code PIN n'est pas validé, il renverra un http/400.

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

```
{
```

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

Générer ou vérifier un CVV pour une carte donnée

CVVou CVV1 est une valeur traditionnellement incorporée dans la bande magnétique d'une carte. Ce n'est pas la même chose que CVV2 (visible pour le titulaire de la carte et pour les achats en ligne).

La première étape consiste à créer une clé. Dans le cadre de ce didacticiel, vous allez créer une clé 3 DES (2 KEYTDES) de CVKlongueur double.

> **Note**
>
> CVV, CVV2 et j'utilise CVV tous des algorithmes similaires, sinon identiques, mais je fais varier les données d'entrée. Ils utilisent tous le même type de clé TR31 _C0_ CARD _ VERIFICATION _KEY, mais il est recommandé d'utiliser des clés distinctes pour chaque utilisation. Ils peuvent être distingués à l'aide d'alias et/ou de balises, comme dans l'exemple ci-dessous.

Créez la clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVV"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
            "Key": {
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
r52o3wbqxyf6qlqr",
                "KeyAttributes": {
                    "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY",
```

```
                "KeyClass": "SYMMETRIC_KEY",
                "KeyAlgorithm": "TDES_2KEY",
                "KeyModesOfUse": {
                    "Encrypt": false,
                    "Decrypt": false,
                    "Wrap": false,
                    "Unwrap": false,
                    "Generate": true,
                    "Sign": false,
                    "Verify": true,
                    "DeriveKey": false,
                    "NoRestrictions": false
                }
            },
            "KeyCheckValue": "DE89F9",
            "KeyCheckValueAlgorithm": "ANSI_X9_24",
            "Enabled": true,
            "Exportable": true,
            "KeyState": "CREATE_COMPLETE",
            "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
            "CreateTimestamp": "2023-06-05T06:41:46.648000-07:00",
            "UsageStartTimestamp": "2023-06-05T06:41:46.626000-07:00"
        }
    }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/r52o3wbqxyf6qlqr. Vous en aurez besoin à l'étape suivante.

Générez un CVV

Example

Dans cet exemple, nous allons générer un CVVpour une donnée PAN avec les entrées dePAN, un code de service (tel que défini parISO/IEC7813) de 121 et la date d'expiration de la carte.

Pour tous les paramètres disponibles, voir CardVerificationValue1 dans le guide API de référence.

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
r52o3wbqxyf6qlqr --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=121}'
```

```
                {
                    "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/r52o3wbqxyf6qlqr",
                    "KeyCheckValue": "DE89F9",
                    "ValidationData": "801"
                }
```

Valider CVV

Example

Dans cet exemple, nous allons vérifier un CVVpour un compte PAN en saisissant unCVK, un code de service de 121PAN, la date d'expiration de la carte et le code CVV fourni lors de la transaction pour valider.

Pour tous les paramètres disponibles, voir CardVerificationValue1 dans le guide API de référence.

> **Note**
>
> CVVn'est pas une valeur saisie par l'utilisateur (commeCVV2) mais elle est généralement intégrée à une bande magnétique. Il convient de déterminer s'il doit toujours être validé lorsqu'il est fourni.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/r52o3wbqxyf6qlqr
 --primary-account-number=171234567890123 --verification-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=121} --validation-data 801
```

```
{
                    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
r52o3wbqxyf6qlqr",
                    "KeyCheckValue": "DE89F9",
                    "ValidationData": "801"
}
```

Générer ou vérifier une CVV2 pour une carte spécifique

CVV2est une valeur qui est traditionnellement indiquée au dos d'une carte et qui est utilisée pour les achats en ligne. Pour les cartes virtuelles, elles peuvent également être affichées sur une application ou un écran. Sur le plan cryptographique, c'est le même queCVV1, mais avec une valeur de code de service différente.

Créez la clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVV2"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
               "Key": {
                   "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu",
                   "KeyAttributes": {
                       "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY",
                       "KeyClass": "SYMMETRIC_KEY",
                       "KeyAlgorithm": "TDES_2KEY",
                       "KeyModesOfUse": {
                           "Encrypt": false,
                           "Decrypt": false,
                           "Wrap": false,
                           "Unwrap": false,
                           "Generate": true,
                           "Sign": false,
                           "Verify": true,
                           "DeriveKey": false,
                           "NoRestrictions": false
                       }
                   },
                   "KeyCheckValue": "AEA5CD",
                   "KeyCheckValueAlgorithm": "ANSI_X9_24",
                   "Enabled": true,
                   "Exportable": true,
                   "KeyState": "CREATE_COMPLETE",
                   "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
```

```
                "CreateTimestamp": "2023-06-05T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2023-06-05T06:41:46.626000-07:00"
            }
        }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/7f7g4spf3xcklhzu. Vous en aurez besoin à l'étape suivante.

Générez un CVV2

Example

Dans cet exemple, nous allons générer un CVV2pour une donnée en saisissant la date d'expiration PAN de la carte PAN et la date d'expiration de la carte.

Pour tous les paramètres disponibles, voir CardVerificationValue2 dans le guide API de référence.

```
$ aws payment-cryptography-data generate-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/7f7g4spf3xcklhzu
 --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue2='{CardExpiryDate=1127}'
```

```
                    {
                        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu",
                        "KeyCheckValue": "AEA5CD",
                        "ValidationData": "321"
                    }
```

Valider un CVV2

Example

Dans cet exemple, nous allons vérifier un CVV2pour une donnée PAN en saisissant une CVK date d'expiration de la carte PAN et celle CVV fournie lors de la transaction pour valider.

Pour tous les paramètres disponibles, voir CardVerificationValue2 dans le guide API de référence.

> **Note**
>
> CVV2et les autres entrées sont des valeurs saisies par l'utilisateur. Ce n'est donc pas nécessairement le signe d'un problème que cela échoue périodiquement à valider.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/7f7g4spf3xcklhzu
 --primary-account-number=171234567890123 --verification-attributes
 CardVerificationValue2='{CardExpiryDate=1127} --validation-data 321
```

```
{
                      "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/7f7g4spf3xcklhzu",
                      "KeyCheckValue": "AEA5CD",
                      "ValidationData": "801"
   }
```

Générer ou vérifier un i CVV pour une carte spécifique

J'CVVutilise le même algorithme queCVV/CVV2mais il CVV est intégré dans une carte à puce. Son code de service est 999.

Créez la clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"ICVV"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
               "Key": {
                   "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
c7dsi763r6s7lfp3",
                   "KeyAttributes": {
```

```
                    "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY",
                    "KeyClass": "SYMMETRIC_KEY",
                    "KeyAlgorithm": "TDES_2KEY",
                    "KeyModesOfUse": {
                        "Encrypt": false,
                        "Decrypt": false,
                        "Wrap": false,
                        "Unwrap": false,
                        "Generate": true,
                        "Sign": false,
                        "Verify": true,
                        "DeriveKey": false,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "1201FB",
                "KeyCheckValueAlgorithm": "ANSI_X9_24",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2023-06-05T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2023-06-05T06:41:46.626000-07:00"
            }
        }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/c7dsi763r6s7lfp3. Vous en aurez besoin à l'étape suivante.

Générer un i CVV

Example

Dans cet exemple, nous allons générer un i CVV pour une donnée PAN avec les entrées dePAN, un code de service (tel que défini parISO/IEC7813) de 999 et la date d'expiration de la carte.

Pour tous les paramètres disponibles, voir CardVerificationValue1 dans le guide API de référence.

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
c7dsi763r6s7lfp3 --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=999}'
```

```
                    {
                        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/c7dsi763r6s7lfp3",
                        "KeyCheckValue": "1201FB",
                        "ValidationData": "532"
                    }
```

Validez i CVV

Example

Pour la validation, les entrées sont CVK un code de service 999, la date d'expiration de la carte et le i CVV fourni lors de la transaction pour valider. PAN

Pour tous les paramètres disponibles, voir CardVerificationValue1 dans le guide API de référence.

> **Note**
>
> i n'CVVest pas une valeur saisie par l'utilisateur (commeCVV2) mais elle est généralement intégrée à une carte EMV /chip. Il convient de déterminer s'il doit toujours être validé lorsqu'il est fourni.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/c7dsi763r6s7lfp3
 --primary-account-number=171234567890123 --verification-attributes
 CardVerificationValue1='{CardExpiryDate=1127,ServiceCode=999} --validation-data 532
```

```
{
                    "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/c7dsi763r6s7lfp3",
                    "KeyCheckValue": "1201FB",
                    "ValidationData": "532"
   }
```

Vérifiez EMV ARQC et générez un ARPC

ARQC(Cryptogramme de demande d'autorisation) est un cryptogramme généré par une carte EMV (à puce) et utilisé pour valider les détails de la transaction ainsi que l'utilisation d'une carte autorisée. Il intègre les données de la carte, du terminal et de la transaction elle-même.

Au moment de la validation sur le backend, les mêmes entrées sont fournies à AWS Payment Cryptography, le cryptogramme est recréé en interne et celui-ci est comparé à la valeur fournie avec la transaction. En ce sens, il est similaire à unMAC. EMV4.4 Le livre 2 définit trois aspects de cette fonction : les méthodes de dérivation de clés (connues sous le nom de clé de session commune - CSK) pour générer des clés de transaction uniques, une charge utile minimale et les méthodes de génération d'une réponse ()ARPC.

Les schémas de cartes individuels peuvent spécifier des champs transactionnels supplémentaires à intégrer ou l'ordre dans lequel ces champs apparaissent. D'autres schémas de dérivation spécifiques au schéma (généralement obsolètes) existent également et sont abordés ailleurs dans cette documentation.

Pour plus d'informations, consultez VerifyCardValidationDatale API guide.

Créez la clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS,KeyClass=SYMMETRIC_KEY,KeyMod
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVN18"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
                "Key": {
                    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk",
                    "KeyAttributes": {
                        "KeyUsage": "TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS",
                        "KeyClass": "SYMMETRIC_KEY",
                        "KeyAlgorithm": "TDES_2KEY",
                        "KeyModesOfUse": {
                            "Encrypt": false,
                            "Decrypt": false,
                            "Wrap": false,
```

```
                        "Unwrap": false,
                        "Generate": false,
                        "Sign": false,
                        "Verify": false,
                        "DeriveKey": true,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "08D7B4",
                "KeyCheckValueAlgorithm": "ANSI_X9_24",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
            }
        }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/pw3s6nl62t5ushfk. Vous en aurez besoin à l'étape suivante.

Générez un ARQC

ARQCIl est généré exclusivement par une EMV carte. En tant que telle, la cryptographie des AWS paiements ne permet pas de générer une telle charge utile. À des fins de test, un certain nombre de bibliothèques sont disponibles en ligne et peuvent générer une charge utile appropriée ainsi que des valeurs connues généralement fournies par les différents schémas.

Validez un ARQC

Example

Si AWS Payment Cryptography est en mesure de le validerARQC, un http/200 est renvoyé. Une ARPC (réponse) peut éventuellement être fournie et incluse dans la réponse une fois ARQC celle-ci validée.

```
$ aws payment-cryptography-data verify-auth-request-cryptogram
 --auth-request-cryptogram 61EDCC708B4C97B4 --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk
 --major-key-derivation-mode EMV_OPTION_A --transaction-data
 00000000170000000000000008400080008000084016051700000000093800000B1F220103000000000000000000000
 --session-key-derivation-attributes='{"EmvCommon":
```

```
{"ApplicationTransactionCounter":"000B",
 "PanSequenceNumber":"01","PrimaryAccountNumber":"9137631040001422"}}' --auth-response-
attributes='{"ArpcMethod2":{"CardStatusUpdate":"12345678"}}'
```

```
{
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk",
    "KeyCheckValue": "08D7B4",
    "AuthResponseValue":"2263AC85"
}
```

Générer et vérifier un EMV MAC

EMVMACMACutilise l'entrée d'une clé EMV dérivée, puis exécute un ISO9797 -3 (Retail) MAC sur les données résultantes. EMVMACest généralement utilisé pour envoyer des commandes à une EMV carte, telles que des scripts de déblocage.

> **Note**
>
> AWS La cryptographie des paiements ne valide pas le contenu du script. Veuillez consulter le manuel de votre schéma ou de votre carte pour plus de détails sur les commandes spécifiques à inclure.

Pour plus d'informations, consultez MacAlgorithmEmvle API guide.

Rubriques

- Créez la clé
- Générez un EMV MAC

Créez la clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E2_EMV_MKEY_INTEGRITY,KeyClass=SYMMETRIC_KEY,KeyModesOfUs
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVN18"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
        "Key": {
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk",
            "KeyAttributes": {
                "KeyUsage": "TR31_E2_EMV_MKEY_INTEGRITY",
                "KeyClass": "SYMMETRIC_KEY",
                "KeyAlgorithm": "TDES_2KEY",
                "KeyModesOfUse": {
                    "Encrypt": false,
                    "Decrypt": false,
                    "Wrap": false,
                    "Unwrap": false,
                    "Generate": false,
                    "Sign": false,
                    "Verify": false,
                    "DeriveKey": true,
                    "NoRestrictions": false
                }
            },
            "KeyCheckValue": "08D7B4",
            "KeyCheckValueAlgorithm": "ANSI_X9_24",
            "Enabled": true,
            "Exportable": true,
            "KeyState": "CREATE_COMPLETE",
            "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
            "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
            "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
        }
    }
```


Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/pw3s6nl62t5ushfk. Vous en aurez besoin à l'étape suivante.

Générez un EMV MAC

Généralement, un processus principal génère un EMV script (tel que le déblocage d'une carte), le signe à l'aide de cette commande (qui déduit une clé à usage unique spécifique à une carte en particulier), puis renvoie le. MAC Ensuite, les MAC commandes+ sont envoyées à la carte à appliquer. L'envoi de la commande à la carte ne relève pas du champ d'application de la cryptographie des AWS paiements.

> **Note**
>
> Cette commande est destinée aux commandes lorsqu'aucune donnée cryptée (telle quePIN) n'est envoyée. EMVEncrypt peut être combiné à cette commande pour ajouter des données chiffrées au script de l'émetteur avant d'appeler cette commande

Données du message

Les données du message incluent l'APDUen-tête et la commande. Bien que cela puisse varier selon l'implémentation, cet exemple est l'APDUen-tête du déblocage (84 24 00 00 08), suivi de ATC (0007) puis ARQC de la transaction précédente (CACE999E57FD0F47). Le service ne valide pas le contenu de ce champ.

Mode de dérivation des clés de session

Ce champ définit le mode de génération de la clé de session. EMV_ COMMON _ SESSION _ KEY est généralement utilisé pour les nouvelles implémentations, tandis que EMV2 000 | AMEX | MASTERCARD _ SESSION _ KEY | VISA peut également être utilisé.

MajorKeyDerivationMode

EMVDéfinit le mode A, B ou C. Le mode A est le plus courant et la cryptographie des AWS paiements prend actuellement en charge le mode A ou le mode B.

PAN

Le numéro de compte, généralement disponible dans le champ de puce 5A ou ISO8583 le champ 2, mais peut également être extrait du système de carte.

PSN

Le numéro de séquence de la carte. S'il n'est pas utilisé, entrez 00.

SessionKeyDerivationValue

Il s'agit des données de dérivation par session. Il peut s'agir du dernier ARQC (ApplicationCryptogram) du champ 9F26 ou du dernier du champ 9F36 selon le schéma ATC de dérivation.

Remplissage

Le rembourrage est automatiquement appliqué et utilise la méthode de rembourrageISO/IEC9797-1 2.

Example

```
$ aws payment-cryptography-data generate-mac --message-data
 842400000800079 99E57FD0F47CACE --key-identifier arn:aws:payment-
cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk --message-
data 8424000008999E57FD0F47CACE0007 --generation-attributes
 EmvMac="{MajorKeyDerivationMode=EMV_OPTION_A,PanSequenceNumber='00',PrimaryAccountNumber='2235
```

```
{
"KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk",
"KeyCheckValue": "08D7B4",
"Mac":"5652EEDF83EA0D84"
}
```

Fonctions spécifiques au réseau

Rubriques

- Fonctions spécifiques à Visa
- Fonctions spécifiques à Mastercard
- Fonctions spécifiques à American Express

Fonctions spécifiques à Visa

Rubriques

- ARQC - CVN18/CVN22
- ARQC- CVN1 0
- CAVVV7

ARQC - CVN18/CVN22

CVN18et CVN22 utilisez la CSKméthode de dérivation des clés. Les données de transaction exactes varient entre ces deux méthodes. Consultez la documentation du schéma pour plus de détails sur la création du champ de données de transaction.

ARQC- CVN1 0

CVN10 est une ancienne méthode Visa pour les EMV transactions qui utilise la dérivation de clé par carte plutôt que la dérivation de session (par transaction) et qui utilise également une charge utile

différente. Pour plus d'informations sur le contenu de la charge utile, veuillez contacter le programme pour plus de détails.

Créer une clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS,KeyClass=SYMMETRIC_KEY,KeyMod
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVN10"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
                        "Key": {
                            "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/pw3s6nl62t5ushfk",
                            "KeyAttributes": {
                                "KeyUsage": "TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS",
                                "KeyClass": "SYMMETRIC_KEY",
                                "KeyAlgorithm": "TDES_2KEY",
                                "KeyModesOfUse": {
                                    "Encrypt": false,
                                    "Decrypt": false,
                                    "Wrap": false,
                                    "Unwrap": false,
                                    "Generate": false,
                                    "Sign": false,
                                    "Verify": false,
                                    "DeriveKey": true,
                                    "NoRestrictions": false
                                }
                            },
                            "KeyCheckValue": "08D7B4",
                            "KeyCheckValueAlgorithm": "ANSI_X9_24",
                            "Enabled": true,
                            "Exportable": true,
                            "KeyState": "CREATE_COMPLETE",
                            "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                            "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
                            "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
                        }
                    }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/pw3s6nl62t5ushfk. Vous en aurez besoin à l'étape suivante.

Validez le ARQC

Example

Dans cet exemple, nous allons valider un ARQC généré à l'aide de Visa CVN1 0.

Si AWS Payment Cryptography est en mesure de le validerARQC, un http/200 est renvoyé. Si l'arqc n'est pas validé, il renverra une réponse 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"
             }
```

CAVVV7

Pour les transactions Visa Secure (3DS), une CAVV (valeur de vérification de l'authentification du titulaire de la carte) est générée par le serveur de contrôle d'accès de l'émetteur (). ACS CAVVC'est une preuve que l'authentification du titulaire de la carte a eu lieu, elle est unique pour chaque transaction d'authentification et est fournie par l'acquéreur dans le message d'autorisation. CAVVLa version 7 lie des données supplémentaires concernant la transaction à l'approbation, notamment des éléments tels que le nom du commerçant, le montant de l'achat et la date d'achat. De cette manière, il s'agit effectivement d'un hachage cryptographique de la charge utile de la transaction.

Sur le plan cryptographique, la CAVV V7 utilise l'CVValgorithme, mais les entrées ont toutes été modifiées/réutilisées. Veuillez consulter la documentation tierce ou Visa appropriée pour savoir comment produire les entrées nécessaires à la génération d'une charge utile CAVV V7.

Créez la clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CAVV-V7"},
{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
                    "Key": {
                        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/dnaeyrjgdjjtw6dk",
                        "KeyAttributes": {
                            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY",
                            "KeyClass": "SYMMETRIC_KEY",
                            "KeyAlgorithm": "TDES_2KEY",
                            "KeyModesOfUse": {
                                "Encrypt": false,
                                "Decrypt": false,
                                "Wrap": false,
                                "Unwrap": false,
                                "Generate": true,
                                "Sign": false,
                                "Verify": true,
                                "DeriveKey": false,
                                "NoRestrictions": false
                            }
                        },
                        "KeyCheckValue": "F3FB13",
                        "KeyCheckValueAlgorithm": "ANSI_X9_24",
                        "Enabled": true,
                        "Exportable": true,
                        "KeyState": "CREATE_COMPLETE",
                        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                        "CreateTimestamp": "2023-06-05T06:41:46.648000-07:00",
                        "UsageStartTimestamp": "2023-06-05T06:41:46.626000-07:00"
                    }
                }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/dnaeyrjgdjjtw6dk. Vous en aurez besoin à l'étape suivante.

Générer un CAVV V7

Example

Dans cet exemple, nous allons générer un CAVV V7 pour une transaction donnée avec des entrées telles que spécifiées dans les spécifications. Notez que pour cet algorithme, les champs peuvent être réutilisés/réutilisés. Il ne faut donc pas supposer que les étiquettes des champs correspondent aux entrées.

Pour tous les paramètres disponibles, voir CardVerificationValue1 dans le guide API de référence.

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
dnaeyrjgdjjtw6dk --primary-account-number=171234567890123 --generation-attributes
 CardVerificationValue1='{CardExpiryDate=9431,ServiceCode=431}'
```

```
{
    "KeyArn": "",
    "KeyCheckValue": "F3FB13",
    "ValidationData": "491"
}
```

Valider la CAVV V7

Example

Pour la validation, les entrées sont CVK les valeurs d'entrée calculées et celles CAVV fournies lors de la transaction à valider.

Pour tous les paramètres disponibles, voir CardVerificationValue1 dans le guide API de référence.

> **Note**
>
> CAVVn'est pas une valeur saisie par l'utilisateur (similaireCVV2) mais est calculée par l'émetteur. ACS Il convient de déterminer s'il doit toujours être validé lorsqu'il est fourni.

```
$ aws payment-cryptography-data verify-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/dnaeyrjgdjjtw6dk
 --primary-account-number=171234567890123 --verification-attributes
 CardVerificationValue1='{CardExpiryDate=9431,ServiceCode=431} --validation-data 491
```

```
{
                        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/dnaeyrjgdjjtw6dk",
                        "KeyCheckValue": "F3FB13",
                        "ValidationData": "491"
    }
```

Fonctions spécifiques à Mastercard

Rubriques

- DCVC3
- ARQC - CVN14/CVN15
- ARQC - CVN12/CVN13

DCVC3

DCVC3est antérieur EMV CSK aux CVN12 systèmes Mastercard et représente une autre approche d'utilisation des clés dynamiques. Il est parfois également réutilisé pour d'autres cas d'utilisation. Dans ce schéma, les entrées sont des données Track1/Track2 PANPSN, un nombre imprévisible et un compteur de transactions (). ATC

Créer une clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS,KeyClass=SYMMETRIC_KEY,KeyMod
 --tags='[{"Key":"KEY_PURPOSE","Value":"DCVC3"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
                    "Key": {
                        "KeyArn": "",
```

```
                    "KeyAttributes": {
                        "KeyUsage": "TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS",
                        "KeyClass": "SYMMETRIC_KEY",
                        "KeyAlgorithm": "TDES_2KEY",
                        "KeyModesOfUse": {
                            "Encrypt": false,
                            "Decrypt": false,
                            "Wrap": false,
                            "Unwrap": false,
                            "Generate": false,
                            "Sign": false,
                            "Verify": false,
                            "DeriveKey": true,
                            "NoRestrictions": false
                        }
                    },
                    "KeyCheckValue": "",
                    "KeyCheckValueAlgorithm": "ANSI_X9_24",
                    "Enabled": true,
                    "Exportable": true,
                    "KeyState": "CREATE_COMPLETE",
                    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                    "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
                    "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
                }
            }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple. Vous en aurez besoin à l'étape suivante.

Générez un DCVC3

Example

Bien qu'elle DCVC3 puisse être générée par une carte à puce, elle peut également être générée manuellement, comme dans cet exemple

```
$ aws payment-cryptography-data generate-card-validation-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk
 --primary-account-number=5413123456784808 --generation-attributes
 DynamicCardVerificationCode='{ApplicationTransactionCounter=0000,TrackData=524106000000069D13
```

```
{
```

```
            "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk",
            "KeyCheckValue": "08D7B4",
            "ValidationData": "865"
          }
```

Validez le DCVC3

Example

Dans cet exemple, nous allons valider unDCVC3. Notez qu'il ATC doit être fourni sous forme de nombre hexadécimal, par exemple un compteur de 11 doit être représenté par 000B. Le service attend une valeur à 3 chiffresDCVC3, donc si vous avez enregistré une valeur à 4 (ou 5) chiffres, tronquez simplement les caractères de gauche jusqu'à obtenir 3 chiffres (par exemple, 15321 devrait entraîner une valeur de donnée de validation de 321).

Si AWS Payment Cryptography est en mesure de valider, un http/200 est renvoyé. Si la valeur n'est pas validée, elle renverra une réponse http/400.

```
$ aws payment-cryptography-data verify-card-validation-data  --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6nl62t5ushfk
 --primary-account-number=5413123456784808 --verification-attributes
 DynamicCardVerificationCode='{ApplicationTransactionCounter=000B,TrackData=524106000000069D13
 --validation-data 398
```

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

ARQC - CVN14/CVN15

CVN14et CVN15 utilisez la EMVCSKméthode de dérivation des clés. Les données de transaction exactes varient entre ces deux méthodes. Consultez la documentation du schéma pour plus de détails sur la création du champ de données de transaction.

ARQC - CVN12/CVN13

CVN12et CVN13 sont des anciennes méthodes spécifiques à MasterCard pour les EMV transactions qui incorporent un nombre imprévisible dans la dérivation par transaction et utilisent également une

charge utile différente. Pour plus d'informations sur le contenu de la charge utile, veuillez contacter le programme.

Créer une clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS,KeyClass=SYMMETRIC_KEY,KeyMod
 --tags='[{"Key":"KEY_PURPOSE","Value":"CVN12"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
                    "Key": {
                        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/pw3s6nl62t5ushfk",
                        "KeyAttributes": {
                            "KeyUsage": "TR31_E0_EMV_MKEY_APP_CRYPTOGRAMS",
                            "KeyClass": "SYMMETRIC_KEY",
                            "KeyAlgorithm": "TDES_2KEY",
                            "KeyModesOfUse": {
                                "Encrypt": false,
                                "Decrypt": false,
                                "Wrap": false,
                                "Unwrap": false,
                                "Generate": false,
                                "Sign": false,
                                "Verify": false,
                                "DeriveKey": true,
                                "NoRestrictions": false
                            }
                        },
                        "KeyCheckValue": "08D7B4",
                        "KeyCheckValueAlgorithm": "ANSI_X9_24",
                        "Enabled": true,
                        "Exportable": true,
                        "KeyState": "CREATE_COMPLETE",
                        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                        "CreateTimestamp": "2024-03-07T06:41:46.648000-07:00",
                        "UsageStartTimestamp": "2024-03-07T06:41:46.626000-07:00"
                    }
                }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/pw3s6nl62t5ushfk. Vous en aurez besoin à l'étape suivante.

Validez le ARQC

Example

Dans cet exemple, nous allons valider un ARQC produit généré à l'aide de MastercardCVN12.

Si AWS Payment Cryptography est en mesure de le validerARQC, un http/200 est renvoyé. Si l'arqc n'est pas validé, il renverra une réponse http/400.

```
$ aws payment-cryptography-data verify-auth-request-cryptogram --auth-request-
cryptogram 31BE5D49F14A5F01 \
          --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6nl62t5ushfk \
          --major-key-derivation-mode EMV_OPTION_A \
          --transaction-data 0000000015000000000000000840000000000008402312120197695905
 \
          --session-key-derivation-attributes='{"Mastercard":{"PanSequenceNumber":"01"
 \

 ,"PrimaryAccountNumber":"9137631040001422","ApplicationTransactionCounter":"000B","Unpredictab
```

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

Fonctions spécifiques à American Express

Rubriques

- CSC1
- CSC2

CSC1

CSCLa version 1 est également connue sous le nom d'CSCalgorithme classique. Le service peut le fournir sous forme de numéro à 3,4 ou 5 chiffres.

Pour tous les paramètres disponibles, voir AmexCardSecurityCodeVersion1 dans le guide API de référence.

Créer une clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CSC1"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
                    "Key": {
                        "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/esh6hn7pxdtttzgq",
                        "KeyAttributes": {
                            "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY",
                            "KeyClass": "SYMMETRIC_KEY",
                            "KeyAlgorithm": "TDES_2KEY",
                            "KeyModesOfUse": {
                                "Encrypt": false,
                                "Decrypt": false,
                                "Wrap": false,
                                "Unwrap": false,
                                "Generate": true,
                                "Sign": false,
                                "Verify": true,
                                "DeriveKey": false,
                                "NoRestrictions": false
                            }
                        },
                        "KeyCheckValue": "8B5077",
                        "KeyCheckValueAlgorithm": "ANSI_X9_24",
                        "Enabled": true,
                        "Exportable": true,
                        "KeyState": "CREATE_COMPLETE",
                        "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                        "CreateTimestamp": "2023-06-05T06:41:46.648000-07:00",
                        "UsageStartTimestamp": "2023-06-05T06:41:46.626000-07:00"
                    }
                }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/esh6hn7pxdtttzgq. Vous en aurez besoin à l'étape suivante.

Générez un CSC1

Example

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
esh6hn7pxdtttzgq --primary-account-number=344131234567848 --generation-attributes
 AmexCardSecurityCodeVersion1='{CardExpiryDate=1224}' --validation-data-length 4
```

```
{
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
esh6hn7pxdtttzgq",
        "KeyCheckValue": "8B5077",
        "ValidationData": "3938"
      }
```

Validez le CSC1

Example

Dans cet exemple, nous allons valider unCSC1.

Si AWS Payment Cryptography est en mesure de valider, un http/200 est renvoyé. Si la valeur n'est pas validée, elle renverra une réponse http/400.

```
$ aws payment-cryptography-data verify-card-validation-data  --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/esh6hn7pxdtttzgq
 --primary-account-number=344131234567848 --verification-attributes
 AmexCardSecurityCodeVersion1='{CardExpiryDate=1224}''  --validation-data 3938
```

```
{
        "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
esh6hn7pxdtttzgq",
        "KeyCheckValue": "8B5077"
      }
```

CSC2

CSCLa version 2 est également connue sous le nom d'CSCalgorithme amélioré. Le service peut le fournir sous forme de numéro à 3,4 ou 5 chiffres.

Pour tous les paramètres disponibles, voir AmexCardSecurityCodeVersion2 dans le guide API de référence.

Créer une clé

```
$ aws payment-cryptography create-key --exportable --key-attributes
 KeyAlgorithm=TDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY,KeyModesO
 --tags='[{"Key":"KEY_PURPOSE","Value":"CSC1"},{"Key":"CARD_BIN","Value":"12345678"}]'
```

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de vérification clé (KCV).

```
{
            "Key": {
                "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
esh6hn7pxdtttzgq",
                "KeyAttributes": {
                    "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY",
                    "KeyClass": "SYMMETRIC_KEY",
                    "KeyAlgorithm": "TDES_2KEY",
                    "KeyModesOfUse": {
                        "Encrypt": false,
                        "Decrypt": false,
                        "Wrap": false,
                        "Unwrap": false,
                        "Generate": true,
                        "Sign": false,
                        "Verify": true,
                        "DeriveKey": false,
                        "NoRestrictions": false
                    }
                },
                "KeyCheckValue": "8B5077",
                "KeyCheckValueAlgorithm": "ANSI_X9_24",
                "Enabled": true,
                "Exportable": true,
                "KeyState": "CREATE_COMPLETE",
```

```
                "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "CreateTimestamp": "2023-06-05T06:41:46.648000-07:00",
                "UsageStartTimestamp": "2023-06-05T06:41:46.626000-07:00"
            }
        }
```

Prenez note de `KeyArn` ce qui représente la clé, par exemple arn:aws:payment-cryptography:us-east - 2:111122223333:key/esh6hn7pxdtttzgq. Vous en aurez besoin à l'étape suivante.

Générez un CSC2

Dans cet exemple, nous allons générer un CSC2 avec une longueur de 5. CSCpeut être généré avec une longueur de 3,4 ou 5. Pour American Express, PANs il doit comporter 15 chiffres et commencer par 34 ou 37.

Example

```
$ aws payment-cryptography-data generate-card-validation-data --key-
identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
esh6hn7pxdtttzgq --primary-account-number=344131234567848 --generation-attributes
 AmexCardSecurityCodeVersion2='{CardExpiryDate=1224,ServiceCode=999}' --validation-
data-length 4
```

```
{
"KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/erlm445qvunmvoda",
"KeyCheckValue": "BF1077",
"ValidationData": "3982"
}
```

Validez le CSC2

Example

Dans cet exemple, nous allons valider unCSC2.

Si AWS Payment Cryptography est en mesure de valider, un http/200 est renvoyé. Si la valeur n'est pas validée, elle renverra une réponse http/400.

```
$ aws payment-cryptography-data verify-card-validation-data  --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/erlm445qvunmvoda
```

```
--primary-account-number=344131234567848 --verification-attributes
AmexCardSecurityCodeVersion2='{CardExpiryDate=1224,ServiceCode=999}' --validation-data
3982
```

```
{
"KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/erlm445qvunmvoda",
"KeyCheckValue": "BF1077"
}
```

Facilitateurs d'acquisition et de paiement

Les acquéreurs, les fournisseurs de services de paiement et les facilitateurs de paiement ont généralement des exigences cryptographiques différentes de celles des émetteurs. Cas d'utilisation courants :

Déchiffrement des données

Les données (en particulier les données panoramiques) peuvent être cryptées par un terminal de paiement et doivent être déchiffrées par le backend. Le déchiffrement des données et le chiffrement des données prennent en charge diverses méthodes, notamment les techniques de dérivation TDES, AES et DUKPT. Le service AWS de cryptographie des paiements lui-même est également conforme à la norme PCI P2PE et est enregistré en tant que composant de déchiffrement PCI P2PE.

TranslatePin

Pour garantir la conformité au code PIN PCI, les systèmes d'acquisition ne doivent pas afficher le code PIN du titulaire de la carte en clair une fois qu'il a été saisi sur un appareil sécurisé. Par conséquent, pour transmettre le code PIN du terminal à un système en aval (tel qu'un réseau de paiement ou un émetteur), il est nécessaire de le rechiffrer à l'aide d'une clé différente de celle utilisée par le terminal de paiement. Translate Pin y parvient en convertissant un code PIN crypté d'une clé à une autre en toute sécurité avec le servicebbb. À l'aide de cette commande, les broches peuvent être converties entre différents schémas tels que la dérivation TDES, AES et DUKPT et les formats de blocs de broches tels que ISO-0, ISO-3 et ISO-4.

VerifyMac

Les données d'un terminal de paiement peuvent être enregistrées sur MAC pour s'assurer qu'elles n'ont pas été modifiées pendant le transport. Verify Mac et GenerateMac prend en charge

diverses techniques utilisant des clés symétriques, notamment les techniques de dérivation TDES, AES et DUKPT à utiliser avec l'algorithme 1 ISO-9797-1, l'algorithme ISO-9797-1 3 (Retail MAC) et les techniques CMAC.

Rubriques supplémentaires

- Utilisation de touches dynamiques

Utilisation de touches dynamiques

Les clés dynamiques permettent d'utiliser des clés à usage unique ou limité pour des opérations cryptographiques telles que`EncryptData`. Ce flux peut être utilisé lorsque le matériel clé change fréquemment (par exemple à chaque transaction par carte) et que l'on souhaite éviter d'importer le matériel clé dans le service. Les clés de courte durée peuvent être utilisées dans le cadre de SoftPOS/MPOC ou d'autres solutions.

> **Note**
>
> Cela peut être utilisé à la place du flux classique utilisant la cryptographie des AWS paiements, dans lequel les clés cryptographiques sont créées ou importées dans le service et les clés sont spécifiées à l'aide d'un alias de clé ou d'un arn de clé.

Les opérations suivantes prennent en charge les clés dynamiques :

- EncryptData
- DecryptData
- ReEncryptData
- TranslatePin

Déchiffrement de données

L'exemple suivant montre l'utilisation de clés dynamiques avec la commande de déchiffrement. Dans ce cas, l'identifiant de clé est la clé d'encapsulation (KEK) qui sécurise la clé de déchiffrement (fournie dans le paramètre wrapped-key au format TR-31). La clé encapsulée doit être l'objectif principal de D0 à utiliser avec la commande de déchiffrement ainsi qu'un mode d'utilisation de B ou D.

Example

```
$ aws payment-cryptography-data decrypt-data --key-identifier
 arn:aws:payment-cryptography:us-east-2:111122223333:key/ov6icy4ryas4zcza
 --cipher-text 12341234123412341234123412341234123A --decryption-attributes
 'Symmetric={Mode=CBC,InitializationVector=1234123412341234}'   --wrapped-key
 WrappedKeyMaterial={"Tr31KeyBlock"="D0112D0TN00E0000B05A6E82D7FC68B95C84306634B0000DA4701BE9BC
```

```
{
   "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza",
   "KeyCheckValue": "0A3674",
   "PlainText": "2E138A746A0032023BEF5B85BA5060BA"
}
```

Traduire une épingle

L'exemple suivant montre l'utilisation de touches dynamiques associées à la commande translate pin pour passer d'une clé dynamique à une clé de travail semi-statique pour acquéreur (AWK). Dans ce cas, l'identifiant de clé entrante est la clé d'encapsulation (KEK) qui protège la clé de chiffrement dynamique par code PIN (PEK) fournie au format TR-31. La clé encapsulée doit avoir un objectif clé `P0` ainsi qu'un mode d'utilisation de B ou D. L'identifiant de clé sortante est une clé de type `TR31_P0_PIN_ENCRYPTION_KEY` et un mode d'utilisation de Encrypt=True, Wrap=True

Example

```
$ aws payment-cryptography-data translate-pin-data --encrypted-pin-block
 "C7005A4C0FA23E02" --incoming-translation-
attributes=IsoFormat0='{PrimaryAccountNumber=171234567890123}'
 --incoming-key-identifier alias/PARTNER1_KEK  --outgoing-key-
identifier alias/ACQUIRER_AWK_PEK --outgoing-translation-attributes
 IsoFormat0="{PrimaryAccountNumber=171234567890123}"  --incoming-wrapped-key
 WrappedKeyMaterial={"Tr31KeyBlock"="D0112P0TB00S0000EB5D8E63076313162B04245C8CE351C956EA4A16CC
```

```
{
   "PinBlock": "2E66192BDA390C6F",
   "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
ov6icy4ryas4zcza",
```

```
    "KeyCheckValue": "0A3674"
}
```

Sécurité dans le domaine de la cryptographie des AWS paiements

La sécurité du cloud AWS est la priorité absolue. En tant que AWS client, vous bénéficiez d'un centre de données et d'une architecture réseau conçus pour répondre aux exigences des entreprises les plus sensibles en matière de sécurité.

La sécurité est une responsabilité partagée entre vous AWS et vous. Le modèle de responsabilité partagée décrit ceci comme la sécurité du cloud et la sécurité dans le cloud :

- Sécurité du cloud :AWS est chargée de protéger l'infrastructure qui exécute les AWS services dans le AWS cloud. AWS vous fournit également des services que vous pouvez utiliser en toute sécurité. Des auditeurs tiers testent et vérifient régulièrement l'efficacité de notre sécurité dans le cadre des programmes de AWS conformité Programmes de de conformité. Pour en savoir plus sur les programmes de conformité qui s'appliquent à la cryptographie des AWS paiements, voir AWSServices concernés par programme de conformité AWS .
- Sécurité dans le cloud : votre responsabilité est déterminée par le AWS service que vous utilisez. Vous êtes également responsable d'autres facteurs, y compris de la sensibilité de vos données, des exigences de votre entreprise, ainsi que de la législation et de la réglementation applicables.

Cette rubrique vous aide à comprendre comment appliquer le modèle de responsabilité partagée lors de l'utilisation de la cryptographie des AWS paiements. Il vous explique comment configurer le chiffrement des AWS paiements pour atteindre vos objectifs de sécurité et de conformité. Vous apprendrez également à utiliser d'autres AWS services qui vous aident à surveiller et à sécuriser vos ressources de cryptographie des AWS paiements.

Rubriques

Protection des données dans le cadre de la cryptographie des AWS paiements

Le modèle de responsabilité AWS partagée Le modèle de s'applique à la protection des données dans le domaine de la cryptographie des AWS paiements. Comme décrit dans ce modèle, AWS est chargé de protéger l'infrastructure mondiale qui gère tous les AWS Cloud. La gestion du contrôle de votre contenu hébergé sur cette infrastructure relève de votre responsabilité. Vous êtes également responsable des tâches de configuration et de gestion de la sécurité des Services AWS que vous utilisez. Pour plus d'informations sur la confidentialité des données, consultez la section Confidentialité des données FAQ. Pour plus d'informations sur la protection des données en Europe, consultez le modèle de responsabilitéAWS partagée et le billet de GDPR blog sur le blog sur la AWS sécurité.

À des fins de protection des données, nous vous recommandons de protéger les Compte AWS informations d'identification et de configurer les utilisateurs individuels avec AWS IAM Identity Center ou AWS Identity and Access Management (IAM). Ainsi, chaque utilisateur se voit attribuer uniquement les autorisations nécessaires pour exécuter ses tâches. Nous vous recommandons également de sécuriser vos données comme indiqué ci-dessous :

- Utilisez l'authentification multifactorielle (MFA) pour chaque compte.
- UtilisezSSL/TLSpour communiquer avec les AWS ressources. Nous avons besoin de la TLS version 1.2 et recommandons la TLS version 1.3.
- Configuration API et journalisation de l'activité des utilisateurs avec AWS CloudTrail.
- Utilisez des solutions de AWS chiffrement, ainsi que tous les contrôles de sécurité par défaut qu'ils contiennent Services AWS.
- Utilisez des services de sécurité gérés avancés tels qu'Amazon Macie, qui contribuent à la découverte et à la sécurisation des données sensibles stockées dans Amazon S3.
- Si vous avez besoin de FIPS 140 à 3 modules cryptographiques validés pour accéder AWS via une interface de ligne de commande ou unAPI, utilisez un point de terminaison. FIPS Pour plus d'informations sur les FIPS points de terminaison disponibles, voir Federal Information Processing Standard (FIPS) 140-3.

Nous vous recommandons fortement de ne jamais placer d'informations confidentielles ou sensibles, telles que les adresses e-mail de vos clients, dans des balises ou des champs de texte libre tels que le champ Name (Nom). Cela inclut lorsque vous travaillez avec AWS Payment Cryptography ou autre

Services AWS à l'aide de la consoleAPI, AWS CLI, ou AWS SDKs. Toutes les données que vous entrez dans des balises ou des champs de texte de forme libre utilisés pour les noms peuvent être utilisées à des fins de facturation ou dans les journaux de diagnostic. Si vous fournissez un URL à un serveur externe, nous vous recommandons vivement de ne pas inclure d'informations d'identification dans le URL afin de valider votre demande auprès de ce serveur.

AWSLa cryptographie des paiements stocke et protège vos clés de chiffrement des paiements afin de les rendre hautement disponibles tout en vous offrant un contrôle d'accès solide et flexible.

Rubriques

Protection des éléments de clé

Par défaut, AWS Payment Cryptography protège le contenu des clés cryptographiques pour les clés de paiement gérées par le service. En outre, AWS Payment Cryptography propose des options pour importer des éléments clés créés en dehors du service. Pour obtenir des informations techniques sur les clés de paiement et les éléments clés, consultez la section Détails cryptographiques de la cryptographie des AWS paiements.

Chiffrement des données

Les données de la cryptographie des AWS paiements se composent des clés de cryptographie des AWS paiements, du matériel de clé de cryptage qu'elles représentent et de leurs attributs d'utilisation. Le contenu clé n'existe en texte clair que dans les modules de sécurité matériels de cryptographie des AWS paiements (HSMs) et uniquement lorsqu'ils sont utilisés. Dans le cas contraire, le matériel et les attributs clés sont chiffrés et stockés dans un stockage persistant durable.

Les éléments clés générés ou chargés AWS par la cryptographie des paiements pour les clés de paiement ne quittent jamais les limites de la cryptographie des AWS paiements sans être chiffrésHSMs. Il peut être exporté crypté par des API opérations AWS de cryptographie des paiements.

Chiffrement au repos

AWSLa cryptographie des paiements génère des informations clés pour les clés de paiement dans PCI PTS HSM HSMs -listed. Lorsqu'ils ne sont pas utilisés, les éléments clés sont chiffrés par une HSM clé et enregistrés sur un support de stockage durable et persistant. Le matériel clé pour les clés de cryptographie de paiement et les clés de chiffrement qui protègent le matériel clé ne le HSMs quittent jamais sous forme de texte clair.

Le chiffrement et la gestion des éléments clés pour les clés de chiffrement des paiements sont entièrement gérés par le service.

Pour plus de détails, consultez la section Détails cryptographiques du service de gestion des AWS clés.

Chiffrement en transit

Les éléments clés générés ou chargés AWS par Payment Cryptography pour les clés de paiement ne sont jamais exportés ou transmis en texte clair dans le API cadre des opérations de cryptographie des AWS paiements. AWSLa cryptographie des paiements utilise des identifiants de clé pour représenter les clés dans API les opérations.

Cependant, certaines API opérations AWS de cryptographie des paiements exportent des clés chiffrées par une clé d'échange de clés précédemment partagée ou asymétrique. Les clients peuvent également utiliser API des opérations pour importer des informations clés cryptées pour les clés de paiement.

Tous les API appels AWS de cryptographie des paiements doivent être signés et transmis à l'aide de Transport Layer Security (TLS). AWSLa cryptographie des paiements nécessite TLS des versions et des suites de chiffrement définies par PCI le terme « cryptographie forte ». Tous les points de terminaison de service prennent en charge le format TLS 1.0—1.3 et le post-quantum hybride. TLS

Pour plus de détails, consultez la section Détails cryptographiques du service de gestion des AWS clés.

Confidentialité du trafic inter-réseau

AWSLa cryptographie des paiements prend en charge une console de AWS gestion et un ensemble d'APIopérations qui vous permettent de créer et de gérer des clés de paiement et de les utiliser dans des opérations cryptographiques.

AWSPayment Cryptography prend en charge deux options de connectivité réseau allant de votre réseau privé à. AWS

- Une IPSec VPN connexion via Internet.
- AWSDirect Connect, qui relie votre réseau interne à un emplacement AWS Direct Connect via un câble Ethernet à fibre optique standard.

Tous les API appels de cryptographie des paiements doivent être signés et transmis à l'aide de Transport Layer Security (TLS). Les appels nécessitent également une suite de chiffrement moderne qui prend en charge une confidentialité persistante et parfaite. Le trafic vers les modules de sécurité matériels (HSMs) qui stockent les informations clés pour les clés de paiement est autorisé uniquement à partir d'APIhôtes connus de cryptographie des AWS paiements via le réseau AWS interne.

Pour vous connecter directement à AWS Payment Cryptography depuis votre cloud privé virtuel (VPC) sans envoyer de trafic sur l'Internet public, utilisez des VPC points de terminaison alimentés par. AWS PrivateLink Pour plus d'informations, consultez la section Connexion à la cryptographie des AWS paiements via un VPC point de terminaison.

AWSLa cryptographie des paiements prend également en charge une option hybride d'échange de clés post-quantiques pour le protocole de chiffrement réseau Transport Layer Security (TLS). Vous pouvez utiliser cette option TLS lorsque vous vous connectez à des points de API terminaison AWS de chiffrement des paiements.

Résilience dans la cryptographie des AWS paiements

AWS l'infrastructure mondiale est construite autour AWS des régions et des zones de disponibilité. Les régions fournissent plusieurs zones de disponibilité physiquement séparées et isolées, reliées par un réseau à latence faible, à débit élevé et à forte redondance. Avec les zones de disponibilité, vous pouvez concevoir et exploiter des applications et des bases de données qui basculent automatiquement d'une zone à l'autre sans interruption. Les zones de disponibilité sont davantage disponibles, tolérantes aux pannes et ont une plus grande capacité de mise à l'échelle que les infrastructures traditionnelles à un ou plusieurs centres de données.

Pour plus d'informations sur AWS les régions et les zones de disponibilité, consultez la section Infrastructure AWS mondiale.

Isolement régional

AWSLa cryptographie des paiements est un service régional disponible dans plusieurs régions.

La conception isolée au niveau régional de la cryptographie des AWS paiements garantit qu'un problème de disponibilité dans une AWS région ne peut affecter le fonctionnement de la cryptographie des AWS paiements dans aucune autre région. AWSLa cryptographie des paiements est conçue pour garantir l'absence d'interruption planifiée, toutes les mises à jour logicielles et les opérations de dimensionnement étant effectuées de manière fluide et imperceptible.

Le contrat de niveau de service de cryptographie des AWS paiements (SLA) inclut un engagement de service de 99,99 % pour l'ensemble de la cryptographie des paiements. APIs Pour respecter cet engagement, AWS Payment Cryptography garantit que toutes les données et informations d'autorisation requises pour exécuter une API demande sont disponibles sur tous les hôtes régionaux qui reçoivent la demande.

L'infrastructure AWS de cryptographie des paiements est répliquée dans au moins trois zones de disponibilité (AZs) dans chaque région. Pour garantir que les défaillances de plusieurs hôtes n'affectent pas les AWS performances de la cryptographie des AWS paiements, la cryptographie des paiements est conçue pour traiter le trafic client AZs en provenance de n'importe quelle région.

Les modifications que vous apportez aux propriétés ou aux autorisations d'une clé de paiement sont reproduites sur tous les hôtes de la région afin de garantir que les demandes ultérieures puissent être traitées correctement par tous les hôtes de la région. Les demandes d'opérations cryptographiques utilisant votre clé de paiement sont transmises à une flotte de modules de sécurité matériels de cryptographie des AWS paiements (HSMs), chacun d'entre eux pouvant effectuer l'opération avec la clé de paiement.

Conception à locataires multiples

La conception multi-tenant de la cryptographie des AWS paiements lui permet de garantir la disponibilité SLA et de maintenir des taux de demandes élevés, tout en protégeant la confidentialité de vos clés et de vos données.

Plusieurs mécanismes de renforcement de l'intégrité sont déployés pour garantir que la clé de paiement que vous avez spécifiée pour l'opération cryptographique est toujours celle qui est utilisée.

Le contenu clé en texte clair de vos clés de cryptographie de paiement est largement protégé. Le matériel clé est crypté HSM dès sa création, et le matériel clé crypté est immédiatement transféré

vers un stockage sécurisé. La clé cryptée est récupérée et déchiffrée HSM juste à temps pour être utilisée. La clé en texte clair ne reste en HSM mémoire que le temps nécessaire à l'exécution de l'opération cryptographique. Le contenu clé en texte brut ne quitte jamais le HSMs ; il n'est jamais écrit dans un stockage permanent.

Pour plus d'informations sur les mécanismes utilisés par la cryptographie des AWS paiements pour sécuriser vos clés, consultez la section Détails cryptographiques de la cryptographie des AWS paiements.

Sécurité de l'infrastructure dans AWS Payment Cryptography

En tant que service géré, AWS Payment Cryptography il est protégé par les procédures de sécurité du réseau AWS mondial décrites dans le livre blanc Amazon Web Services : présentation des processus de sécurité.

Vous utilisez API les appels AWS publiés pour accéder AWS Payment Cryptography via le réseau. Les clients doivent prendre en charge Transport Layer Security (TLS) 1.2 ou version ultérieure. Les clients doivent également prendre en charge les suites de chiffrement parfaitement confidentielles (), telles que Ephemeral Diffie-Hellman (PFS) ou Elliptic Curve Ephemeral Diffie-Hellman (DHE). ECDHE La plupart des systèmes modernes tels que Java 7 et les versions ultérieures prennent en charge ces modes.

En outre, les demandes doivent être signées à l'aide d'un identifiant de clé d'accès et d'une clé d'accès secrète associés à un IAM principal. Vous pouvez également utiliser AWS Security Token Service (AWS STS) pour générer des informations d'identification de sécurité temporaires et signer les demandes.

Isolement des hôtes physiques

La sécurité de l'infrastructure physique utilisée par AWS Payment Cryptography est soumise aux contrôles décrits dans la section Sécurité physique et environnementale d'Amazon Web Services : présentation des processus de sécurité. Vous trouverez plus de détails dans les rapports de conformité et les résultats d'audit tiers répertoriés dans la section précédente.

AWSLa cryptographie des paiements est prise en charge par des modules de sécurité matériels dédiés commercial-off-the-shelf PCI PTS HSM répertoriés ()HSMs. Le matériel clé des clés AWS de cryptographie de paiement est stocké uniquement dans la mémoire volatile duHSMs, et uniquement pendant que la clé de cryptographie de paiement est utilisée. HSMsse trouvent dans des racks à

accès contrôlé au sein des centres de données Amazon qui appliquent un double contrôle pour tout accès physique. Pour obtenir des informations détaillées sur le fonctionnement de la cryptographie des AWS paiementsHSMs, consultez la section Détails cryptographiques de la cryptographie des AWS paiements.

Connexion à la cryptographie des AWS paiements via un terminal VPC

Vous pouvez vous connecter directement à AWS Payment Cryptography via un point de terminaison d'interface privé dans votre cloud privé virtuel (VPC). Lorsque vous utilisez un point de VPC terminaison d'interface, la communication entre vous VPC et AWS Payment Cryptography s'effectue entièrement au sein du AWS réseau.

AWS La cryptographie des paiements prend en charge les points de terminaison Amazon Virtual Private Cloud (AmazonVPC) alimentés par. AWS PrivateLink Chaque VPC point de terminaison est représenté par une ou plusieurs interfaces réseau élastiques (ENIs) avec des adresses IP privées dans vos VPC sous-réseaux.

Le point de VPC terminaison de l'interface vous connecte VPC directement à AWS Payment Cryptography sans passerelle Internet, NAT appareil, VPN connexion ou AWS Direct Connect connexion. Les instances de votre navigateur VPC n'ont pas besoin d'adresses IP publiques pour communiquer avec AWS Payment Cryptography.

Régions

AWS La cryptographie des paiements prend en charge les VPC terminaux et les politiques relatives aux VPC terminaux Régions AWS dans tous les cas où la cryptographie des AWS paiements est prise en charge.

Rubriques

- Considérations relatives aux terminaux AWS de cryptographie VPC des paiements
- Création d'un VPC point de terminaison pour la cryptographie des AWS paiements
- Connexion à un point de AWS terminaison de chiffrement VPC des paiements
- Contrôle de l'accès à un VPC terminal
- Utilisation d'un VPC point de terminaison dans une déclaration de politique
- Enregistrement de votre VPC terminal

Considérations relatives aux terminaux AWS de cryptographie VPC des paiements

> **Note**
>
> Bien que les VPC points de terminaison vous permettent de vous connecter au service dans une seule zone de disponibilité (AZ), nous vous recommandons de vous connecter à trois zones de disponibilité pour des raisons de haute disponibilité et de redondance.

Avant de configurer un point de VPC terminaison d'interface pour le chiffrement des AWS paiements, consultez la rubrique Propriétés et limites du point de terminaison d'interface dans le AWS PrivateLink Guide.

AWS La prise en charge de la cryptographie des paiements pour un VPC terminal inclut les éléments suivants.

- Vous pouvez utiliser votre VPC point de terminaison pour appeler toutes les opérations du plan de contrôle AWS de cryptographie des paiements et toutes les opérations du plan de données AWS de cryptographie des paiements à partir d'un. VPC
- Vous pouvez créer un point de VPC terminaison d'interface qui se connecte à un point de terminaison de région de cryptographie des AWS paiements.
- AWS La cryptographie des paiements comprend un plan de contrôle et un plan de données. Vous pouvez choisir de configurer un ou les deux sous-services, mais chacun est configuré séparément.
- Vous pouvez utiliser AWS CloudTrail les journaux pour vérifier votre utilisation des clés de chiffrement des AWS paiements via le VPC terminal. Pour plus de détails, consultez Enregistrement de votre VPC terminal.

Création d'un VPC point de terminaison pour la cryptographie des AWS paiements

Vous pouvez créer un VPC point de terminaison pour AWS le chiffrement des paiements à l'aide de la VPC console Amazon ou d'Amazon VPCAPI. Pour plus d'informations, consultez Création d'un point de terminaison d'interface dans le Guide AWS PrivateLink .

- Pour créer un VPC point de terminaison pour AWS le chiffrement des paiements, utilisez les noms de service suivants :

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

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

  Par exemple, dans la région USA Ouest (Oregon) (`us-west-2`), les noms des services seraient les suivants :

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

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

Pour faciliter l'utilisation du VPC point de terminaison, vous pouvez activer un DNSnom privé pour votre VPC point de terminaison. Si vous sélectionnez l'option Activer le DNS nom, le nom d'DNShôte standard de cryptographie des AWS paiements correspond à votre VPC point de terminaison. Par exemple, `https://controlplane.payment-cryptography.us-west-2.amazonaws.com` serait résolu en un point de VPC terminaison connecté au nom du service`com.amazonaws.us-west-2.payment-cryptography.controlplane`.

Cette option facilite l'utilisation du VPC point de terminaison. Les AWS SDKs et AWS CLI utilisent le DNS nom d'hôte standard AWS de cryptographie des paiements par défaut. Vous n'avez donc pas besoin de spécifier le VPC point de terminaison URL dans les applications et les commandes.

Pour de plus amples informations, veuillez consulter Accès à un service via un point de terminaison d'interface dans le Guide AWS PrivateLink .

Connexion à un point de AWS terminaison de chiffrement VPC des paiements

Vous pouvez vous connecter à AWS Payment Cryptography via le VPC terminal en utilisant un AWS SDK, le AWS CLI ou AWS Tools for PowerShell. Pour spécifier le VPC point de terminaison, utilisez son DNS nom.

Par exemple, cette commande list-keys utilise le `endpoint-url` paramètre pour spécifier le point de terminaison. VPC Pour utiliser une commande comme celle-ci, remplacez l'exemple d'ID de point de VPC terminaison par un identifiant de votre compte.

```
$ aws payment-cryptography list-keys --endpoint-url https://
vpce-1234abcdf5678c90a-09p7654s-us-east-1a.ec2.us-east-1.vpce.amazonaws.com
```

Si vous avez activé les noms d'hôte privés lors de la création de votre VPC point de terminaison, vous n'avez pas besoin de spécifier le VPC point de terminaison URL dans vos CLI commandes ou dans la configuration de l'application. Le DNS nom d'hôte standard AWS de Payment Cryptography correspond à votre VPC terminal. Le AWS CLI et SDKs utilisez ce nom d'hôte par défaut, afin que vous puissiez commencer à utiliser le VPC point de terminaison pour vous connecter à un point de terminaison régional de cryptographie des AWS paiements sans rien modifier dans vos scripts et applications.

Pour utiliser des noms d'hôte privés, les `enableDnsSupport` attributs `enableDnsHostnames` et de votre nom VPC doivent être définis sur. `true` Pour définir ces attributs, utilisez l'ModifyVpcAttributeopération. Pour plus de détails, consultez la section Afficher et mettre à jour vos DNS attributs VPC dans le guide de VPC l'utilisateur Amazon.

Contrôle de l'accès à un VPC terminal

Pour contrôler l'accès à votre VPC terminal pour la cryptographie des AWS paiements, associez une politique de point de VPC terminaison à votre VPC terminal. La politique de point de terminaison détermine si les principaux peuvent utiliser le VPC point de terminaison pour appeler des opérations de cryptographie de AWS paiement avec des ressources de cryptographie AWS de paiement spécifiques.

Vous pouvez créer une politique de point de VPC terminaison lorsque vous créez votre point de terminaison, et vous pouvez modifier la politique de VPC point de terminaison à tout moment. Utilisez la console VPC de gestion ou les ModifyVpcEndpointopérations CreateVpcEndpointor. Vous pouvez également créer et modifier une politique de point de VPC terminaison à l'aide d'un AWS CloudFormation modèle. Pour obtenir de l'aide sur l'utilisation de la console de VPC gestion, consultez les sections Création d'un point de terminaison d'interface et Modification d'un point de terminaison d'interface dans le AWS PrivateLink Guide.

Rubriques

- À propos des politiques relatives aux VPC terminaux

- Politique relative aux VPC terminaux par défaut
- Création d'une politique de point de VPC terminaison
- Afficher une politique de point de VPC terminaison

À propos des politiques relatives aux VPC terminaux

Pour qu'une demande AWS de cryptographie de paiement utilisant un VPC point de terminaison aboutisse, le principal doit obtenir des autorisations provenant de deux sources :

- Une politique basée sur l'identité doit autoriser le principal à appeler l'opération sur la ressource (clés de chiffrement des AWS paiements ou alias).
- Une politique de VPC point de terminaison doit donner au principal l'autorisation d'utiliser le point de terminaison pour effectuer la demande.

Par exemple, une politique en matière de clés peut autoriser un principal à appeler Decrypt pour une clé de cryptographie AWS de paiement particulière. Cependant, la politique du VPC point de terminaison peut ne pas autoriser ce principal à faire appel `Decrypt` à ces clés de cryptographie de AWS paiement en utilisant le point de terminaison.

Une politique de VPC point de terminaison peut également autoriser un principal à utiliser le point de terminaison pour appeler StopKeyUsagecertaines clés AWS de cryptographie de paiement. Mais si le principal ne dispose pas de ces autorisations en vertu d'une IAM politique, la demande échoue.

Politique relative aux VPC terminaux par défaut

Chaque VPC point de VPC terminaison possède une politique de point de terminaison, mais vous n'êtes pas obligé de la spécifier. Si vous ne spécifiez pas de politique, la politique de point de terminaison par défaut autorise toutes les opérations effectuées par tous les principaux sur toutes les ressources du point de terminaison.

Toutefois, pour les ressources AWS de cryptographie des paiements, le principal doit également être autorisé à appeler l'opération à partir d'une IAMpolitique. Par conséquent, en pratique, la politique par défaut indique que si un principal a l'autorisation d'appeler une opération sur une ressource, il peut également l'appeler à l'aide du point de terminaison.

```
{
  "Statement": [
    {
```

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

Pour permettre aux principaux d'utiliser le VPC point de terminaison uniquement pour un sous-ensemble de leurs opérations autorisées, créez ou mettez à jour la politique du VPC point de terminaison.

Création d'une politique de point de VPC terminaison

Une politique de VPC point de terminaison détermine si un principal est autorisé à utiliser le VPC point de terminaison pour effectuer des opérations sur une ressource. Pour les ressources de cryptographie des AWS paiements, le mandant doit également être autorisé à effectuer les opérations conformément à une IAMpolitique.

Chaque déclaration de politique relative aux VPC terminaux nécessite les éléments suivants :

- Le principal qui peut exécuter des actions.
- Les actions qui peuvent être effectuées.
- Les ressources sur lesquelles les actions peuvent être exécutées.

La déclaration de politique ne précise pas le VPC point de terminaison. Elle s'applique plutôt à tout VPC point de terminaison auquel la politique est attachée. Pour plus d'informations, consultez la section Contrôle de l'accès aux services avec des VPC points de terminaison dans le guide de VPC l'utilisateur Amazon.

Voici un exemple de politique de VPC point de terminaison pour la cryptographie des AWS paiements. Lorsqu'il est connecté à un VPC point de terminaison, cette politique permet `ExampleUser` d'utiliser le VPC point de terminaison pour appeler les opérations spécifiées sur les clés de cryptographie de AWS paiement spécifiées. Avant d'utiliser une politique comme celle-ci, remplacez l'exemple d'identifiant principal et de clé par des valeurs valides provenant de votre compte.

```
{
   "Statement":[
      {
```

```
        "Sid": "AllowDecryptAndView",
        "Principal": {"AWS": "arn:aws:iam::111122223333:user/ExampleUser"},
        "Effect":"Allow",
        "Action": [
            "payment-cryptography:Decrypt",
            "payment-cryptography:GetKey",
            "payment-cryptography:ListAliases",
            "payment-cryptography:ListKeys",
            "payment-cryptography:GetAlias"
         ],
        "Resource": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h"
      }
   ]
}
```

AWS CloudTrail enregistre toutes les opérations qui utilisent le VPC point de terminaison. Toutefois, vos CloudTrail journaux n'incluent pas les opérations demandées par les principaux sur d'autres comptes ni les opérations relatives aux clés de chiffrement des AWS paiements sur d'autres comptes.

Ainsi, vous souhaiterez peut-être créer une politique de point de VPC terminaison qui empêche les principaux titulaires de comptes externes d'utiliser le VPC point de terminaison pour effectuer des opérations de cryptographie de AWS paiement sur n'importe quelle clé du compte local.

L'exemple suivant utilise la clé de condition PrincipalAccount globale aws : pour refuser l'accès à tous les principaux pour toutes les opérations sur toutes les clés de chiffrement des AWS paiements, sauf si le principal se trouve sur le compte local. Avant d'utiliser une politique comme celle-ci, remplacez l'ID de compte d'exemple par un ID valide.

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

```
      }
    }
  ]
}
```

Afficher une politique de point de VPC terminaison

Pour consulter la politique de VPC point de terminaison d'un point de terminaison, utilisez la console de VPC gestion ou l'DescribeVpcEndpointsopération.

La AWS CLI commande suivante permet d'obtenir la politique du point de terminaison avec l'ID de point de VPC terminaison spécifié.

Avant d'utiliser cette commande, remplacez l'exemple d'ID de point de terminaison d'exemple par un ID valide provenant de votre compte.

```
$ aws ec2 describe-vpc-endpoints \
--query 'VpcEndpoints[?VpcEndpointId==`vpce-1234abcdf5678c90a`].[PolicyDocument]'
--output text
```

Utilisation d'un VPC point de terminaison dans une déclaration de politique

Vous pouvez contrôler l'accès aux ressources et aux opérations de cryptographie des AWS paiements lorsque la demande provient VPC ou utilise un VPC point de terminaison. Pour ce faire, utilisez-en une et une IAMpolitique

- Utilisez la clé de `aws:sourceVpce` condition pour accorder ou restreindre l'accès en fonction du VPC point de terminaison.
- Utilisez la clé de `aws:sourceVpc` condition pour accorder ou restreindre l'accès en fonction de VPC celle qui héberge le point de terminaison privé.

> **Note**
>
> La clé de `aws:sourceIP` condition n'est pas effective lorsque la demande provient d'un point de VPCterminaison Amazon. Pour limiter les demandes à un VPC point de terminaison, utilisez les clés de `aws:sourceVpc` condition `aws:sourceVpce` ou. Pour plus d'informations, consultez la section Gestion des identités et des accès pour les VPC terminaux et les services de point de VPC terminaison dans le AWS PrivateLink Guide.

Vous pouvez utiliser ces clés de condition globales pour contrôler l'accès aux clés de chiffrement des AWS paiements, aux alias et aux opérations de CreateKeyce type qui ne dépendent d'aucune ressource en particulier.

Par exemple, l'exemple de politique de clé suivant permet à un utilisateur d'effectuer des opérations cryptographiques particulières avec des clés de cryptographie de AWS paiement uniquement lorsque la demande utilise le point de VPC terminaison spécifié, bloquant ainsi l'accès à la fois depuis Internet et les connexions (si elles sont configurées). Lorsqu'un utilisateur fait une demande à AWS Payment Cryptography, l'identifiant du VPC point de terminaison indiqué dans la demande est comparé à la valeur de la clé de `aws:sourceVpce` condition indiquée dans la politique. S'il n'y a pas de concordance, la requête est refusée.

Pour utiliser une politique comme celle-ci, remplacez l' Compte AWS identifiant fictif et le point de VPC terminaison IDs par des valeurs valides pour votre compte.

```
{
    "Id": "example-key-1",
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "Enable IAM policies",
            "Effect": "Allow",
            "Principal": {"AWS":["111122223333"]},
            "Action": ["payment-cryptography:*"],
            "Resource": "*"
        },
        {
            "Sid": "Restrict usage to my VPC endpoint",
            "Effect": "Deny",
            "Principal": "*",
            "Action": [
                "payment-cryptography:Encrypt",
                "payment-cryptography:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringNotEquals": {
                    "aws:sourceVpce": "vpce-1234abcdf5678c90a"
                }
            }
        }
```

```
    ]
}
```

Vous pouvez également utiliser la clé de `aws:sourceVpc` condition pour restreindre l'accès à vos clés de cryptographie de AWS paiement en fonction du terminal VPC dans lequel réside le VPC terminal.

L'exemple de politique de clé suivant autorise les commandes qui gèrent les clés de cryptographie des AWS paiements uniquement lorsqu'elles proviennent`vpc-12345678`. En outre, il autorise les commandes qui utilisent les clés de cryptographie des AWS paiements pour les opérations cryptographiques uniquement lorsqu'elles proviennent de. `vpc-2b2b2b2b` Vous pouvez utiliser une politique comme celle-ci si une application s'exécute dans l'une d'entre ellesVPC, mais vous en utilisez une seconde, isolée VPC pour les fonctions de gestion.

Pour utiliser une politique comme celle-ci, remplacez l' Compte AWS identifiant fictif et le point de VPC terminaison IDs par des valeurs valides pour votre compte.

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

```
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "payment-cryptography:Encrypt","payment-cryptography:Decrypt"
            ],
            "Resource": "*",
            "Condition": {
                "StringEquals": {
                    "aws:sourceVpc": "vpc-2b2b2b2b"
                }
            }
        },
        {
            "Sid": "Allow list/read actions from everywhere",
            "Effect": "Allow",
            "Principal": {"AWS": "111122223333"},
            "Action": [
                "payment-cryptography:List*","payment-cryptography:Get*"
            ],
            "Resource": "*",
        }
    ]
}
```

Enregistrement de votre VPC terminal

AWS CloudTrail enregistre toutes les opérations qui utilisent le VPC point de terminaison. Lorsqu'une demande adressée à AWS Payment Cryptography utilise un VPC point de terminaison, l'ID du VPC point de terminaison apparaît dans l'entrée du AWS CloudTrail journal qui enregistre la demande. Vous pouvez utiliser l'identifiant du point de terminaison pour vérifier l'utilisation de votre point de VPC terminaison de cryptographie des AWS paiements.

Pour vous protégerVPC, les demandes refusées par une politique de point de VPC terminaison, mais qui auraient autrement été autorisées, ne sont pas enregistrées dans AWS CloudTrail.

Par exemple, cet exemple d'entrée de journal enregistre une GenerateMacdemande utilisant le point de VPC terminaison. Le champ `vpcEndpointId` apparaît à la fin de l'entrée de journal.

```
{
      "eventVersion": "1.08",
      "userIdentity": {
          "principalId": "TESTXECZ5U9M4LGF2N6Y5:i-98761b8890c09a34a",
```

```
          "arn": "arn:aws:sts::111122223333:assumed-role/samplerole/
i-98761b8890c09a34a",
          "accountId": "111122223333",
          "accessKeyId": "TESTXECZ5U2ZULLHHMJG",
          "sessionContext": {
              "sessionIssuer": {
                  "type": "Role",
                  "principalId": "TESTXECZ5U9M4LGF2N6Y5",
                  "arn": "arn:aws:iam::111122223333:role/samplerole",
                  "accountId": "111122223333",
                  "userName": "samplerole"
              },
              "webIdFederationData": {},
              "attributes": {
                  "creationDate": "2024-05-27T19:34:10Z",
                  "mfaAuthenticated": "false"
              },
              "ec2RoleDelivery": "2.0"
          }
      },
      "eventTime": "2024-05-27T19:49:54Z",
      "eventSource": "payment-cryptography.amazonaws.com",
      "eventName": "CreateKey",
      "awsRegion": "us-east-1",
      "sourceIPAddress": "172.31.85.253",
      "userAgent": "aws-cli/2.14.5 Python/3.9.16 Linux/6.1.79-99.167.amzn2023.x86_64
 source/x86_64.amzn.2023 prompt/off command/payment-cryptography.create-key",
      "requestParameters": {
          "keyAttributes": {
              "keyUsage": "TR31_M1_ISO_9797_1_MAC_KEY",
              "keyClass": "SYMMETRIC_KEY",
              "keyAlgorithm": "TDES_2KEY",
              "keyModesOfUse": {
                  "encrypt": false,
                  "decrypt": false,
                  "wrap": false,
                  "unwrap": false,
                  "generate": true,
                  "sign": false,
                  "verify": true,
                  "deriveKey": false,
                  "noRestrictions": false
              }
          },
```

```
            "exportable": true
        },
        "responseElements": {
            "key": {
                "keyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaifllw2h",
                "keyAttributes": {
                    "keyUsage": "TR31_M1_ISO_9797_1_MAC_KEY",
                    "keyClass": "SYMMETRIC_KEY",
                    "keyAlgorithm": "TDES_2KEY",
                    "keyModesOfUse": {
                        "encrypt": false,
                        "decrypt": false,
                        "wrap": false,
                        "unwrap": false,
                        "generate": true,
                        "sign": false,
                        "verify": true,
                        "deriveKey": false,
                        "noRestrictions": false
                    }
                },
                "keyCheckValue": "A486ED",
                "keyCheckValueAlgorithm": "ANSI_X9_24",
                "enabled": true,
                "exportable": true,
                "keyState": "CREATE_COMPLETE",
                "keyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
                "createTimestamp": "May 27, 2024, 7:49:54 PM",
                "usageStartTimestamp": "May 27, 2024, 7:49:54 PM"
            }
        },
        "requestID": "f3020b3c-4e86-47f5-808f-14c7a4a99161",
        "eventID": "b87c3d30-f3ab-4131-87e8-bc54cfef9d29",
        "readOnly": false,
        "eventType": "AwsApiCall",
        "managementEvent": true,
        "recipientAccountId": "111122223333",
        "vpcEndpointId": "vpce-1234abcdf5678c90a",
        "eventCategory": "Management",
        "tlsDetails": {
            "tlsVersion": "TLSv1.3",
            "cipherSuite": "TLS_AES_128_GCM_SHA256",
```

```
          "clientProvidedHostHeader": "vpce-1234abcdf5678c90a-
oo28vrvr.controlplane.payment-cryptography.us-east-1.vpce.amazonaws.com"
      }
  }
```

Bonnes pratiques de sécurité pour la cryptographie des AWS paiements

AWS La cryptographie des paiements prend en charge de nombreuses fonctionnalités de sécurité intégrées ou que vous pouvez éventuellement implémenter pour améliorer la protection de vos clés de chiffrement et garantir qu'elles sont utilisées conformément à leur destination, notamment des IAMpolitiques, un ensemble complet de clés de conditions de politique pour affiner vos politiques et IAM politiques clés et l'application intégrée des PCI PIN règles concernant les blocs de clés.

> **Important**
>
> Les directives générales fournies ne constituent pas une solution de sécurité complète. Étant donné que toutes les bonnes pratiques ne conviennent pas à toutes les situations, elles ne sont pas censées être prescriptives.

- Utilisation des clés et modes d'utilisation : La cryptographie des AWS paiements suit et applique les restrictions relatives à l'utilisation des clés et au mode d'utilisation, telles que décrites dans la spécification des blocs de clés d'échange de clés sécurisés interopérables ANSI X9 TR 31-2018 et conformément à l'exigence de sécurité 18-3. PCI PIN Cela limite la possibilité d'utiliser une seule clé à des fins multiples et lie cryptographiquement les métadonnées de la clé (telles que les opérations autorisées) au contenu de la clé lui-même. AWS La cryptographie des paiements applique automatiquement ces restrictions, de sorte qu'une clé de chiffrement (TR31_K0_ _ KEY ENCRYPTION _KEY) ne peut pas également être utilisée pour le déchiffrement des données. Pour plus d'informations, consultez Comprendre les principaux attributs de la clé AWS de cryptographie des paiements.
- Limitez le partage des clés symétriques : ne partagez que les clés symétriques (telles que les clés de chiffrement par code PIN ou les clés de chiffrement par clé) avec au plus une autre entité. S'il est nécessaire de transmettre des informations sensibles à un plus grand nombre d'entités ou de partenaires, créez des clés supplémentaires. AWS La cryptographie des paiements n'expose jamais le contenu d'une clé symétrique ou d'une clé privée asymétrique en clair.

- Utilisez des alias ou des tags pour associer des clés à certains cas d'utilisation ou à certains partenaires : les alias peuvent être utilisés pour indiquer facilement le cas d'utilisation associé à une clé, par exemple alias/ BIN _12345_ CVK pour désigner une clé de vérification de carte associée à 12345. BIN Pour plus de flexibilité, pensez à créer des balises telles que bin=12345, use_case=acquire, country=us, partner=foo. Les alias et les tags peuvent également être utilisés pour limiter l'accès, par exemple en renforçant les contrôles d'accès entre l'émission et l'acquisition des cas d'utilisation.
- Pratiquez l'accès le moins privilégié : IAM peut être utilisé pour limiter l'accès à la production aux systèmes plutôt qu'aux individus, par exemple en interdisant aux utilisateurs individuels de créer des clés ou d'exécuter des opérations cryptographiques. IAMpeut également être utilisé pour limiter l'accès aux commandes et aux touches qui peuvent ne pas être applicables à votre cas d'utilisation, par exemple pour limiter la capacité de générer ou de valider des épingles pour un acquéreur. Une autre façon d'utiliser l'accès le moins privilégié consiste à limiter les opérations sensibles (telles que l'importation de clés) à des comptes de service spécifiques. Pour obtenir des exemples, consultez AWS Exemples de politiques basées sur l'identité en matière de cryptographie des paiements.

Voir aussi

- Gestion des identités et des accès pour la cryptographie des AWS paiements
- Bonnes pratiques en matière de sécurité IAM décrites dans le guide de IAM l'utilisateur

Validation de conformité pour la cryptographie des AWS paiements

Des auditeurs tiers évaluent la sécurité et la conformité de la cryptographie des AWS paiements dans le cadre de multiples programmes de AWS conformité. Il s'agit notamment du SOC, du PCI et d'autres.

AWS La cryptographie des paiements a été évaluée pour plusieurs normes PCI en plus de la norme PCI DSS. Il s'agit notamment de la sécurité par code PIN PCI (code PIN PCI) et du chiffrement point à point (P2PE). Consultez les AWS Artifact attestations et les guides de conformité disponibles.

Pour obtenir la liste des AWS services concernés par des programmes de conformité spécifiques, voir Services AWS concernés par programme de conformité . Pour obtenir des renseignements généraux, consultez Programmes de conformitéAWS .

Vous pouvez télécharger des rapports d'audit tiers à l'aide de AWS Artifact. Pour plus d'informations, voir Téléchargement de rapports dans AWS Artifact .

Lorsque vous utilisez le chiffrement des AWS paiements, votre responsabilité en matière de conformité dépend de la sensibilité de vos données, des objectifs de conformité de votre entreprise et des lois et réglementations applicables. AWS fournit les ressources suivantes pour faciliter la mise en conformité :

- Guides de démarrage rapide sur la sécurité et la conformité Guides sur la sécurité et la conformité : ces guides de déploiement abordent les considérations architecturales et indiquent les étapes à suivre pour déployer des environnements de base axés sur la sécurité et la conformité sur. AWS
- AWS Ressources de conformité Ressources de : cette collection de classeurs et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- Évaluation des ressources à l'aide des règles énoncées dans le guide du AWS Config développeur :AWS Configévalue dans quelle mesure les configurations de vos ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- AWS Security Hub—Ce AWS service fournit une vue complète de l'état de votre sécurité interne, AWS ce qui vous permet de vérifier votre conformité aux normes et aux meilleures pratiques du secteur de la sécurité.

Gestion des identités et des accès pour la cryptographie des AWS paiements

AWS Identity and Access Management (IAM) est un outil Service AWS qui permet à un administrateur de contrôler en toute sécurité l'accès aux AWS ressources. IAMles administrateurs contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources de cryptographie des AWS paiements. IAMest un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

Rubriques

Public ciblé

La façon dont vous utilisez AWS Identity and Access Management (IAM) varie en fonction du travail que vous effectuez dans le domaine de la cryptographie des AWS paiements.

Utilisateur du service : si vous utilisez le service de cryptographie des AWS paiements pour effectuer votre travail, votre administrateur vous fournit les informations d'identification et les autorisations dont vous avez besoin. Au fur et à mesure que vous utilisez AWS de plus en plus de fonctionnalités de cryptographie des paiements dans le cadre de votre travail, vous aurez peut-être besoin d'autorisations supplémentaires. En comprenant bien la gestion des accès, vous saurez demander les autorisations appropriées à votre administrateur. Si vous ne parvenez pas à accéder à une fonctionnalité de la cryptographie des AWS paiements, consultezRésolution des problèmes d'identité et d'accès liés à la cryptographie des AWS paiements.

Administrateur du service — Si vous êtes responsable des ressources de cryptographie des AWS paiements au sein de votre entreprise, vous avez probablement un accès complet à la cryptographie des AWS paiements. C'est à vous de déterminer les fonctionnalités et les ressources AWS de cryptographie des paiements auxquelles les utilisateurs de vos services doivent accéder. Vous

devez ensuite envoyer des demandes à votre IAM administrateur pour modifier les autorisations des utilisateurs de votre service. Consultez les informations de cette page pour comprendre les concepts de base deIAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser IAM la cryptographie des AWS paiements, consultezComment fonctionne la cryptographie des AWS paiements avec IAM.

IAMadministrateur — Si vous êtes IAM administrateur, vous souhaiterez peut-être en savoir plus sur la manière dont vous pouvez rédiger des politiques pour gérer l'accès à la cryptographie des AWS paiements. Pour consulter des exemples AWS de politiques basées sur l'identité liées à la cryptographie des paiements que vous pouvez utiliser, consultez. IAM AWS Exemples de politiques basées sur l'identité en matière de cryptographie des paiements

Authentification par des identités

L'authentification est la façon dont vous vous connectez à AWS l'aide de vos informations d'identification. Vous devez être authentifié (connecté à AWS) en tant que Utilisateur racine d'un compte AWS, en tant qu'IAMutilisateur ou en assumant un IAM rôle.

Vous pouvez vous connecter en AWS tant qu'identité fédérée en utilisant les informations d'identification fournies par le biais d'une source d'identité. AWS IAM Identity Center Les utilisateurs (IAMIdentity Center), l'authentification unique de votre entreprise et vos informations d'identification Google ou Facebook sont des exemples d'identités fédérées. Lorsque vous vous connectez en tant qu'identité fédérée, votre administrateur a préalablement configuré la fédération d'identité à l'aide de IAM rôles. Lorsque vous accédez à AWS l'aide de la fédération, vous assumez indirectement un rôle.

Selon le type d'utilisateur que vous êtes, vous pouvez vous connecter au portail AWS Management Console ou au portail AWS d'accès. Pour plus d'informations sur la connexion à AWS, consultez la section Comment vous connecter à votre compte Compte AWS dans le guide de Connexion à AWS l'utilisateur.

Si vous y accédez AWS par programmation, AWS fournit un kit de développement logiciel (SDK) et une interface de ligne de commande (CLI) pour signer cryptographiquement vos demandes à l'aide de vos informations d'identification. Si vous n'utilisez pas d' AWS outils, vous devez signer vous-même les demandes. Pour plus d'informations sur l'utilisation de la méthode recommandée pour signer vous-même les demandes, consultez la section Signature des AWS API demandes dans le guide de IAM l'utilisateur.

Quelle que soit la méthode d'authentification que vous utilisez, vous devrez peut-être fournir des informations de sécurité supplémentaires. Par exemple, il vous AWS recommande d'utiliser

l'authentification multifactorielle (MFA) pour renforcer la sécurité de votre compte. Pour en savoir plus, consultez Authentification multifactorielle dans le guide de AWS IAM Identity Center l'utilisateur et Utilisation de l'authentification multifactorielle (MFA) AWS dans le guide de l'IAMutilisateur.

Compte AWS utilisateur root

Lorsque vous créez un Compte AWS, vous commencez par une identité de connexion unique qui donne un accès complet à toutes Services AWS les ressources du compte. Cette identité est appelée utilisateur Compte AWS root et est accessible en vous connectant avec l'adresse e-mail et le mot de passe que vous avez utilisés pour créer le compte. Il est vivement recommandé de ne pas utiliser l'utilisateur racine pour vos tâches quotidiennes. Protégez vos informations d'identification d'utilisateur racine et utilisez-les pour effectuer les tâches que seul l'utilisateur racine peut effectuer. Pour obtenir la liste complète des tâches qui nécessitent que vous vous connectiez en tant qu'utilisateur root, consultez la section Tâches nécessitant des informations d'identification utilisateur root dans le guide de IAM l'utilisateur.

Utilisateurs et groupes IAM

Un IAMutilisateur est une identité au sein de vous Compte AWS qui possède des autorisations spécifiques pour une seule personne ou une seule application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des IAM utilisateurs dotés d'informations d'identification à long terme, telles que des mots de passe et des clés d'accès. Toutefois, si vous avez des cas d'utilisation spécifiques qui nécessitent des informations d'identification à long terme auprès des IAM utilisateurs, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, voir Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification à long terme dans le Guide de IAM l'utilisateur.

Un IAMgroupe est une identité qui définit un ensemble d'IAMutilisateurs. Vous ne pouvez pas vous connecter en tant que groupe. Vous pouvez utiliser les groupes pour spécifier des autorisations pour plusieurs utilisateurs à la fois. Les groupes permettent de gérer plus facilement les autorisations pour de grands ensembles d'utilisateurs. Par exemple, vous pouvez nommer un groupe IAMAdminset lui donner les autorisations nécessaires pour administrer IAM des ressources.

Les utilisateurs sont différents des rôles. Un utilisateur est associé de manière unique à une personne ou une application, alors qu'un rôle est conçu pour être endossé par tout utilisateur qui en a besoin. Les utilisateurs disposent d'informations d'identification permanentes, mais les rôles fournissent des informations d'identification temporaires. Pour en savoir plus, voir Quand créer un IAM utilisateur (au lieu d'un rôle) dans le Guide de IAM l'utilisateur.

IAMrôles

Un IAMrôle est une identité au sein de Compte AWS vous dotée d'autorisations spécifiques. Il est similaire à un IAM utilisateur, mais n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un IAM rôle dans le en AWS Management Console changeant de rôle. Vous pouvez assumer un rôle en appelant une AWS API opération AWS CLI or ou en utilisant une option personnaliséeURL. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez la section Utilisation IAM des rôles dans le Guide de IAM l'utilisateur.

IAMles rôles dotés d'informations d'identification temporaires sont utiles dans les situations suivantes :

- Accès utilisateur fédéré – Pour attribuer des autorisations à une identité fédérée, vous créez un rôle et définissez des autorisations pour le rôle. Quand une identité externe s'authentifie, l'identité est associée au rôle et reçoit les autorisations qui sont définies par celui-ci. Pour plus d'informations sur les rôles pour la fédération, voir Création d'un rôle pour un fournisseur d'identité tiers dans le guide de IAM l'utilisateur. Si vous utilisez IAM Identity Center, vous configurez un ensemble d'autorisations. Pour contrôler les accès auxquels vos identités peuvent accéder après leur authentification, IAM Identity Center met en corrélation l'ensemble d'autorisations avec un rôle dans. IAM Pour plus d'informations sur les jeux d'autorisations, consultez la rubrique Jeux d'autorisations dans le Guide de l'utilisateur AWS IAM Identity Center .
- Autorisations IAM utilisateur temporaires : un IAM utilisateur ou un rôle peut assumer un IAM rôle afin d'obtenir temporairement différentes autorisations pour une tâche spécifique.
- Accès entre comptes : vous pouvez utiliser un IAM rôle pour autoriser une personne (un mandant fiable) d'un autre compte à accéder aux ressources de votre compte. Les rôles constituent le principal moyen d'accorder l'accès intercompte. Toutefois, dans certains Services AWS cas, vous pouvez associer une politique directement à une ressource (au lieu d'utiliser un rôle comme proxy). Pour connaître la différence entre les rôles et les politiques basées sur les ressources pour l'accès entre comptes, voir Accès aux ressources entre comptes IAM dans le guide de l'IAMutilisateur.
- Accès multiservices — Certains Services AWS utilisent des fonctionnalités dans d'autres Services AWS. Par exemple, lorsque vous effectuez un appel dans un service, il est courant que ce service exécute des applications dans Amazon EC2 ou stocke des objets dans Amazon S3. Un service peut le faire en utilisant les autorisations d'appel du principal, un rôle de service ou un rôle lié au service.
  - Sessions d'accès transmises (FAS) — Lorsque vous utilisez un IAM utilisateur ou un rôle pour effectuer des actions AWS, vous êtes considéré comme un mandant. Lorsque vous utilisez certains services, vous pouvez effectuer une action qui initie une autre action dans un autre

service. FASutilise les autorisations du principal appelant an Service AWS, combinées à la demande Service AWS pour adresser des demandes aux services en aval. FASles demandes ne sont effectuées que lorsqu'un service reçoit une demande qui nécessite des interactions avec d'autres personnes Services AWS ou des ressources pour être traitée. Dans ce cas, vous devez disposer d'autorisations nécessaires pour effectuer les deux actions. Pour plus de détails sur les politiques relatives FAS aux demandes, consultez la section Transférer les sessions d'accès.

- Rôle de service — Un rôle de service est un IAMrôle qu'un service assume pour effectuer des actions en votre nom. Un IAM administrateur peut créer, modifier et supprimer un rôle de service de l'intérieurIAM. Pour plus d'informations, consultez la section Création d'un rôle auquel déléguer des autorisations Service AWS dans le Guide de IAM l'utilisateur.
- Rôle lié à un service — Un rôle lié à un service est un type de rôle de service lié à un. Service AWS Le service peut endosser le rôle afin d'effectuer une action en votre nom. Les rôles liés à un service apparaissent dans votre Compte AWS répertoire et appartiennent au service. Un IAM administrateur peut consulter, mais pas modifier les autorisations pour les rôles liés à un service.

- Applications exécutées sur Amazon EC2 : vous pouvez utiliser un IAM rôle pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une EC2 instance et qui font AWS CLI des AWS API demandes. Cela est préférable au stockage des clés d'accès dans l'EC2instance. Pour attribuer un AWS rôle à une EC2 instance et le rendre disponible pour toutes ses applications, vous devez créer un profil d'instance attaché à l'instance. Un profil d'instance contient le rôle et permet aux programmes exécutés sur l'EC2instance d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez la section Utilisation d'un IAM rôle pour accorder des autorisations aux applications exécutées sur des EC2 instances Amazon dans le Guide de IAM l'utilisateur.

Pour savoir s'il faut utiliser IAM des rôles ou des IAM utilisateurs, voir Quand créer un IAM rôle (au lieu d'un utilisateur) dans le guide de IAM l'utilisateur.

Gestion des accès à l'aide de politiques

Vous contrôlez l'accès en AWS créant des politiques et en les associant à AWS des identités ou à des ressources. Une politique est un objet AWS qui, lorsqu'il est associé à une identité ou à une ressource, définit leurs autorisations. AWS évalue ces politiques lorsqu'un principal (utilisateur, utilisateur root ou session de rôle) fait une demande. Les autorisations dans les politiques déterminent si la demande est autorisée ou refusée. La plupart des politiques sont stockées AWS sous forme de JSON documents. Pour plus d'informations sur la structure et le contenu

des documents de JSON politique, voir Présentation des JSON politiques dans le guide de IAM l'utilisateur.

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

Par défaut, les utilisateurs et les rôles ne disposent d'aucune autorisation. Pour autoriser les utilisateurs à effectuer des actions sur les ressources dont ils ont besoin, un IAM administrateur peut créer des IAM politiques. L'administrateur peut ensuite ajouter les IAM politiques aux rôles, et les utilisateurs peuvent assumer les rôles.

IAMles politiques définissent les autorisations pour une action, quelle que soit la méthode que vous utilisez pour effectuer l'opération. Par exemple, supposons que vous disposiez d'une politique qui autorise l'action `iam:GetRole`. Un utilisateur appliquant cette politique peut obtenir des informations sur le rôle auprès du AWS Management Console AWS CLI, ou du AWS API.

Politiques basées sur l'identité

Les politiques basées sur l'identité sont JSON des documents de politique d'autorisation que vous pouvez joindre à une identité, telle qu'un IAM utilisateur, un groupe d'utilisateurs ou un rôle. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour savoir comment créer une politique basée sur l'identité, consultez la section Création de IAM politiques dans le Guide de l'IAMutilisateur.

Les politiques basées sur l'identité peuvent être classées comme des politiques en ligne ou des politiques gérées. Les politiques en ligne sont intégrées directement à un utilisateur, groupe ou rôle. Les politiques gérées sont des politiques autonomes que vous pouvez associer à plusieurs utilisateurs, groupes et rôles au sein de votre Compte AWS. Les politiques gérées incluent les politiques AWS gérées et les politiques gérées par le client. Pour savoir comment choisir entre une politique gérée ou une politique intégrée, voir Choisir entre des politiques gérées et des politiques intégrées dans le Guide de l'IAMutilisateur.

Politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents JSON de stratégie que vous attachez à une ressource. Les politiques de confiance dans les IAM rôles et les politiques relatives aux compartiments Amazon S3 sont des exemples de politiques basées sur les ressources. Dans les services qui sont compatibles avec les politiques basées sur les ressources, les administrateurs de service peuvent les utiliser pour contrôler l'accès à une ressource spécifique. Pour la ressource dans

laquelle se trouve la politique, cette dernière définit quel type d'actions un principal spécifié peut effectuer sur cette ressource et dans quelles conditions. Vous devez spécifier un principal dans une politique basée sur les ressources. Les principaux peuvent inclure des comptes, des utilisateurs, des rôles, des utilisateurs fédérés ou. Services AWS

Les politiques basées sur les ressources sont des politiques en ligne situées dans ce service. Vous ne pouvez pas utiliser de politiques AWS gérées depuis une IAM stratégie basée sur les ressources.

Listes de contrôle d'accès (ACLs)

Les listes de contrôle d'accès (ACLs) contrôlent les principaux (membres du compte, utilisateurs ou rôles) autorisés à accéder à une ressource. ACLssont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format du document JSON de stratégie.

Amazon S3 et Amazon VPC sont des exemples de services compatiblesACLs. AWS WAF Pour en savoir plusACLs, consultez la présentation de la liste de contrôle d'accès (ACL) dans le guide du développeur Amazon Simple Storage Service.

Autres types de politique

AWS prend en charge d'autres types de politiques moins courants. Ces types de politiques peuvent définir le nombre maximum d'autorisations qui vous sont accordées par des types de politiques plus courants.

- Limites d'autorisations — Une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le maximum d'autorisations qu'une politique basée sur l'identité peut accorder à une IAM entité (IAMutilisateur ou rôle). Vous pouvez définir une limite d'autorisations pour une entité. Les autorisations en résultant représentent la combinaison des politiques basées sur l'identité d'une entité et de ses limites d'autorisation. Les politiques basées sur les ressources qui spécifient l'utilisateur ou le rôle dans le champ `Principal` ne sont pas limitées par les limites d'autorisations. Un refus explicite dans l'une de ces politiques remplace l'autorisation. Pour plus d'informations sur les limites d'autorisations, voir Limites d'autorisations pour les IAM entités dans le Guide de IAM l'utilisateur.
- Politiques de contrôle des services (SCPs) : SCPs JSON politiques qui spécifient les autorisations maximales pour une organisation ou une unité organisationnelle (UO) dans AWS Organizations. AWS Organizations est un service permettant de regrouper et de gérer de manière centralisée Comptes AWS les multiples propriétés de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer des politiques de contrôle des services (SCPs) à l'un ou à l'ensemble de vos comptes. Les SCP limites d'autorisations pour les entités

présentes dans les comptes des membres, y compris chacune d'entre elles Utilisateur racine d'un compte AWS. Pour plus d'informations sur les OrganizationsSCPs, voir Politiques de contrôle des services dans le Guide de AWS Organizations l'utilisateur.

- Politiques de séance : les politiques de séance sont des politiques avancées que vous utilisez en tant que paramètre lorsque vous créez par programmation une séance temporaire pour un rôle ou un utilisateur fédéré. Les autorisations de séance en résultant sont une combinaison des politiques basées sur l'identité de l'utilisateur ou du rôle et des politiques de séance. Les autorisations peuvent également provenir d'une politique basée sur les ressources. Un refus explicite dans l'une de ces politiques annule l'autorisation. Pour plus d'informations, consultez la section Politiques de session dans le guide de IAM l'utilisateur.

Plusieurs types de politique

Lorsque plusieurs types de politiques s'appliquent à la requête, les autorisations en résultant sont plus compliquées à comprendre. Pour savoir comment AWS déterminer s'il faut autoriser une demande lorsque plusieurs types de politiques sont impliqués, consultez la section Logique d'évaluation des politiques dans le guide de IAM l'utilisateur.

Comment fonctionne la cryptographie des AWS paiements avec IAM

Avant de gérer l'accès IAM à la cryptographie des AWS paiements, vous devez comprendre quelles IAM fonctionnalités peuvent être utilisées avec la cryptographie des AWS paiements. Pour obtenir une vue d'ensemble du fonctionnement de la cryptographie des AWS paiements et AWS des autres servicesIAM, consultez la section AWS Services compatibles IAM dans le guide de l'IAMutilisateur.

Rubriques

- AWS Cryptographie des paiements Politiques basées sur l'identité
- Autorisation basée sur les balises AWS de cryptographie des paiements

AWS Cryptographie des paiements Politiques basées sur l'identité

Avec les politiques IAM basées sur l'identité, vous pouvez spécifier les actions et les ressources autorisées ou refusées ainsi que les conditions dans lesquelles les actions sont autorisées ou refusées. AWS La cryptographie des paiements prend en charge des actions, des ressources et des

clés de condition spécifiques. Pour en savoir plus sur tous les éléments que vous utilisez dans une JSON politique, consultez la section Référence des éléments de IAM JSON stratégie dans le guide de IAM l'utilisateur.

Actions

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'`Action`élément d'une JSON politique décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès dans une politique. Les actions de stratégie portent généralement le même nom que l' AWS APIopération associée. Il existe certaines exceptions, telles que les actions avec autorisation uniquement qui n'ont pas d'opération correspondante. API Certaines opérations nécessitent également plusieurs actions dans une politique. Ces actions supplémentaires sont nommées actions dépendantes.

Intégration d'actions dans une stratégie afin d'accorder l'autorisation d'exécuter les opérations associées.

Les actions politiques dans AWS Payment Cryptography utilisent le préfixe suivant avant l'action :. `payment-cryptography:` Par exemple, pour autoriser une personne à exécuter une `VerifyCardData` API opération de cryptographie des AWS paiements, vous devez inclure cette `payment-cryptography:VerifyCardData` action dans sa politique. Les déclarations de politique doivent inclure un élément `Action` ou `NotAction`. AWS La cryptographie des paiements définit son propre ensemble d'actions décrivant les tâches que vous pouvez effectuer avec ce service.

Pour spécifier plusieurs actions dans une seule déclaration, séparez-les par des virgules comme suit :

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

Vous pouvez aussi spécifier plusieurs actions à l'aide de caractères génériques (*). Par exemple, pour spécifier toutes les actions commençant par le mot `List` (telles que `ListKeys` et`ListAliases`), incluez l'action suivante :

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

Pour consulter la liste des actions de cryptographie des AWS paiements, consultez la section Actions définies par la cryptographie des AWS paiements dans le guide de l'IAMutilisateur.

Ressources

Les administrateurs peuvent utiliser AWS JSON des politiques pour spécifier qui a accès à quoi. C'est-à-dire, quel principal peut effectuer des actions sur quelles ressources et dans quelles conditions.

L'élément `Resource` JSON de stratégie indique le ou les objets auxquels s'applique l'action. Les instructions doivent inclure un élément `Resource` ou `NotResource`. Il est recommandé de spécifier une ressource en utilisant son Amazon Resource Name (ARN). Vous pouvez le faire pour des actions qui prennent en charge un type de ressource spécifique, connu sous la dénomination autorisations de niveau ressource.

Pour les actions qui ne sont pas compatibles avec les autorisations de niveau ressource, telles que les opérations de liste, utilisez un caractère générique (*) afin d'indiquer que l'instruction s'applique à toutes les ressources.

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

La ressource clé de cryptographie pour les paiements comprend les éléments suivants : ARN

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

Pour plus d'informations sur le format deARNs, consultez Amazon Resource Names (ARNs) et AWS Service Namespaces.

Par exemple, pour spécifier l'`arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h`instance dans votre relevé, utilisez ce qui suit ARN :

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

Pour spécifier toutes les clés appartenant à un compte spécifique, utilisez le caractère générique (*) :

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

Certaines actions AWS de chiffrement des paiements, telles que celles relatives à la création de clés, ne peuvent pas être effectuées sur une ressource spécifique. Dans ces cas-là, vous devez utiliser le caractère générique (*).

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

Pour spécifier plusieurs ressources dans une seule instruction, utilisez une virgule comme indiqué ci-dessous :

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

Exemples

Pour consulter des exemples de politiques basées sur l'identité en matière de cryptographie des AWS paiements, consultez. AWS Exemples de politiques basées sur l'identité en matière de cryptographie des paiements

Autorisation basée sur les balises AWS de cryptographie des paiements

AWS Exemples de politiques basées sur l'identité en matière de cryptographie des paiements

Par défaut, IAM les utilisateurs et les rôles ne sont pas autorisés à créer ou à modifier des ressources AWS de chiffrement des paiements. Ils ne peuvent pas non plus effectuer de tâches à l'aide du AWS Management Console AWS CLI, ou AWS API. Un IAM administrateur doit créer des IAM politiques qui accordent aux utilisateurs et aux rôles l'autorisation d'effectuer des API opérations spécifiques sur les ressources spécifiques dont ils ont besoin. L'administrateur doit ensuite associer ces politiques aux IAM utilisateurs ou aux groupes qui ont besoin de ces autorisations.

Pour savoir comment créer une politique IAM basée sur l'identité à l'aide de ces exemples de documents de JSON stratégie, voir Création de politiques dans l'JSONonglet du guide de l'IAMutilisateur.

Rubriques

- Bonnes pratiques en matière de politiques

- Utilisation de la console AWS de chiffrement des paiements
- Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations
- Possibilité d'accéder à tous les aspects de la cryptographie des AWS paiements
- Possibilité d'appeler à APIs l'aide des touches spécifiées
- Possibilité de refuser spécifiquement une ressource

Bonnes pratiques en matière de politiques

Les politiques basées sur l'identité déterminent si quelqu'un peut créer, accéder ou supprimer des ressources de chiffrement des AWS paiements dans votre compte. Ces actions peuvent entraîner des frais pour votre Compte AWS. Lorsque vous créez ou modifiez des politiques basées sur l'identité, suivez ces instructions et recommandations :

- Commencez AWS par les politiques gérées et passez aux autorisations du moindre privilège : pour commencer à accorder des autorisations à vos utilisateurs et à vos charges de travail, utilisez les politiques AWS gérées qui accordent des autorisations pour de nombreux cas d'utilisation courants. Ils sont disponibles dans votre Compte AWS. Nous vous recommandons de réduire davantage les autorisations en définissant des politiques gérées par les AWS clients spécifiques à vos cas d'utilisation. Pour plus d'informations, consultez les politiques AWS gérées ou les politiques AWS gérées pour les fonctions professionnelles dans le Guide de IAM l'utilisateur.
- Appliquer les autorisations du moindre privilège : lorsque vous définissez des autorisations à IAM l'aide de politiques, accordez uniquement les autorisations nécessaires à l'exécution d'une tâche. Pour ce faire, vous définissez les actions qui peuvent être entreprises sur des ressources spécifiques dans des conditions spécifiques, également appelées autorisations de moindre privilège. Pour plus d'informations sur l'utilisation IAM pour appliquer des autorisations, consultez la section Politiques et autorisations du Guide de IAM l'utilisateur. IAM
- Utilisez des conditions dans IAM les politiques pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques pour limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez rédiger une condition de politique pour spécifier que toutes les demandes doivent être envoyées en utilisantSSL. Vous pouvez également utiliser des conditions pour accorder l'accès aux actions de service si elles sont utilisées par le biais d'un service spécifique Service AWS, tel que AWS CloudFormation. Pour plus d'informations, voir Éléments IAM JSON de politique : Condition dans le guide de IAM l'utilisateur.
- Utilisez IAM Access Analyzer pour valider vos IAM politiques afin de garantir des autorisations sécurisées et fonctionnelles. IAM Access Analyzer valide les politiques nouvelles et existantes afin

qu'elles soient conformes au langage des IAM politiques (JSON) et IAM aux meilleures pratiques. IAMAccess Analyzer fournit plus de 100 vérifications des politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez la section Validation des politiques d'IAMAccess Analyzer dans le guide de IAM l'utilisateur.

- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des IAM utilisateurs ou un utilisateur root Compte AWS, activez-le MFA pour une sécurité supplémentaire. Pour exiger le MFA moment où les API opérations sont appelées, ajoutez MFA des conditions à vos politiques. Pour plus d'informations, consultez la section Configuration de l'APIaccès MFA protégé dans le Guide de l'IAMutilisateur.

Pour plus d'informations sur les meilleures pratiques en matière de sécuritéIAM, consultez la section Bonnes pratiques en matière de sécurité IAM dans le Guide de IAM l'utilisateur.

Utilisation de la console AWS de chiffrement des paiements

Pour accéder à la console AWS Payment Cryptography, vous devez disposer d'un minimum d'autorisations. Ces autorisations doivent vous permettre de répertorier et de consulter les informations relatives aux ressources AWS de chiffrement des paiements de votre AWS compte. Si vous créez une politique basée sur l'identité qui est plus restrictive que les autorisations minimales requises, la console ne fonctionnera pas comme prévu pour les entités (IAMutilisateurs ou rôles) dotées de cette politique.

Pour garantir que ces entités peuvent toujours utiliser la console AWS Payment Cryptography, associez également la politique AWS gérée suivante aux entités. Pour plus d'informations, consultez la section Ajouter des autorisations à un utilisateur dans le guide de IAM l'utilisateur.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement le AWS CLI ou le AWS API. Au lieu de cela, autorisez uniquement l'accès aux actions correspondant à l'APIopération que vous essayez d'effectuer.

Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux IAM utilisateurs de consulter les politiques intégrées et gérées associées à leur identité d'utilisateur. Cette politique inclut les autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide du AWS CLI ou. AWS API

```
{
    "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": "*"
        }
    ]
}
```

Possibilité d'accéder à tous les aspects de la cryptographie des AWS paiements

> **Warning**
>
> Cet exemple fournit des autorisations étendues et n'est pas recommandé. Envisagez plutôt les modèles d'accès les moins privilégiés.

Dans cet exemple, vous souhaitez accorder à un IAM utilisateur de votre AWS compte l'accès à toutes vos clés de chiffrement des AWS paiements et la possibilité d'appeler toutes les API de cryptographie des AWS paiements, y compris ControlPlane les DataPlane deux opérations.

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

Possibilité d'appeler à APIs l'aide des touches spécifiées

Dans cet exemple, vous souhaitez autoriser un IAM utilisateur de votre AWS compte à accéder à l'une de vos clés de chiffrement des AWS paiements, `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaifllw2h` puis utiliser cette ressource en deuxAPIs, `GenerateCardData` et`VerifyCardData`. À l'inverse, IAM l'utilisateur n'aura pas accès à cette clé pour d'autres opérations telles que `DeleteKey` ou `ExportKey`

Les ressources peuvent être soit des clés, préfixées par, `key` soit des alias, préfixés par. `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"
            ]
        }
    ]
}
```

Possibilité de refuser spécifiquement une ressource

> **Warning**
>
> Réfléchissez bien aux implications de l'octroi d'un accès générique. Envisagez plutôt un modèle de moindre privilège.

Dans cet exemple, vous souhaitez autoriser un IAM utilisateur de votre AWS compte à accéder à l'une de vos clés de chiffrement des AWS paiements, mais vous souhaitez refuser les autorisations relatives à une clé spécifique. L'utilisateur aura accès à `VerifyCardData` et `GenerateCardData` avec toutes les clés, à l'exception de celle spécifiée dans la déclaration de refus.

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

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

Résolution des problèmes d'identité et d'accès liés à la cryptographie des AWS paiements

Des sujets seront ajoutés à cette section IAM au fur et à mesure que les problèmes connexes spécifiques à la cryptographie des AWS paiements seront identifiés. Pour des informations générales sur le dépannage sur IAM différents sujets, reportez-vous à la section du Guide de l'IAMutilisateur consacrée à la résolution des problèmes.

Surveillance de la cryptographie des AWS paiements

La surveillance joue un rôle important dans le maintien de la fiabilité, de la disponibilité et des performances de AWS Payment Cryptography et de vos autres solutions AWS. AWS fournit les outils de surveillance suivants pour surveiller le chiffrement des AWS paiements, signaler tout problème et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille vos AWS ressources et les applications que vous utilisez AWS en temps réel. Vous pouvez collecter et suivre les métriques, créer des tableaux de bord personnalisés, et définir des alarmes qui vous informent ou prennent des mesures lorsqu'une métrique spécifique atteint un seuil que vous spécifiez. Par exemple, vous pouvez CloudWatch suivre l'utilisation de certaines API ou vous avertir si vous approchez de vos quotas de cryptographie des AWS paiements. Pour plus d'informations, consultez le guide de CloudWatch l'utilisateur Amazon.
- Amazon CloudWatch Logs vous permet de surveiller, de stocker et d'accéder à vos fichiers journaux à partir d'instances Amazon EC2 et d'autres sources. CloudTrail CloudWatch Les journaux peuvent surveiller les informations contenues dans les fichiers journaux et vous avertir lorsque certains seuils sont atteints. Vous pouvez également archiver vos données de journaux dans une solution de stockage hautement durable. Pour plus d'informations, consultez le guide de l'utilisateur d'Amazon CloudWatch Logs.
- AWS CloudTrailcapture les appels d'API et les événements associés effectués par ou pour le compte de votre AWS compte et envoie les fichiers journaux dans un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes appelés AWS, le point de terminaison appelé, les ressources (clés) utilisées, l'adresse IP source à partir de laquelle les appels ont été effectués et le moment où les appels ont eu lieu. Pour de plus amples informations, veuillez consulter le Guide de l'utilisateur AWS CloudTrail.

Rubriques

- Enregistrement des API appels AWS de cryptographie de paiement à l'aide de AWS CloudTrail

Enregistrement des API appels AWS de cryptographie de paiement à l'aide de AWS CloudTrail

AWS La cryptographie des paiements est intégrée à AWS CloudTrail un service qui fournit un enregistrement des actions entreprises par un utilisateur, un rôle ou un AWS service dans le domaine de la cryptographie des AWS paiements. CloudTrail capture tous les API appels à la cryptographie des AWS paiements sous forme d'événements. Les appels capturés incluent des appels provenant de la console et des appels de code vers les API opérations. Si vous créez un suivi, vous pouvez activer la diffusion continue d' CloudTrail événements vers un compartiment Amazon S3, y compris des événements pour le chiffrement des AWS paiements. Si vous ne configurez pas de suivi, vous pouvez toujours consulter les derniers événements de gestion (plan de contrôle) dans la CloudTrail console dans l'historique des événements. À l'aide des informations collectées par CloudTrail, vous pouvez déterminer la demande qui a été faite à AWS Payment Cryptography, l'adresse IP à partir de laquelle la demande a été faite, qui a fait la demande, quand elle a été faite et des informations supplémentaires.

Pour en savoir plus CloudTrail, consultez le guide de AWS CloudTrail l'utilisateur.

Rubriques

AWS Informations relatives à la cryptographie des paiements dans CloudTrail

CloudTrail est activé sur votre AWS compte lorsque vous le créez. Lorsqu'une activité se produit dans la cryptographie des AWS paiements, cette activité est enregistrée dans un CloudTrail événement avec d'autres événements de AWS service dans l'historique des événements. Vous pouvez consulter, rechercher et télécharger les événements récents dans votre AWS compte. Pour plus d'informations, consultez la section Affichage des événements à l'aide de l'historique des CloudTrail événements.

Pour un enregistrement continu des événements de votre AWS compte, y compris les événements liés à la cryptographie des AWS paiements, créez une trace. Un suivi permet CloudTrail de fournir

des fichiers journaux à un compartiment Amazon S3. Par défaut, lorsque vous créez un parcours dans la console, celui-ci s'applique à toutes les AWS régions. Le journal enregistre les événements de toutes les régions de la AWS partition et transmet les fichiers journaux au compartiment Amazon S3 que vous spécifiez. En outre, vous pouvez configurer d'autres AWS services pour analyser plus en détail les données d'événements collectées dans les CloudTrail journaux et agir en conséquence. Pour plus d'informations, consultez les ressources suivantes :

- Présentation de la création d'un journal de suivi
- CloudTrail services et intégrations pris en charge
- Configuration des SNS notifications Amazon pour CloudTrail
- Réception de fichiers CloudTrail journaux provenant de plusieurs régions
- Réception de fichiers CloudTrail journaux provenant de plusieurs comptes

Chaque événement ou entrée de journal contient des informations sur la personne ayant initié la demande. Les informations relatives à l'identité permettent de déterminer les éléments suivants :

- Si la demande a été faite avec les informations d'identification de l'utilisateur root ou AWS Identity and Access Management (IAM).
- Si la demande a été effectuée avec les informations d'identification de sécurité temporaires d'un rôle ou d'un utilisateur fédéré.
- Si la demande a été faite par un autre AWS service.

Pour plus d'informations, consultez l'CloudTrail userIdentity élément.

Événements du plan de contrôle dans CloudTrail

CloudTrail enregistre les opérations AWS de cryptographie des paiements, telles que CreateKey, ImportKey,, DeleteKey, ListKeysTagResource, et toutes les autres opérations du plan de contrôle.

Événements liés aux données dans CloudTrail

Les événements de données fournissent des informations sur les opérations de ressources effectuées sur ou dans une ressource, telles que le chiffrement d'une charge utile ou la traduction d'un code PIN. Les événements de données sont des activités volumineuses qui CloudTrail ne sont pas enregistrées par défaut. Vous pouvez activer la journalisation des API actions relatives aux événements de données pour les événements du plan de données AWS Payment Cryptography

à l'aide de notre console CloudTrail APIs. Pour plus d'informations, consultez Journalisation des événements de données dans le Guide de l'utilisateur AWS CloudTrail .

Avec CloudTrail, vous devez utiliser des sélecteurs d'événements avancés pour décider quelles API activités de cryptographie des AWS paiements sont enregistrées et enregistrées. Pour enregistrer les événements du plan de données de cryptographie des AWS paiements, vous devez inclure le type `AWS Payment Cryptography key` de ressource et. `AWS Payment Cryptography alias` Une fois cette configuration effectuée, vous pouvez peaufiner vos préférences de journalisation en spécifiant les événements de données à enregistrer, par exemple en utilisant le filtre `eventName` pour suivre les événements `EncryptData`. Pour plus d'informations, reportez-vous `AdvancedEventSelector`à la section AWS CloudTrail APIRéférence.

> **Note**
>
> Pour vous abonner aux événements relatifs aux données de cryptographie des AWS paiements, vous devez utiliser des sélecteurs d'événements avancés. Nous vous recommandons de vous abonner aux événements clés et aux alias pour être sûr de recevoir tous les événements.

événements liés aux données :

- `DecryptData`
- `EncryptData`
- `GenerateCardValidationData`
- `GenerateMac`
- `GeneratePinData`
- `ReEncryptData`
- `TranslatePinData`
- `VerifyAuthRequestCryptogram`
- `VerifyCardValidationData`
- `VerifyMac`
- `VerifyPinData`

Des frais supplémentaires s'appliquent pour les événements de données. Pour plus d'informations, consultez la section AWS CloudTrailTarification.

Comprendre les entrées AWS du fichier journal du plan de contrôle de la cryptographie des paiements

Un suivi est une configuration qui permet de transmettre des événements sous forme de fichiers journaux à un compartiment Amazon S3 que vous spécifiez. CloudTrail les fichiers journaux contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et inclut des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la demande, etc. CloudTrail les fichiers journaux ne constituent pas une trace ordonnée des API appels publics, ils n'apparaissent donc pas dans un ordre spécifique.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`CreateKey`action AWS de chiffrement des paiements.

```
      {
    CloudTrailEvent: {
      tlsDetails= {
        TlsDetails: {
          cipherSuite=TLS_AES_128_GCM_SHA256,
          tlsVersion=TLSv1.3,
          clientProvidedHostHeader=controlplane.paymentcryptography.us-
west-2.amazonaws.com
        }
      },
    requestParameters=CreateKeyInput (
      keyAttributes=KeyAttributes(
        KeyUsage=TR31_B0_BASE_DERIVATION_KEY,
        keyClass=SYMMETRIC_KEY,
        keyAlgorithm=AES_128,
        keyModesOfUse=KeyModesOfUse(
          encrypt=false,
          decrypt=false,
          wrap=false
          unwrap=false,
          generate=false,
          sign=false,
          verify=false,
```

```
          deriveKey=true,
          noRestrictions=false)
        ),
      keyCheckValueAlgorithm=null,
      exportable=true,
      enabled=true,
      tags=null),
    eventName=CreateKey,
    userAgent=Coral/Apache-HttpClient5,
    responseElements=CreateKeyOutput(
      key=Key(
        keyArn=arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwsp,
          keyAttributes=KeyAttributes(
            KeyUsage=TR31_B0_BASE_DERIVATION_KEY,
            keyClass=SYMMETRIC_KEY,
            keyAlgorithm=AES_128,
            keyModesOfUse=KeyModesOfUse(
              encrypt=false,
              decrypt=false,
              wrap=false,
              unwrap=false,
              generate=false,
              sign=false,
              verify=false,
              deriveKey=true,
              noRestrictions=false)
            ),
          keyCheckValue=FE23D3,
          keyCheckValueAlgorithm=ANSI_X9_24,
          enabled=true,
          exportable=true,
          keyState=CREATE_COMPLETE,
          keyOrigin=AWS_PAYMENT_CRYPTOGRAPHY,
          createTimestamp=Sun May 21 18:58:32 UTC 2023,
          usageStartTimestamp=Sun May 21 18:58:32 UTC 2023,
          usageStopTimestamp=null,
          deletePendingTimestamp=null,
          deleteTimestamp=null)
        ),
      sourceIPAddress=192.158.1.38,
        userIdentity={
          UserIdentity: {
```

```
            arn=arn:aws:sts::111122223333:assumed-role/TestAssumeRole-us-west-2/
ControlPlane-IntegTest-68211a2a-3e9d-42b7-86ac-c682520e0410,
            invokedBy=null,
            accessKeyId=TESTXECZ5U2ZULLHHMJG,
            type=AssumedRole,
            sessionContext={
              SessionContext: {
                sessionIssuer={
                  SessionIssuer: {arn=arn:aws:iam::111122223333:role/TestAssumeRole-us-
west-2,
                  type=Role,
                  accountId=111122223333,
                  userName=TestAssumeRole-us-west-2,
                  principalId=TESTXECZ5U9M4LGF2N6Y5}
                },
                attributes={
                  SessionContextAttributes: {
                    creationDate=Sun May 21 18:58:31 UTC 2023,
                    mfaAuthenticated=false
                  }
                },
                webIdFederationData=null
              }
            },
          username=null,
          principalId=: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={
    }
  }
}
```

Comprendre les AWS entrées du fichier journal du plan de données de cryptographie des paiements

Les événements du plan de données peuvent éventuellement être configurés et fonctionner de la même manière que les journaux du plan de contrôle, mais il s'agit généralement de volumes beaucoup plus importants. Compte tenu de la nature sensible de certaines entrées et sorties relatives aux opérations du plan de données de cryptographie des AWS paiements, certains champs peuvent contenir le message « *** Données sensibles expurgées*** ». Ceci n'est pas configurable et vise à empêcher l'apparition de données sensibles dans les journaux ou les traces.

L'exemple suivant montre une entrée de CloudTrail journal illustrant l'`EncryptData`action AWS de chiffrement des paiements.

```
      {
        "Records": [
            {
                "eventVersion": "1.09",
                "userIdentity": {
                    "type": "AssumedRole",
                    "principalId": "TESTXECZ5U2ZULLHHMJG:DataPlane-User",
                    "arn": "arn:aws:sts::111122223333:assumed-role/Admin/DataPlane-
User",
                    "accountId": "111122223333",
                    "accessKeyId": "TESTXECZ5U2ZULLHHMJG",
                    "userName": "",
                    "sessionContext": {
                        "sessionIssuer": {
                            "type": "Role",
                            "principalId": "TESTXECZ5U9M4LGF2N6Y5",
                            "arn": "arn:aws:iam::111122223333:role/Admin",
                            "accountId": "111122223333",
                            "userName": "Admin"
                        },
                        "attributes": {
                            "creationDate": "2024-07-09T14:23:05Z",
                            "mfaAuthenticated": "false"
                        }
                    }
```

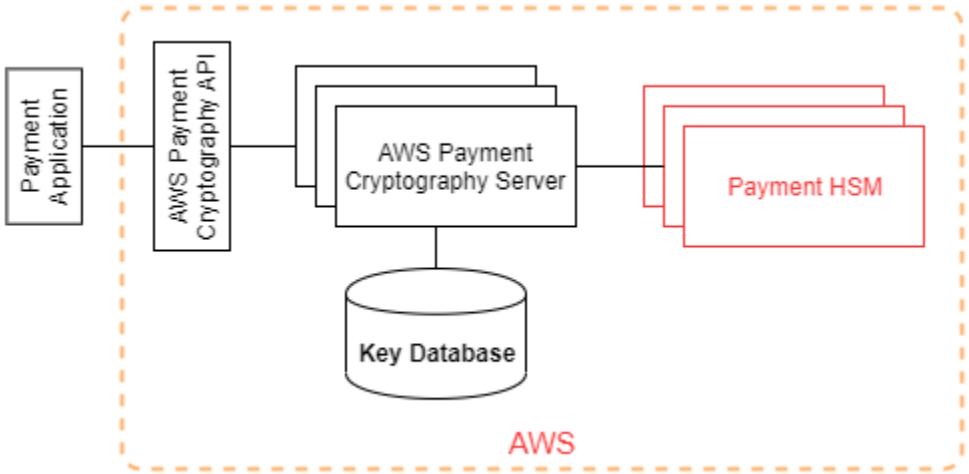
```
                },
                "eventTime": "2024-07-09T14:24:02Z",
                "eventSource": "payment-cryptography.amazonaws.com",
                "eventName": "GenerateCardValidationData",
                "awsRegion": "us-east-2",
                "sourceIPAddress": "192.158.1.38",
                "userAgent": "aws-cli/2.17.6 md/awscrt#0.20.11 ua/2.0 os/macos#23.4.0
 md/arch#x86_64 lang/python#3.11.8 md/pyimpl#CPython cfg/retry-mode#standard md/
installer#exe md/prompt#off md/command#payment-cryptography-data.generate-card-
validation-data",
                "requestParameters": {
                    "key_identifier": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwsp",
                    "primary_account_number": "*** Sensitive Data Redacted ***",
                    "generation_attributes": {
                        "CardVerificationValue2": {
                            "card_expiry_date": "*** Sensitive Data Redacted ***"
                        }
                    }
                },
                "responseElements": null,
                "requestID": "f2a99da8-91e2-47a9-b9d2-1706e733991e",
                "eventID": "e4eb3785-ac6a-4589-97a1-babdd3d4dd95",
                "readOnly": true,
                "resources": [
                    {
                        "accountId": "111122223333",
                        "type": "AWS::PaymentCryptography::Key",
                        "ARN": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwsp"
                    }
                ],
                "eventType": "AwsApiCall",
                "managementEvent": false,
                "recipientAccountId": "111122223333",
                "eventCategory": "Data",
                "tlsDetails": {
                    "tlsVersion": "TLSv1.3",
                    "cipherSuite": "TLS_AES_128_GCM_SHA256",
                    "clientProvidedHostHeader": "dataplane.payment-cryptography.us-
east-2.amazonaws.com"
                }
            }
        ]
```

```
}
```

Détails cryptographiques

AWS Payment Cryptography fournit une interface Web permettant de générer et de gérer des clés cryptographiques pour les transactions de paiement. AWS Payment Cryptography propose des services standard de gestion des clés et de cryptographie des transactions de paiement, ainsi que des outils que vous pouvez utiliser pour une gestion et un audit centralisés. Cette documentation fournit une description détaillée des opérations cryptographiques que vous pouvez utiliser dans AWS Payment Cryptography pour vous aider à évaluer les fonctionnalités proposées par le service.

AWS La cryptographie des paiements contient plusieurs interfaces (y compris une API RESTful, via la CLI AWS, le SDK AWS et le AWS Management Console) pour demander des opérations cryptographiques à un parc distribué de modules de sécurité matériels certifiés PCI PTS HSM.



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

AWS Les serveurs de cryptographie des paiements se connectent au HSM sous-jacent via un réseau privé non virtuel. Les connexions entre les composants du service et le HSM sont sécurisées par le protocole TLS mutuel (MTL) pour l'authentification et le chiffrement.

Objectifs de conception

AWSLa cryptographie des paiements est conçue pour répondre aux critères suivants :

- Fiable : l'utilisation des clés est protégée par des politiques de contrôle d'accès que vous définissez et gérez. Il n'existe aucun mécanisme permettant d'exporter des clés cryptographiques AWS de paiement en texte brut. La confidentialité de vos clés cryptographiques est primordiale. Plusieurs employés d'Amazon disposant d'un accès spécifique aux contrôles d'accès basés sur le quorum doivent effectuer des actions administratives sur les HSM. Aucun employé d'Amazon n'a accès aux clés principales (ou principales) ou aux sauvegardes du HSM. Les clés principales ne peuvent pas être synchronisées avec les HSM qui ne font pas partie d'une région de cryptographie des AWS paiements. Toutes les autres clés sont protégées par les clés principales HSM. Par conséquent, les clés AWS de cryptographie des paiements du client ne sont pas utilisables en dehors du service de cryptographie des AWS paiements opérant sur le compte du client.
- Faible latence et haut débit — La cryptographie des AWS paiements fournit des opérations cryptographiques à des niveaux de latence et de débit adaptés à la gestion des clés cryptographiques de paiement et au traitement des transactions de paiement.
- Durabilité : la durabilité des clés cryptographiques est conçue pour être égale à celle des services de durabilité les plus élevés dans AWS. Une clé cryptographique unique peut être partagée avec un terminal de paiement, une carte à puce EMV ou tout autre dispositif cryptographique sécurisé (SCD) utilisé depuis de nombreuses années.
- Régions indépendantes : AWS fournit des régions indépendantes pour les clients qui ont besoin de restreindre l'accès aux données dans différentes régions ou qui ont besoin de se conformer aux critères de résidence des données. L'utilisation de la clé peut être isolée dans une région AWS.
- Source validée de nombres aléatoires : Étant donné que la cryptographie forte dépend de la génération de nombres aléatoires vraiment imprévisibles, étant donné que la cryptographie des AWS paiements fournit une source validée de nombres aléatoires de haute qualité. Toutes les générations de clés pour la cryptographie des AWS paiements utilisent un HSM répertorié PCI PTS HSM, fonctionnant en mode PCI.
- Audit — La cryptographie des AWS paiements enregistre l'utilisation et la gestion des clés cryptographiques dans les CloudTrail journaux et les journaux de service disponibles via Amazon. CloudWatch Vous pouvez utiliser CloudTrail les journaux pour inspecter l'utilisation de vos clés cryptographiques, notamment l'utilisation des clés par les comptes avec lesquels vous avez partagé des clés. AWS La cryptographie des paiements est vérifiée par des évaluateurs tiers par rapport aux normes PCI, aux marques de cartes et aux normes régionales de sécurité

des paiements applicables. Les guides sur les attestations et la responsabilité partagée sont disponibles sur AWS Artifact.

- Elastic — La cryptographie des AWS paiements évolue progressivement en fonction de vos besoins. Au lieu de prévoir et de réserver la capacité du HSM, AWS Payment Cryptography fournit une cryptographie des paiements à la demande. AWS La cryptographie des paiements est chargée de maintenir la sécurité et la conformité du HSM afin de fournir une capacité suffisante pour répondre aux pics de demande des clients.

Fondations

Les rubriques de ce chapitre décrivent les primitives cryptographiques de la cryptographie des AWS paiements et les endroits où elles sont utilisées. Ils présentent également les éléments de base du service.

Rubriques

Primitives cryptographiques

AWS La cryptographie des paiements utilise des algorithmes cryptographiques standard paramétrables afin que les applications puissent implémenter les algorithmes nécessaires à leur cas d'utilisation. L'ensemble des algorithmes cryptographiques est défini par les normes PCI, ANSI X9, EMVco et ISO. Toute la cryptographie est effectuée par des HSM conformes à la norme PCI PTS HSM exécutés en mode PCI.

Entropie et génération de nombres aléatoires

AWS La génération de clés de cryptographie de paiement est effectuée sur les HSM de cryptographie AWS de paiement. Les HSM mettent en œuvre un générateur de nombres aléatoires qui répond aux exigences HSM de la norme PCI PTS pour tous les types de clés et paramètres pris en charge.

Opérations clés symétriques

Les algorithmes clés symétriques et les points forts définis dans les normes ANSI X9 TR 31, ANSI X9.24 et PCI PIN Annex C sont pris en charge :

- Fonctions de hachage — Algorithmes des familles SHA2 et SHA3 dont la taille de sortie est supérieure à 2551. À l'exception de la rétrocompatibilité avec les terminaux pré-PCI PTS POI v3.
- Chiffrement et déchiffrement : AES avec une taille de clé supérieure ou égale à 128 bits, ou TDEA avec une taille de clé supérieure ou égale à 112 bits (2 clés ou 3 clés).
- Codes d'authentification de message (MAC) CMAC ou GMAC avec AES, ainsi que HMAC avec une fonction de hachage approuvée et une taille de clé supérieure ou égale à 128.

AWS La cryptographie des paiements utilise la norme AES 256 pour les clés principales HSM, les clés de protection des données et les clés de session TLS.

Remarque : Certaines des fonctions répertoriées sont utilisées en interne pour prendre en charge les protocoles et les structures de données standard. Consultez la documentation de l'API pour connaître les algorithmes pris en charge par des actions spécifiques.

Opérations clés asymétriques

Les algorithmes clés asymétriques et les points forts définis dans les normes ANSI X9 TR 31, ANSI X9.24 et PCI PIN Annex C sont pris en charge :

- Schémas d'établissement de clés approuvés, tels que décrits dans le NIST SP800-56A (accord de clé basé sur ECC/FCC2), le NIST SP800-56B (accord de clé basé sur IFC) et le NIST SP800-38F (cryptage/encapsulage de clés basé sur AES).

AWS Les hôtes de Payment Cryptography autorisent uniquement les connexions au service via TLS avec une suite de chiffrement garantissant une parfaite confidentialité des transmissions.

Remarque : Certaines des fonctions répertoriées sont utilisées en interne pour prendre en charge les protocoles et les structures de données standard. Consultez la documentation de l'API pour connaître les algorithmes pris en charge par des actions spécifiques.

Rangement des clés

AWS Les clés de cryptographie de paiement sont protégées par des clés principales HSM AES 256 et stockées dans des blocs de clés ANSI X9 TR 31 dans une base de données cryptée. La base de données est répliquée dans une base de données en mémoire sur les serveurs de cryptographie des AWS paiements.

Selon l'annexe C de la norme de sécurité PCI PIN, les clés AES 256 sont aussi puissantes ou plus puissantes que :

- TDEA à 3 touches
- RSA 15360 bits
- ECC 512 bits
- DSA, DH et MQV 15360/512

Importation de clés à l'aide de clés symétriques

AWS La cryptographie des paiements prend en charge l'importation de cryptogrammes et de blocs de clés dotés de clés symétriques ou publiques avec une clé de chiffrement à clé symétrique (KEK) aussi forte ou plus puissante que la clé protégée à importer.

Importation de clés à l'aide de clés asymétriques

AWS La cryptographie des paiements prend en charge l'importation de cryptogrammes et de blocs de clés dotés de clés symétriques ou publiques protégées par une clé de chiffrement à clé privée (KEK) aussi forte ou plus puissante que la clé protégée à importer. L'authenticité et l'intégrité de la clé publique fournie pour le déchiffrement doivent être garanties par un certificat délivré par une autorité de confiance du client.

Les KEK publics fournis par AWS Payment Cryptography disposent de l'authentification et de la protection d'intégrité d'une autorité de certification (CA) avec une conformité attestée à la sécurité par code PIN PCI et à l'annexe A de la norme PCI P2PE.

Exportation de clés

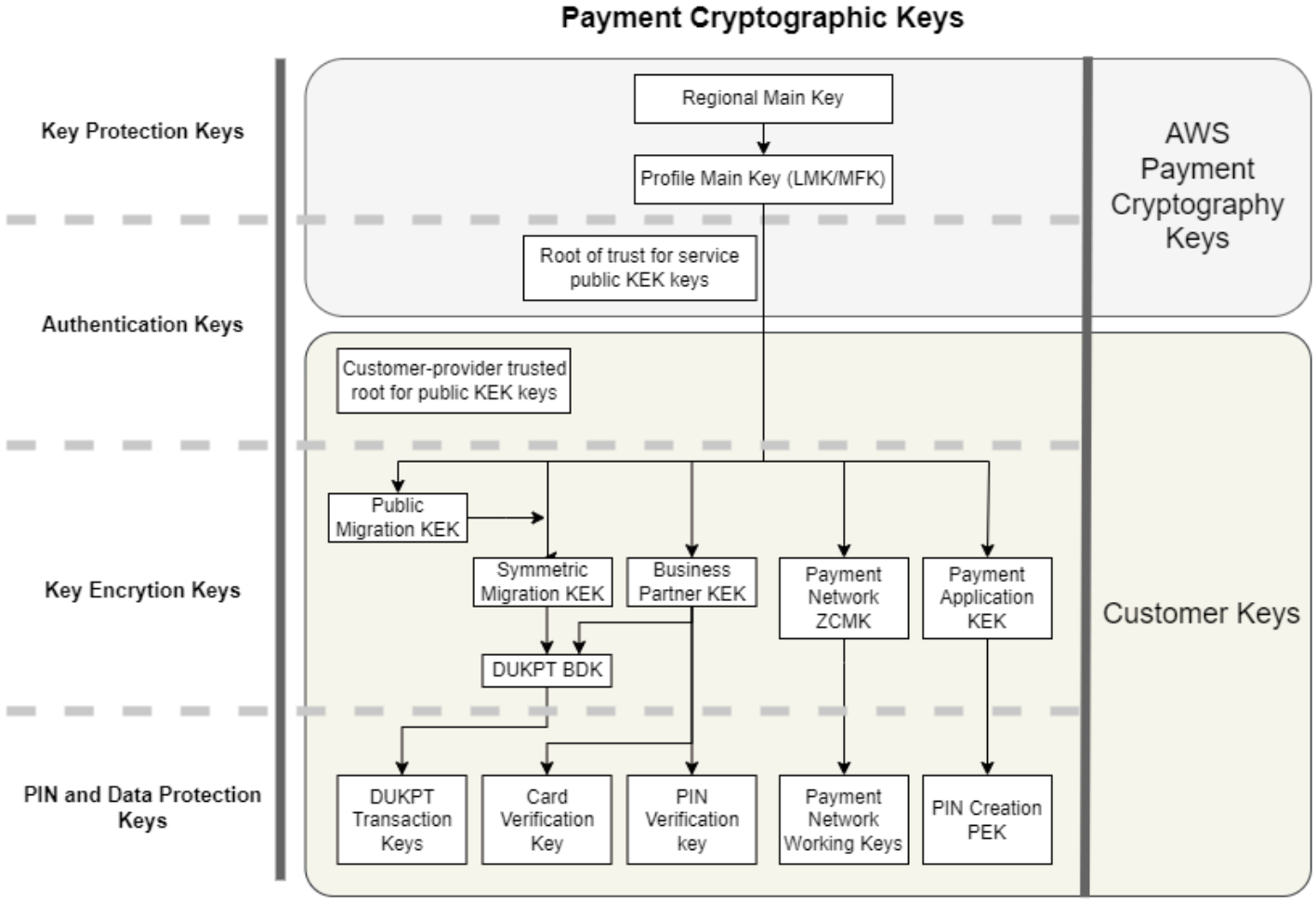
Les clés peuvent être exportées et protégées par des clés dotées de la valeur appropriée KeyUsage et qui sont aussi fortes ou plus fortes que la clé à exporter.

Protocole DUKPT (clé unique dérivée par transaction)

AWS La cryptographie des paiements prend en charge les clés de dérivation de base (BDK) TDEA et AES, comme décrit par la norme ANSI X9.24-3.

Hiérarchie des clés

La hiérarchie des clés de chiffrement des AWS paiements garantit que les clés sont toujours protégées par des clés aussi fortes ou plus fortes que les clés qu'elles protègent.



AWS Les clés cryptographiques de paiement sont utilisées pour la protection des clés au sein du service :

Clé	Description
Clé principale régionale	Protège les images ou profils HSM virtuels utilisés pour le traitement cryptographique. Cette clé n'existe que dans le HSM et les sauvegardes sécurisées.
Clé principale du profil	Clé de protection de clé client de haut niveau, traditionnellement appelée clé principale locale (LMK) ou clé de fichier principale (MFK) pour les clés client. Cette clé n'existe que dans le HSM et les sauvegardes sécurisées. Les profils définissent des configurations HSM distinctes conformément aux normes de sécurité pour les cas d'utilisation des paiements.
Racine de confiance pour les clés de chiffrement à clé publique (KEK) de cryptographie des AWS paiements	La clé publique racine fiable et le certificat d'authentification et de validation des clés publiques fournis par AWS Payment Cryptography pour l'importation et l'exportation de clés à l'aide de clés asymétriques.

Les clés client sont regroupées en fonction des clés utilisées pour protéger les autres clés et des clés qui protègent les données relatives au paiement. Voici des exemples de clés client des deux types :

Clé	Description
Root sécurisé fourni par le client pour les clés KEK publiques	Clé publique et certificat fournis par vous en tant que base de confiance pour authentifier et valider les clés publiques que vous fournissez pour l'importation et l'exportation de clés à l'aide de clés asymétriques.
Clés de chiffrement clés (KEK)	Les KEK sont utilisés uniquement pour chiffrer d'autres clés destinées à être échangées entre des magasins de clés externes et AWS Payment Cryptography, des partenair

Clé	Description
	es commerciaux, des réseaux de paiement ou différentes applications au sein de votre organisation.
Clé unique dérivée par transaction (DUKPT) Clé de dérivation de base (BDK)	Les BDK sont utilisés pour créer des clés uniques pour chaque terminal de paiement et traduire les transactions de plusieurs terminaux en une seule clé de travail de la banque acquéreuse, ou de l'acquéreur. La meilleure pratique, requise par le chiffrement point à point PCI (P2PE), est d'utiliser différents BDK pour différents modèles de terminaux, services d'injection de clés ou d'initialisation, ou pour toute autre segmentation afin de limiter l'impact de la compromission d'un BDK.
Clé principale de contrôle de zone du réseau de paiement (ZCMK)	Les ZCMK, également appelés clés de zone ou clés principales de zone, sont fournis par les réseaux de paiement pour établir les clés de travail initiales.
Clés de transaction DUKPT	Les terminaux de paiement configurés pour DUKPT obtiennent une clé unique pour le terminal et la transaction. Le HSM recevant la transaction peut déterminer la clé à partir de l'identifiant du terminal et du numéro de séquence de transaction.

Clé	Description
Clés de préparation des données des cartes	Les clés principales de l'émetteur EMV, les clés de carte EMV et les valeurs de vérificat ion, ainsi que les clés de protection des fichiers de données de personnalisation des cartes sont utilisées pour créer des données pour des cartes individuelles destinées à être utilisées par un fournisseur de personnalisation de cartes. Ces clés et données de validation cryptographique sont également utilisées par les banques émettrices, ou émetteurs, pour authentifier les données des cartes dans le cadre de l'autorisation des transactions.
Clés de préparation des données des cartes	Les clés principales de l'émetteur EMV, les clés de carte EMV et les valeurs de vérificat ion, ainsi que les clés de protection des fichiers de données de personnalisation des cartes sont utilisées pour créer des données pour des cartes individuelles destinées à être utilisées par un fournisseur de personnalisation de cartes. Ces clés et données de validation cryptographique sont également utilisées par les banques émettrices, ou émetteurs, pour authentifier les données des cartes dans le cadre de l'autorisation des transactions.
Clés de fonctionnement du réseau de paiement	Souvent appelées clé de travail de l'émetteu r ou clé de travail de l'acquéreur, ces clés cryptent les transactions envoyées ou reçues depuis les réseaux de paiement. Ces clés font l'objet d'une rotation fréquente par le réseau, souvent tous les jours ou toutes les heures. Il s'agit de clés de chiffrement par code PIN (PEK) pour les transactions par code PIN/débit.

Clé	Description
Clés de chiffrement (PEK) par numéro d'identification personnel (PIN)	Les applications qui créent ou déchiffrent des blocs de code PIN utilisent le PEK pour empêcher le stockage ou la transmission de code PIN en texte clair.

Opérations internes

Cette rubrique décrit les exigences internes mises en œuvre par le service pour sécuriser les clés des clients et les opérations cryptographiques dans le cadre d'un service de cryptographie et de gestion des clés de paiement évolutif et distribué à l'échelle mondiale.

Rubriques

Spécifications et cycle de vie du HSM

AWS La cryptographie des paiements utilise une flotte de HSM disponibles dans le commerce. Les HSM sont validés par la norme FIPS 140-2 de niveau 3 et utilisent également les versions du microprogramme et la politique de sécurité répertoriée sur la liste des périphériques PCI PTS approuvés par le PCI Security Standards Council en tant que conformes à la norme PCI HSM v3. La norme PCI PTS HSM inclut des exigences supplémentaires pour la fabrication, l'expédition, le déploiement, la gestion et la destruction du matériel HSM, qui sont importantes pour la sécurité et la conformité des paiements, mais qui ne sont pas traitées par la norme FIPS 140.

Tous les HSM fonctionnent en mode PCI et sont configurés selon la politique de sécurité PCI PTS HSM. Seules les fonctions nécessaires pour prendre en charge les cas d'utilisation de la

cryptographie des AWS paiements sont activées. AWS La cryptographie des paiements ne permet pas d'imprimer, d'afficher ou de renvoyer des codes PIN en texte clair.

Sécurité physique des appareils HSM

Seuls les HSM dont les clés d'appareil sont signées par une autorité de certification (CA) de cryptographie des AWS paiements (CA) par le fabricant avant la livraison peuvent être utilisés par le service. La cryptographie des AWS paiements est une sous-autorité de certification de l'autorité de certification du fabricant qui est à l'origine de la confiance en ce qui concerne les certificats des fabricants et des appareils HSM. L'autorité de certification du fabricant met en œuvre la norme ANSI TR 34 et a attesté la conformité à l'annexe A de sécurité du code PIN PCI et à l'annexe A de la norme PCI P2PE. Le fabricant vérifie que tous les HSM dotés de clés d'appareil signées par l'autorité de certification de chiffrement des AWS paiements sont expédiés au destinataire désigné par AWS.

Conformément à la norme PCI PIN Security, le fabricant fournit une liste de numéros de série via un canal de communication différent de celui utilisé pour l'expédition du HSM. Ces numéros de série sont vérifiés à chaque étape du processus d'installation du HSM dans les centres de données AWS. Enfin, les opérateurs de chiffrement des AWS paiements valident la liste des HSM installés par rapport à la liste du fabricant avant d'ajouter le numéro de série à la liste des HSM autorisés à recevoir des clés de chiffrement des AWS paiements.

Les HSM sont stockés de manière sécurisée ou font l'objet d'un double contrôle à tout moment, ce qui inclut :

- Expédition par le fabricant vers une installation d'assemblage de racks AWS.
- Pendant le montage du rack.
- Expédition depuis l'installation d'assemblage des racks vers un centre de données.
- Réception et installation dans la salle de traitement sécurisée d'un centre de données. Les racks HSM assurent un double contrôle avec des serrures à accès contrôlé par carte, des capteurs de porte avec alarme et des caméras.
- Pendant les opérations.
- Lors de la mise hors service et de la destruction.

Un compte rendu complet chain-of-custody, assorti d'une responsabilité individuelle, est maintenu et contrôlé pour chaque HSM.

Initialisation du HSM

Un HSM n'est initialisé dans le cadre de la flotte de cryptographie des AWS paiements qu'une fois que son identité et son intégrité ont été validées par des numéros de série, des clés de périphérique installées par le fabricant et une somme de contrôle du microprogramme. Une fois l'authenticité et l'intégrité d'un HSM validées, celui-ci est configuré, notamment en activant le mode PCI. Ensuite, les clés principales de la région de cryptographie des AWS paiements et les clés principales du profil sont établies et le HSM est mis à la disposition du service.

Service et réparation du HSM

Les HSM comportent des composants réparables qui ne nécessitent pas de violation de la limite cryptographique de l'appareil. Ces composants incluent les ventilateurs de refroidissement, les blocs d'alimentation et les batteries. Si un HSM ou un autre périphérique du rack HSM a besoin d'être réparé, le double contrôle est maintenu pendant toute la période d'ouverture du rack.

Mise hors service du HSM

La mise hors service est due à une défaillance end-of-life ou à une défaillance d'un HSM. Les HSM sont logiquement mis à zéro avant d'être retirés de leur rack, s'ils sont fonctionnels, puis détruits dans les salles de traitement sécurisées des centres de données AWS. Ils ne sont jamais renvoyés au fabricant pour réparation, utilisés à d'autres fins ou retirés d'une salle de traitement sécurisée avant d'être détruits.

Mise à jour du microprogramme HSM

Les mises à jour du microprogramme HSM sont appliquées lorsque cela est nécessaire pour maintenir l'alignement avec les versions répertoriées PCI PTS HSM et FIPS 140-2 (ou FIPS 140-3), si une mise à jour est liée à la sécurité ou s'il est déterminé que les clients peuvent bénéficier des fonctionnalités d'une nouvelle version. AWS Les HSM de chiffrement des paiements exécutent le off-the-shelf microprogramme correspondant aux versions répertoriées par PCI PTS HSM. L'intégrité des nouvelles versions du microprogramme est validée avec les versions de microprogramme certifiées PCI ou FIPS, puis leur fonctionnalité est testée avant d'être déployées sur tous les HSM.

Accès de l'opérateur

Les opérateurs peuvent accéder au HSM sans console à des fins de dépannage dans de rares cas où les informations recueillies auprès du HSM au cours des opérations normales ne sont pas

suffisantes pour identifier un problème ou planifier une modification. Les étapes suivantes sont exécutées :

- Les activités de dépannage sont développées et approuvées et la session hors console est planifiée.
- Un HSM est supprimé du service de traitement des clients.
- Les touches principales sont supprimées, sous double contrôle.
- L'opérateur est autorisé à accéder au HSM sans console pour effectuer des activités de dépannage approuvées, sous double contrôle.
  - Après la fin de la session hors console, le processus de provisionnement initial est effectué sur le HSM, en renvoyant le microprogramme et la configuration standard, puis en synchronisant la clé principale, avant de renvoyer le HSM au service des clients.
  - Les enregistrements de la session sont enregistrés dans le suivi des modifications.
  - Les informations obtenues lors de la session sont utilisées pour planifier les modifications futures.

Tous les enregistrements d'accès hors console sont examinés pour vérifier la conformité des processus et les modifications potentielles apportées à la surveillance du HSM, au processus de non-console-access gestion ou à la formation des opérateurs.

Gestion des clés

Tous les HSM d'une région sont synchronisés avec une clé principale de région. Une clé principale de région protège au moins une clé principale de profil. Une clé principale de profil protège les clés des clients.

Toutes les clés principales sont générées par un HSM et distribuées par distribution de clés symétrique à l'aide de techniques asymétriques, conformément à la norme ANSI X9 TR 34 et à l'annexe A du code PIN PCI.

Rubriques

- Rotation de la clé principale du profil
- Protection
- Durabilité
- Sécurité des communications
- Gestion des clés clients
- Journalisation et surveillance

Génération

Les clés principales AES 256 bits sont générées sur l'un des HSM fournis pour le parc de services HSM, à l'aide du générateur de nombres aléatoires PCI PTS HSM.

Synchronisation des touches principales de la région

Les clés principales de la région HSM sont synchronisées par le service sur l'ensemble de la flotte régionale avec les mécanismes définis par la norme ANSI X9 TR-34, notamment :

- Authentification mutuelle à l'aide de clés et de certificats de l'hôte de distribution de clés (KDH) et du dispositif de réception de clés (KRD) pour garantir l'authentification et l'intégrité des clés publiques.
- Les certificats sont signés par une autorité de certification (CA) qui répond aux exigences de l'annexe A2 du code PIN PCI, à l'exception des algorithmes asymétriques et des points forts de clé appropriés pour protéger les clés AES 256 bits.
- Identification et protection des clés symétriques distribuées conformément à la norme ANSI X9 TR-34 et à l'annexe A1 du code PIN PCI, à l'exception des algorithmes asymétriques et des atouts de clé appropriés pour protéger les clés AES 256 bits.

Les clés principales de région sont établies pour les HSM qui ont été authentifiés et approvisionnés pour une région par :

- Une clé principale est générée sur un HSM de la région. Ce HSM est désigné comme hôte de distribution des clés.
- Tous les HSM approvisionnés dans la région génèrent un jeton d'authentification KRD, qui contient la clé publique du HSM et des informations d'authentification non rejouables.
- Les jetons KRD sont ajoutés à la liste d'autorisation du KDH une fois que le KDH a validé l'identité et l'autorisation du HSM de recevoir des clés.

- Le KDH produit un jeton de clé principale authentifiable pour chaque HSM. Les jetons contiennent des informations d'authentification KDH et une clé principale cryptée qui ne peut être chargée que sur un HSM pour lequel elle a été créée.
- Chaque HSM reçoit le jeton de clé principal conçu pour lui. Après avoir validé les propres informations d'authentification du HSM et les informations d'authentification KDH, la clé principale est déchiffrée par la clé privée KRD et chargée dans la clé principale.

Dans le cas où un seul HSM doit être resynchronisé avec une région :

- Il est revalidé et approvisionné avec le microprogramme et la configuration.
- S'il s'agit d'un nouveau produit dans la région :
  - Le HSM génère un jeton d'authentification KRD.
  - Le KDH ajoute le jeton à sa liste d'autorisations.
  - Le KDH génère un jeton de clé principal pour le HSM.
  - Le HSM charge la clé principale.
  - Le HSM est mis à la disposition du service.

Cela garantit que :

- Seul le HSM validé pour le traitement AWS de cryptographie des paiements dans une région peut recevoir la clé principale de cette région.
- Seule une clé principale provenant d'un HSM de chiffrement des AWS paiements peut être distribuée à un HSM de la flotte.

Rotation des clés principales de la région

Les clés principales des régions font l'objet d'une rotation à l'expiration de la période de chiffrement, dans le cas peu probable d'une compromission présumée de la clé, ou après des modifications du service dont il est établi qu'elles ont un impact sur la sécurité de la clé.

Une nouvelle clé principale de région est générée et distribuée comme lors du provisionnement initial. Les clés principales du profil enregistrées doivent être converties en la clé principale de la nouvelle région.

La rotation des principales clés de la région n'a aucune incidence sur le traitement des clients.

Synchronisation des touches principales du profil

Les clés principales du profil sont protégées par les clés principales des régions. Cela limite un profil à une région spécifique.

Les clés principales du profil sont configurées en conséquence :

- Une clé principale de profil est générée sur un HSM dont la clé principale de région est synchronisée.
- La clé principale du profil est stockée et cryptée avec la configuration du profil et d'autres contextes.
- Le profil est utilisé pour les fonctions cryptographiques des clients par n'importe quel HSM de la région doté de la clé principale de région.

Rotation de la clé principale du profil

Les clés principales du profil font l'objet d'une rotation à l'expiration de la période de chiffrement, en cas de suspicion de compromission de la clé ou après des modifications du service dont il est établi qu'elles ont un impact sur la sécurité de la clé.

Étapes de rotation :

- Une nouvelle clé principale de profil est générée et distribuée en tant que clé principale en attente, comme lors du provisionnement initial.
- Un processus d'arrière-plan traduit les informations clés du client de la clé principale du profil établie à la clé principale en attente.
- Lorsque toutes les clés du client ont été chiffrées avec la clé en attente, la clé en attente est promue clé principale du profil.
- Un processus en arrière-plan supprime le contenu clé du client protégé par la clé expirée.

La rotation des clés principales du profil n'a aucune incidence sur le traitement des clients.

Protection

Les clés ne dépendent que de la hiérarchie des clés pour la protection. La protection des clés principales est essentielle pour éviter la perte ou la compromission de toutes les clés des clients.

Les clés principales de région peuvent être restaurées à partir d'une sauvegarde uniquement vers un HSM authentifié et provisionné pour le service. Ces clés ne peuvent être stockées que sous forme de jetons de clé principale chiffrés et authentifiables mutuellement provenant d'un KDH spécifique pour un HSM spécifique.

Les clés principales du profil sont stockées avec la configuration du profil et les informations contextuelles chiffrées par région.

Les clés client sont stockées dans des blocs de clés, protégés par une clé principale de profil.

Toutes les clés existent exclusivement dans un HSM ou sont stockées protégées par une autre clé de puissance cryptographique égale ou supérieure.

Durabilité

Les clés client pour la cryptographie des transactions et les fonctions commerciales doivent être disponibles même dans des situations extrêmes susceptibles de provoquer des pannes. AWS La cryptographie des paiements utilise un modèle de redondance à plusieurs niveaux dans les zones de disponibilité et les régions. AWS Les clients qui exigent une disponibilité et une durabilité des opérations cryptographiques de paiement supérieures à celles fournies par le service doivent mettre en œuvre des architectures multirégionales.

L'authentification HSM et les jetons de clé principale sont enregistrés et peuvent être utilisés pour restaurer une clé principale ou synchroniser avec une nouvelle clé principale, dans le cas où un HSM doit être réinitialisé. Les jetons sont archivés et utilisés uniquement sous double contrôle lorsque cela est nécessaire.

Sécurité des communications

Externe

AWS Les points de terminaison de l'API de cryptographie des paiements répondent aux normes de AWS sécurité, notamment le protocole TLS 1.2 ou supérieur et la version 4 de Signature pour l'authentification et l'intégrité des demandes.

Les connexions TLS entrantes sont interrompues sur les équilibreurs de charge réseau et transmises aux gestionnaires d'API via des connexions TLS internes.

Internal (Interne)

Les communications internes entre les composants de service et entre les composants de service et les autres services AWS sont protégées par le protocole TLS à l'aide d'une cryptographie renforcée.

Les HSM se trouvent sur un réseau privé non virtuel accessible uniquement à partir de composants de service. Toutes les connexions entre le HSM et les composants de service sont sécurisées par le protocole TLS mutuel (mTLS), égal ou supérieur au protocole TLS 1.2. Les certificats internes pour TLS et MTL sont gérés par Amazon Certificate Manager à l'aide d'une autorité de certification privée AWS. Les VPC internes et le réseau HSM sont surveillés pour détecter les activités non exceptées et les modifications de configuration.

Gestion des clés clients

Chez AWS, la confiance du client est notre priorité absolue. Vous gardez le contrôle total des clés que vous téléchargez ou créez dans le service sous votre compte AWS et vous êtes responsable de la configuration de l'accès aux clés.

AWS Payment Cryptography est entièrement responsable de la conformité physique du HSM et de la gestion des clés pour les clés gérées par le service. Cela nécessite la propriété et la gestion des clés principales du HSM et le stockage des clés clients protégées dans la base de données de clés AWS de cryptographie des paiements.

Séparation de l'espace clé pour le client

AWS La cryptographie des paiements applique des politiques relatives aux clés pour toutes les utilisations des clés, notamment en limitant les principaux au compte propriétaire de la clé, sauf si une clé est explicitement partagée avec un autre compte.

Sauvegarde et restauration

Les clés et les informations clés d'une région sont sauvegardées dans des archives cryptées par AWS. Les archives nécessitent un double contrôle par AWS pour être restaurées.

Blocs clés

Toutes les clés sont stockées dans des blocs de touches au format ANSI X9 TR-31.

Les clés peuvent être importées dans le service à partir de cryptogrammes ou d'autres formats de blocs de clés pris en charge par ImportKey. De même, les clés peuvent être exportées, si elles sont exportables, vers d'autres formats de blocs de clés ou cryptogrammes pris en charge par les profils d'exportation de clés.

Utilisation des clés

L'utilisation des clés est limitée à celle configurée KeyUsage par le service. Le service échouera à toute demande impliquant une utilisation de clé, un mode d'utilisation ou un algorithme inappropriés pour l'opération cryptographique demandée.

Principales relations d'échange

Les normes PCI PIN Security et PCI P2PE obligent les entreprises qui partagent des clés chiffrant des codes PIN, y compris le KEK utilisé pour partager ces clés, à ne pas les partager avec d'autres organisations. Il est recommandé que les clés symétriques ne soient partagées qu'entre deux parties, y compris au sein d'une même organisation. Cela permet de minimiser l'impact des compromissions présumées qui obligent à remplacer les clés concernées.

Même les analyses de rentabilisation qui nécessitent le partage de clés entre plus de deux parties devraient limiter le nombre de parties au minimum.

AWS La cryptographie des paiements fournit des balises clés qui peuvent être utilisées pour suivre et appliquer l'utilisation des clés conformément à ces exigences.

Par exemple, les clés KEK et BDK de différentes installations d'injection de clés peuvent être identifiées en définissant un « KIF » = « PosStation » pour toutes les clés partagées avec ce fournisseur de services. Un autre exemple serait de marquer les clés partagées avec les réseaux de paiement avec « Network » = « PayCard ». Le balisage vous permet de créer des contrôles d'accès et de créer des rapports d'audit pour appliquer et démontrer vos principales pratiques de gestion.

Suppression de la clé

DeleteKey marque les clés de la base de données en vue de leur suppression après une période configurée par le client. Passé ce délai, la clé est irrémédiablement supprimée. Il s'agit d'un mécanisme de sécurité destiné à empêcher la suppression accidentelle ou malveillante d'une clé. Les touches marquées pour suppression ne sont disponibles que pour aucune action, sauf RestoreKey.

Les clés supprimées restent dans les sauvegardes du service pendant 7 jours après leur suppression. Ils ne sont pas restaurables pendant cette période.

Les clés appartenant à des comptes AWS fermés sont marquées pour suppression. Si le compte est réactivé avant la fin de la période de suppression, toutes les clés marquées pour suppression sont restaurées, mais désactivées. Vous devez les réactiver pour pouvoir les utiliser pour des opérations cryptographiques.

Journalisation et surveillance

Les journaux de service internes incluent :

- CloudTrail journaux des appels de service AWS effectués par le service
- CloudWatch journaux des deux événements directement enregistrés dans les CloudWatch journaux ou des événements depuis HSM
- Fichiers journaux provenant du HSM et des systèmes de service
- Archives du journal

Toutes les sources de journaux surveillent et filtrent les informations sensibles, y compris celles relatives aux clés. Les journaux sont systématiquement revus pour s'assurer qu'ils ne contiennent pas d'informations sensibles sur les clients.

L'accès aux journaux est limité aux personnes nécessaires pour remplir les rôles professionnels.

Tous les journaux sont conservés conformément aux politiques de conservation des journaux d'AWS.

Opérations auprès des clients

AWS Payment Cryptography est entièrement responsable de la conformité physique du HSM aux normes PCI. Le service fournit également un magasin de clés sécurisé et garantit que les clés ne peuvent être utilisées qu'aux fins autorisées par les normes PCI et spécifiées par vous lors de la création ou de l'importation. Vous êtes responsable de la configuration des attributs clés et de l'accès afin de tirer parti des fonctionnalités de sécurité et de conformité du service.

Rubriques

- Génération de clés
- Importation de clés
- Exportation de clés
- Suppression des clés
- Rotating keys

Génération de clés

Lorsque vous créez des clés, vous définissez les attributs que le service utilise pour garantir une utilisation conforme de la clé :

- Algorithme et longueur de clé
- Utilisation
- Disponibilité et expiration

Les balises utilisées pour le contrôle d'accès basé sur les attributs (ABAC) sont utilisées pour limiter les clés à utiliser avec des partenaires spécifiques ou les applications doivent également être définies lors de la création. Veillez à inclure des politiques limitant les rôles autorisés à supprimer ou à modifier des balises.

Vous devez vous assurer que les politiques qui déterminent les rôles autorisés à utiliser et à gérer la clé sont définies avant la création de la clé.

> **Note**
>
> Les politiques IAM relatives aux CreateKey commandes peuvent être utilisées pour appliquer et démontrer un double contrôle pour la génération de clés.

Importation de clés

Lors de l'importation de clés, les attributs destinés à garantir une utilisation conforme de la clé sont définis par le service à l'aide des informations cryptographiquement liées contenues dans le bloc de clés. Le mécanisme de définition du contexte clé fondamental consiste à utiliser des blocs clés créés avec le HSM source et protégés par un KEK partagé ou asymétrique. Cela correspond aux exigences du code PIN PCI et préserve l'utilisation, l'algorithme et la force clé de l'application source.

Les attributs clés, les balises et les politiques de contrôle d'accès importants doivent être établis lors de l'importation, en plus des informations contenues dans le bloc clé.

L'importation de clés à l'aide de cryptogrammes ne transfère pas les attributs de clé depuis l'application source. Vous devez définir les attributs de manière appropriée à l'aide de ce mécanisme.

Les clés sont souvent échangées à l'aide de composants en texte clair, transmises par les dépositaires des clés, puis chargées lors d'une cérémonie mettant en œuvre un double contrôle dans

une pièce sécurisée. Cela n'est pas directement pris en charge par AWS Payment Cryptography. L'API exportera une clé publique avec un certificat qui peut être importé par votre propre HSM pour exporter un bloc de clés importable par le service. Vous permet d'utiliser votre propre HSM pour charger des composants en texte clair.

Vous devez utiliser les valeurs de contrôle des clés (KCV) pour vérifier que les clés importées correspondent aux clés source.

Les politiques IAM sur l' ImportKey API peuvent être utilisées pour appliquer et démontrer le double contrôle pour l'importation de clés.

Exportation de clés

Le partage de clés avec des partenaires ou des applications sur site peut nécessiter l'exportation de clés. L'utilisation de blocs clés pour les exportations permet de maintenir le contexte clé fondamental avec le contenu clé crypté.

Les balises clés peuvent être utilisées pour limiter l'exportation de clés vers KEK qui partagent la même balise et la même valeur.

AWS La cryptographie des paiements ne fournit ni n'affiche les composants clés en texte clair. Cela nécessite un accès direct des dépositaires de clés aux périphériques cryptographiques sécurisés (SCD) certifiés PCI PTS HSM ou ISO 13491 pour l'affichage ou l'impression. Vous pouvez établir un KEK asymétrique ou un KEK symétrique avec votre SCD pour organiser la cérémonie de création des composants clés en texte clair sous double contrôle.

Les valeurs de contrôle des touches (KCV) doivent être utilisées pour vérifier que les clés source importées par le HSM de destination correspondent.

Suppression des clés

Vous pouvez utiliser l'API de suppression des clés pour planifier la suppression des clés après une période que vous avez configurée. Avant cette date, les clés sont récupérables. Une fois les clés supprimées, elles sont définitivement supprimées du service.

Les politiques IAM sur l' DeleteKey API peuvent être utilisées pour appliquer et démontrer un double contrôle pour la suppression des clés.

Rotating keys

L'effet de rotation des clés peut être implémenté à l'aide d'un alias de clé en créant ou en important une nouvelle clé, puis en modifiant l'alias de clé pour qu'il fasse référence à la nouvelle clé. L'ancienne clé serait supprimée ou désactivée, en fonction de vos pratiques de gestion.

Quotas pour AWS Payment Cryptography

Votre compte AWS dispose de quotas par défaut, anciennement appelés limites, pour chaque service AWS. Sauf indication contraire, chaque quota est spécifique à une région. Vous pouvez demander des augmentations pour certains quotas, et d'autres quotas ne peuvent pas être augmentés.

Nom	Par défaut	Ajusta	Description
Alias	Chaque région prise en charge : 2 000	Oui	Le nombre maximum d'alias que vous pouvez avoir sur ce compte dans la région actuelle.
Taux combiné de demandes de plan de contrôle	Chaque région prise en charge : 5 par seconde	Oui	Le nombre maximum de demandes de plan de contrôle par seconde que vous pouvez effectuer sur ce compte dans la région actuelle. Ce quota s'applique à toutes les opérations du plan de contrôle combinées.
Taux combiné de demandes de plan de données (asymétrique)	Chaque région prise en charge : 20 par seconde	Oui	Le nombre maximum de demandes par seconde pour les opérations du plan de données avec une clé asymétrique que vous pouvez effectuer dans ce compte dans la région actuelle. Ce quota s'applique à toutes les opérations du plan de données combinées.

Nom	Par défaut	Ajusta	Description
Taux combiné de demandes de plan de données (symétrique)	Chaque Région prise en charge : 500 par seconde	Oui	Le nombre maximum de demandes par seconde pour les opérations du plan de données avec une clé symétrique que vous pouvez effectuer dans ce compte dans la région actuelle. Ce quota s'applique à toutes les opérations du plan de données combinées.
Clés	Chaque région prise en charge : 2 000	Oui	Le nombre maximum de clés que vous pouvez avoir dans ce compte dans la région actuelle, à l'exception des clés supprimées.

Historique du document pour le guide de l'utilisateur AWS de Payment Cryptography

Le tableau suivant décrit les versions de documentation relatives à la cryptographie des AWS paiements.

Modification	Description	Date
Lancement d'une nouvelle région	Points de terminaison supplémentaires pour le lancement dans de nouvelles régions en Europe (Francfort), en Europe (Irlande), en Asie-Pacifique (Singapour) et en Asie-Pacifique (Tokyo)	31 juillet 2024
CloudTrail pour Dataplane et Dynamic Keys	Ajout d'informations sur l'utilisation CloudTrail pour les opérations de plan de données (cryptographiques) , y compris des exemples. Nous avons également ajouté des informations sur l'utilisa tion de clés dynamiques pour certaines fonctions afin de mieux prendre en charge les clés à usage unique ou à usage limité qui ne doivent pas être importées dans AWS Payment Cryptography	10 juillet 2024
Exemples mis à jour	Ajout de nouveaux exemples pour l'émission de cartes	1er juillet 2024

Publication de fonctionnalités	Ajout d'informations sur les VPC endpoints (PrivateLink) et dans des CVV exemples.	30 mai 2024
Publication de fonctionnalités	Informations ajoutées sur les nouvelles fonctionnalités relatives à l'importation/exportation de clés, à l'utilisation RSA et à l'exportation de clés DUKPT IPEK /IK.	15 janvier 2024
Première version	Publication initiale du guide de l'utilisateur de la cryptographie des AWS paiements	8 juin 2023

Les traductions sont fournies par des outils de traduction automatique. En cas de conflit entre le contenu d'une traduction et celui de la version originale en anglais, la version anglaise prévaudra.