



Guide de l'utilisateur

# AWS Cryptographie des paiements



# AWS Cryptographie des paiements: Guide de l'utilisateur

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

Les marques et la présentation commerciale d'Amazon ne peuvent être utilisées en relation avec un produit ou un service qui n'est pas d'Amazon, d'une manière susceptible de créer une confusion parmi les clients, ou d'une manière qui dénigre ou discrédite Amazon. Toutes les autres marques commerciales qui ne sont pas la propriété d'Amazon appartiennent à leurs propriétaires respectifs, qui peuvent ou non être affiliés ou connectés à Amazon, ou sponsorisés par Amazon.

---

# Table of Contents

Qu'est-ce que AWS Cryptographie des paiements ? .....	1
Concepts .....	2
Terminologie du secteur .....	4
Types de clés courants .....	5
Autres termes .....	7
Services connexes .....	11
Pour plus d'informations .....	11
Points de terminaison .....	11
Points de terminaison du plan de contrôle .....	12
Points de terminaison du plan de données .....	12
Démarrer .....	13
Prérequis .....	13
Étape 1 : Création d'une clé .....	13
Étape 2 : générer une valeur CVV2 à l'aide de la clé .....	15
Étape 3 : Vérifiez la valeur générée à l'étape 2 .....	15
Étape 4 : Réaliser un test négatif .....	16
Étape 5 : (Facultatif) Nettoyer .....	16
Gestion de clés .....	18
Génération de clés .....	18
Génération d'une clé TDES 2KEY .....	19
Génération d'une clé de chiffrement par code PIN .....	20
Création d'une clé asymétrique (RSA) .....	21
Génération d'une clé de vérification du code PIN (PVV) .....	22
Clés de liste .....	23
Activation et désactivation des clés .....	24
Commencer à utiliser les clés .....	25
Arrêter l'utilisation des clés .....	26
Suppression des clés .....	27
À propos de la période d'attente .....	28
Clés d'importation et d'exportation .....	31
Clés d'importation .....	32
Clés d'exportation .....	42
Utilisation des alias .....	50
À propos des alias .....	51

Utilisation d'alias dans vos applications .....	54
API associées .....	55
Obtenez des clés .....	55
Obtenir la clé publique/le certificat associé à une paire de clés .....	56
Clés de balisage .....	57
À propos des balises dans la cryptographie des AWS paiements .....	58
Afficher les tags clés dans la console .....	59
Gestion des balises clés à l'aide des opérations d'API .....	59
Contrôle de l'accès aux balises .....	62
Utilisation de balises pour contrôler l'accès aux clés .....	66
Comprendre les principaux attributs .....	70
Clés symétriques .....	70
Clés asymétriques .....	73
Opérations relatives aux données .....	74
Chiffrer, déchiffrer et rechiffrer les données .....	74
Chiffrer des données .....	75
Déchiffrer des données .....	79
Générer et vérifier les données de la carte .....	83
Générer des données de carte .....	83
Vérifier les données de la carte .....	85
Générez, traduisez et vérifiez les données PIN .....	86
Translate code PIN .....	87
Générer des données PIN .....	89
Vérifier les données PIN .....	92
Cryptogramme de demande d'authentification (ARQC) .....	94
Création de données de transaction .....	95
Rembourrage des données de transaction .....	95
Exemples .....	96
Générer et vérifier MAC .....	97
Générer un MAC .....	98
Vérifiez le MAC .....	99
Types de clés pour des opérations de données spécifiques .....	100
GenerateCardDonnées .....	101
VerifyCardDonnées .....	102
GeneratePinData (pour les programmes VISA/ABA) .....	103
GeneratePinData (pourIBM3624) .....	104

VerifyPinData (pour les programmes VISA/ABA) .....	105
VerifyPinData (pourIBM3624) .....	106
Déchiffrer des données .....	107
Encrypt Data .....	108
Translate Pin Data .....	110
Générer/vérifier un MAC .....	111
VerifyAuthRequestCryptogram .....	112
Clé d'Import/Export .....	113
Types de clés non utilisés .....	113
Sécurité .....	115
Protection des données .....	116
Protection des éléments de clé .....	117
Chiffrement des données .....	117
Chiffrement au repos .....	118
Chiffrement en transit .....	118
Confidentialité du trafic inter-réseau .....	118
Résilience .....	119
Isolement régional .....	119
Conception à locataires multiples .....	120
Sécurité de l'infrastructure .....	121
Isolement des hôtes physiques .....	121
Utiliser Amazon VPC et AWS PrivateLink .....	121
Considérations relatives aux points de terminaison VPC de cryptographie des AWS paiements .....	122
Création d'un point de terminaison VPC pour AWS la cryptographie des paiements .....	123
Connexion à un point de terminaison VPC .....	124
Contrôle de l'accès à votre point de terminaison d'un VPC .....	125
Utilisation d'un point de terminaison VPC dans une déclaration de politique .....	129
Journalisation de votre point de terminaison d'un VPC .....	132
Bonnes pratiques de sécurité .....	134
Validation de la conformité .....	137
Gestion des identités et des accès .....	138
Public ciblé .....	138
Authentification par des identités .....	139
Compte AWS utilisateur root .....	140
Utilisateurs et groupes IAM .....	140

Rôles IAM .....	141
Gestion des accès à l'aide de politiques .....	142
Politiques basées sur l'identité .....	143
politiques basées sur les ressources .....	143
Listes de contrôle d'accès (ACL) .....	144
Autres types de politique .....	144
Plusieurs types de politique .....	145
Comment fonctionne la cryptographie des AWS paiements avec IAM .....	145
AWS Cryptographie des paiements Politiques basées sur l'identité .....	146
Autorisation basée sur les balises AWS de cryptographie des paiements .....	148
Exemples de politiques basées sur l'identité .....	148
Bonnes pratiques en matière de politiques .....	149
Utilisation de la console .....	150
Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations .....	150
Possibilité d'accéder à tous les aspects de la cryptographie des AWS paiements .....	152
Possibilité d'appeler des API à l'aide de touches spécifiées .....	152
Possibilité de refuser spécifiquement une ressource .....	153
Résolution des problèmes .....	154
Surveillance .....	155
CloudTrailJournaux .....	156
AWSInformations sur la cryptographie des paiements dans CloudTrail .....	156
Comprendre les AWS entrées du fichier journal de cryptographie des paiements .....	158
Détails cryptographiques .....	161
Objectifs de conception .....	162
Fondations .....	163
Primitives cryptographiques .....	163
Entropie et génération de nombres aléatoires .....	164
Opérations clés symétriques .....	164
Opérations clés asymétriques .....	164
Rangement des clés .....	164
Importation de clés à l'aide de clés symétriques .....	165
Importation de clés à l'aide de clés asymétriques .....	165
Exportation de clés .....	165
Protocole DUKPT (clé unique dérivée par transaction) .....	165
Hiérarchie des clés .....	166
Opérations internes .....	170

---

Spécifications et cycle de vie du HSM .....	170
Sécurité physique des appareils HSM .....	170
Initialisation du HSM .....	171
Service et réparation HSM .....	171
Mise hors service du HSM .....	172
Mise à jour du microprogramme HSM .....	172
Accès de l'opérateur .....	172
Gestion des clés .....	173
Opérations auprès des clients .....	180
Génération de clés .....	180
Importation de clés .....	181
Exportation de clés .....	182
Suppression des clés .....	182
Rotating keys .....	182
Quotas .....	183
Historique de la documentation .....	185
.....	clxxxvi

# Qu'est-ce que AWS Cryptographie des paiements ?

AWS La cryptographie des paiements est une solution gérée AWS service 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 est [Certifié PCI PTS HSM V3 et FIPS 140-2 niveau 3](#). Il est conçu pour supporter une faible latence [et des 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 le [AWS API du plan de contrôle de la cryptographie des paiements](#) pour créer et gérer des clés.

Vous pouvez utiliser le [AWS API du plan de données de cryptographie des paiements](#) pour utiliser des clés de chiffrement pour le traitement des transactions liées au paiement et les opérations cryptographiques associées.

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

- Création et gestion de systèmes symétriques et asymétriques AWS Les 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 votre AWS Clés 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 votre AWS Clés de cryptographie de paiement.
- Étiquetez votre AWS Clés 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 entre AWS Cryptographie 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) entre AWS Cryptographie 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 votre AWS Clés de cryptographie des paiements utilisées dans les opérations cryptographiques, telles que :

- Chiffrez, déchiffrez et chiffrez à nouveau les données de manière symétrique ou asymétrique AWS Clés 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/kwapwa6qaiif1lw2h
```

## 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 du secteur

### Rubriques

- [Types de clés courants](#)
- [Autres termes](#)

## 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.

## 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

## 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 par TR31\_P0\_PIN\_ENCRYPTION\_KEY.

## 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.

## 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.

### 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](#) (à l'origine 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 de la clé elle-même. 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.

## 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 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

La valeur de vérification du code PIN (PVV) est une valeur dérivée de manière algorithmique d'une série d'entrées telles que le [numéro de carte](#) et le code PIN qui génère une valeur pouvant être utilisée pour une validation ultérieure. L'un de ces systèmes est connu sous le nom de Visa PVV (également connu sous le nom de méthode ABA), bien qu'il soit utilisé pour les codes PIN sur n'importe quel réseau.

## 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](#)

AWS Service 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 votre AWS Clés KMS.

### [AWS CloudHSM](#)

AWS CloudHSM fournit aux clients des instances HSM à usage général dédiées dans AWS Nuage. AWS CloudHSM peut 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 dans AWS Cryptographie des paiements, voir [AWS Concepts de cryptographie des paiements](#).
- Pour plus d'informations sur AWS API du plan de contrôle de la cryptographie des paiements, voir [AWS Référence de l'API du plan de contrôle de la cryptographie des paiements](#).
- Pour plus d'informations sur AWS API Payment Cryptography Data Plane, voir [AWS Référence de l'API du plan de données de cryptographie des paiements](#).
- Pour des informations techniques détaillées sur la façon dont AWS La cryptographie des paiements utilise la cryptographie et sécurise AWS Clés de cryptographie de paiement, voir [Détails cryptographiques](#).

## Points de terminaison pour AWS Payment Cryptography

Pour vous connecter par programmation AWS Payment Cryptography, vous utilisez un point de terminaison, l'URL du point d'entrée du service. Les AWS SDK 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 demandes d'API.

## 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

## 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
USA Ouest (Oregon)	us-west-2	plan de données. payment-cryptography.us-west-2.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 les valeurs CVV2. [Pour essayer d'autres exemples et explorer les modèles de déploiement au sein d'AWS, 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.

## Rubriques

- [Prérequis](#)
- [Étape 1 : Création d'une clé](#)
- [Étape 2 : générer une valeur CVV2 à l'aide de la clé](#)
- [Étape 3 : Vérifiez la valeur générée à l'étape 2](#)
- [Étape 4 : Réaliser un test négatif](#)
- [Étape 5 : \(Facultatif\) Nettoyer](#)

## Prérequis

Avant de commencer, assurez-vous que :

- Vous êtes autorisé à accéder au service. Pour plus d'informations, consultez la section [Politiques IAM](#).
- Vous l'avez [AWS CLI](#) installé. Vous pouvez également utiliser [AWSdes SDK ou des AWS API](#) pour accéder à 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 [CVK](#) 3DES double longueur (2KEY TDES) pour générer et vérifier les valeurs CVV/CVV2.

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

La réponse renvoie les paramètres de la demande, y compris un ARN pour les appels suivants ainsi qu'une valeur de contrôle 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": "CADD1",  
    "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 valeur CVV2 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-identifiant arn:aws:payment-cryptography:us-east-2:111122223333:key/  
  tqv5yij6wtxx64pi \  
  --primary-account-number=171234567890123 \  
  --generation-attributes CardVerificationValue2={CardExpiryDate=0123}
```

```
{  
  "CardDataGenerationKeyCheckValue": "CADDA1",  
  "CardDataGenerationKeyIdentifiant": "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 CVV2 à partir de l'étape 2 à l'aide de la clé que vous avez créée à l'étape 1.

Exécutez la commande suivante pour valider le CVV2.

```
$ aws payment-cryptography-data verify-card-validation-data \  
  --key-identifiant 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 réponse HTTP de 200 pour indiquer qu'il a validé le CVV2.

## Étape 4 : Réaliser un test négatif

Dans cette étape, vous créez un test négatif dans lequel le CVV2 n'est pas correct et ne se valide pas. Vous essayez de valider un CVV2 incorrect à l'aide 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 saisi le mauvais CVV2 lors du paiement.

```
$ aws payment-cryptography-data verify-card-validation-data \
  --key-identifiant 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 réponse HTTP de 400 avec le message « Échec de la vérification des données de validation de la carte » et le motif INVALID\_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-identifiant=arn:aws:payment-cryptography:us-east-2:111122223333:key/
  tqv5yij6wtxx64pi
```

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

```
    "Exportable": true,
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
    tqv5yij6wtxx64pi",
    "KeyAttributes": {
      "KeyAlgorithm": "TDES_3KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyModesOfUse": {
        "Decrypt": true,
        "DeriveKey": false,
        "Encrypt": true,
        "Generate": false,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": true,
        "Verify": false,
        "Wrap": true
      },
      "KeyUsage": "TR31_P0_PIN_ENCRYPTION_KEY"
    },
    "KeyCheckValue": "CADD1",
    "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 défini 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](#)
- [Clés de liste](#)
- [Activation et désactivation des clés](#)
- [Suppression des clés](#)
- [Clés d'importation et d'exportation](#)
- [Utilisation des alias](#)
- [Obtenez des clés](#)
- [Clés de balisage](#)
- [Comprendre les principaux attributs de la clé AWS de cryptographie des paiements](#)

## Génération de clés

Vous pouvez créer des clés AWS de cryptographie de paiement à l'aide de l'opération CreateKey API. 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 (par exemple TR31\_P0\_PIN\_ENCRYPTION\_KEY), les opérations autorisées KeyUsage (par exemple, chiffrement, signature) et si elle est exportable. Vous ne pouvez pas modifier ces propriétés une fois la clé de chiffrement des AWS paiements créée.

## Exemples

- [Génération d'une clé TDES 2KEY](#)
- [Génération d'une clé de chiffrement par code PIN](#)

- [Création d'une clé asymétrique \(RSA\)](#)
- [Génération d'une clé de vérification du code PIN \(PVV\)](#)

## Génération d'une clé TDES 2KEY

### Exemple

Cette commande génère une clé TDES 2KEY dans le but de générer et de vérifier les valeurs CVV/CVV2. 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/
hjprdg5o4jtg5tw",
    "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

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

Cette commande génère une clé TDES à 3 touches dans le but de chiffrer les valeurs PIN (connue sous le nom de clé de cryptage PIN). Cette clé peut être utilisée pour sécuriser le stockage des codes PIN ou pour déchiffrer les codes PIN fournis 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)

### Exemple

Dans cet exemple, nous allons générer une nouvelle paire de clés asymétriques RSA 2048 bits. Une nouvelle clé privée sera générée ainsi que la clé publique correspondante. La clé publique peut être récupérée à l'aide de l'`PublicCertificateAPI` [get](#).

```

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

```

```

{
  "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 de vérification du code PIN (PVV)

### Exemple

Cette commande génère une clé TDES à 3 touches dans le but de générer des valeurs PVV (connues sous le nom de valeur de vérification des broches). Vous pouvez utiliser cette clé pour générer une valeur PVV qui peut être comparée à une valeur PVV calculée ultérieurement. 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.

### Exemple

```
$ 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

- [Commencer à utiliser les clés](#)
- [Arrêter l'utilisation des clés](#)

## Commencer à utiliser les 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.

### Exemple

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/alsuwxug3pgy6xh"
```

```
{
  "Key": {
    "CreateTimestamp": "2022-10-12T10:58:28.920000-07:00",
    "Enabled": true,
    "Exportable": true,
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/alsuwxug3pgy6xh",
    "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.

### Exemple

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 d'opérations cryptographiques telles que le CVV2 et n'est pas en mesure de déterminer si une clé est nécessaire pour un 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 CloudTrail](#) journal 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 la période 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.

## Exemple

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 le moment où deletePendingTimestamp la suppression de la clé est actuellement planifiée.

```
$ aws payment-cryptography delete-key \  
    --key-identifier arn:aws:payment-cryptography:us-  
east-2:111122223333:key/kwapwa6qaif1lw2h
```

```
{  
  "Key": {  
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
kwapwa6qaif1lw2h",  
    "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"
  }
}

```

## Exemple

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 la valeur `CREATE_COMPLETE`, tandis qu'il `DeletePendingTimestamp` est supprimé.

```

$ aws payment-cryptography restore-key --key-identifier arn:aws:payment-
cryptography:us-east-2:111122223333:key/kwapwa6qaif1lw2h

```

```

{
  "Key": {
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaif1lw2h",
    "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'autres HSM). 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 [ANSI X9.24 TR-34](#) bien établie. 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 conserver un HSM hors ligne 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 ([ANSI X9.24 TR 31-2018](#)) pour échanger des clés de travail. Le TR-31 suppose qu'un KEK a déjà été échangé. Cela est conforme à l'exigence du code PIN PCI 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, BDK, IPEK, etc.

### Rubriques

- [Clés d'importation](#)
- [Clés d'exportation](#)

## Clés d'importation

### Important

Les exemples peuvent nécessiter la dernière version de l'AWS CLI V2. Avant de commencer, assurez-vous d'avoir effectué la mise à niveau vers la [dernière version](#).

### Rubriques

- [Importation de clés symétriques](#)
- [Importation de clés asymétriques \(RSA\)](#)

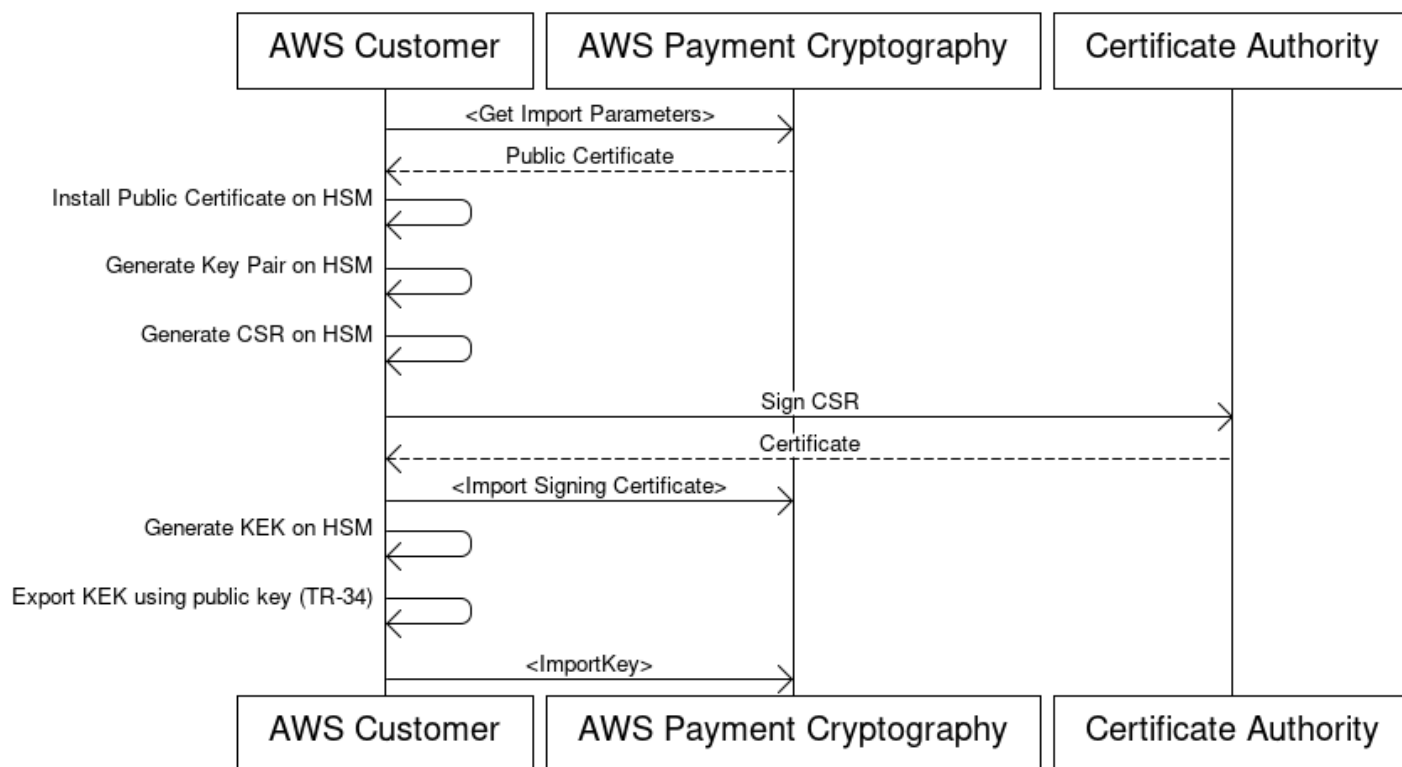
## Importation de clés symétriques

### Rubriques

- [Importation de clés à l'aide de techniques asymétriques \(TR-34\)](#)
- [Importation de clés à l'aide de techniques asymétriques \(RSA Unwrap\)](#)
- [Importation de clés symétriques à l'aide d'une clé d'échange de clés préétablie \(TR-31\)](#)

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

## Key Encryption Key(KEK) Import Process



Vue d'ensemble : Le TR-34 utilise le chiffrement asymétrique RSA 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. Cette 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 certificat KRD. 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

Avec de nombreux HSM, 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 de Key Distribution Host ou KDH). 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 HSM et systèmes associés 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 (chiffrement) et la clé de l'étape 3 en tant que certificat KDH (signature). Pour effectuer l'importation dans AWS Payment Cryptography, vous devez spécifier le format TR-34.2012 à deux passes non CMS, également appelé format TR-34 Diebold.

## 5. Clé d'importation d'appels

À la dernière étape, vous appellerez l'API `ImportKey` avec un `KeyMaterialType` de `TR34_KEY_BLOCK`. Il `certificate-authority-public-key-identifier` s'agira du `KeyARN` de l'autorité de certification racine importé à l'étape 3, du matériel clé de l'étape 4 `key-material` sera encapsulé et `signing-key-certificate` sera le certificat principal 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 les opérations cryptographiques.

## Importation de clés à l'aide de techniques asymétriques (RSA Unwrap)

Vue d'ensemble : La cryptographie des AWS paiements prend en charge l'encapsulation et le déballage RSA pour l'échange de clés lorsque le TR-34 n'est pas possible. Semblable au TR-34, cette technique utilise la cryptographie asymétrique RSA pour chiffrer les 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. En outre, cette technique d'encapsulation RSA ne préserve pas l'intégrité des métadonnées clés pendant le transfert car les blocs clés ne sont pas inclus.

### Note

L'encapsulation RSA peut être utilisée pour importer ou exporter des clés TDES et AES-128.

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

Appel `get-parameters-for-import` pour initialiser le processus d'importation avec un type de matériau clé `KEY_CRYPTOGRAM`. `WrappingKeyAlgorithm` peut être `RSA_2048` lors de l'échange de clés TDES. Le `RSA_3072` ou le `RSA_4096` peuvent être utilisés lors de l'échange de clés TDES ou AES-128. Cette 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

Avec de nombreux HSM, vous devrez peut-être installer/charger/approuver le certificat public (et/ou 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 HSM et systèmes associés permettent d'exporter des clés à l'aide de RSA Wrap. Vous devez spécifier la clé publique de l'étape 1 en tant que WrappingKey certificat (de cryptage). 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 votre HSM, vous devez spécifier le format RSA, 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 l'API ImportKey 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 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/kwapwa6qaiifllw2h",
    "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"
  }
}

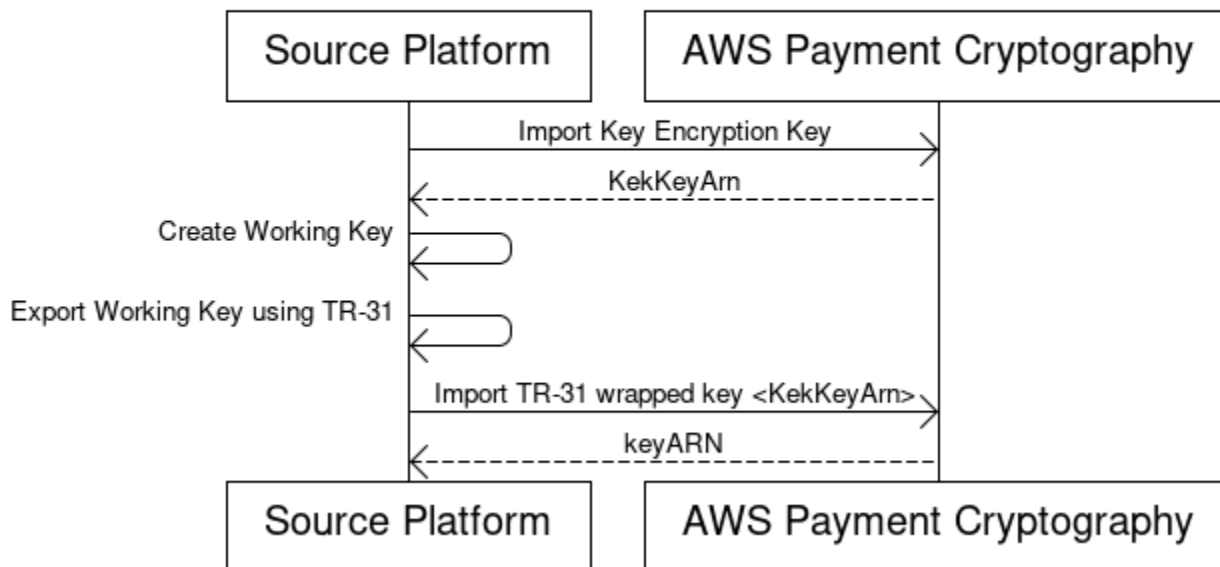
```

## 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 les opérations cryptographiques.

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

### Import symmetric keys using a pre-established key exchange key (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 qu'un KEK est établi, vous pouvez utiliser cette clé pour transporter les clés suivantes (y compris les autres KEK). AWS La cryptographie des paiements prend en charge ce type d'échange de clés à l'aide de la norme ANSI TR-31, largement utilisée et largement prise en charge par les fournisseurs de HSM.

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

On suppose que vous avez déjà importé votre KEK et que le KeyArn (ou KeyAlias) est 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 commande ImportKey, le KeyArn (ou alias) de votre clé de chiffrement WrappingKeyIdentifier doit être le résultat de la plate-forme source. WrappedKeyBlock

### Exemple

```
$ 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 publiques RSA

AWS La cryptographie des paiements prend en charge l'importation de clés RSA 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 format PEM et doit être codé en base64.

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

##### Exemple

```

$ 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":"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSURKVENDQWcyZ0F3SUJBZ01CWkR

```

```

{
  "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_ENCRYPTION` peut être utilisé lors du chiffrement de données destinées à être utilisées avec un autre système.

### Exemple

```

$ aws payment-cryptography import-key \
  --key-material='{"TrustedCertificatePublicKey":
{"CertificateAuthorityPublicKeyIdentifier":"arn:aws:payment-cryptography:us-
east-2:111122223333:key/zabouwe3574jysd1", \
  "KeyAttributes":
{"KeyAlgorithm":"RSA_2048", "KeyClass":"PUBLIC_KEY", "KeyModesOfUse":
{"Verify":true}, "KeyUsage":"TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"}, \
  "PublicKeyCertificate":"LS0tLS1CRUdJTjB..."}}'

```

```
{
```



```
"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 l'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 \(TR-34\)](#)
- [Exporter des clés à l'aide de techniques asymétriques \(RSA Wrap\)](#)
- [Exporter des clés symétriques à l'aide d'une clé d'échange de clés préétablie \(TR-31\)](#)
- [Exporter les clés initiales DUKPT \(IPEK/IK\)](#)

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

Vue d'ensemble : Le TR-34 utilise le chiffrement asymétrique RSA 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. Cette 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 signature KDH. Notez que ces certificats sont de courte durée et ne sont destinés qu'à cette fin. Le paramètre `ParametersValidUntilTimestamp` indique leur durée.

REMARQUE : Tous les certificats sont renvoyés dans un format codé en base64

#### Exemple

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

```
{
  "SigningKeyCertificate":
    "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJBZ01RZFAzSzNHNEFKT0I4WTNpTmUvY1
  "SigningKeyCertificateChain":
    "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUY0VENDQThZ0F3SUJBZ01SQUt1N2piaHFKZjJPd3FGUWI5c3
  "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 de Key Distribution Host ou KDH). Le destinataire (généralement votre HSM ou celui de vos partenaires) souhaitera générer une clé publique à cette fin, puis créer un certificat de clé publique (x.509) qui pourra être renvoyé à Payment Cryptography. AWS 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 l' `ExportKey` API avec un `KeyMaterialType` de `TR34_KEY_BLOCK`. Il `certificate-authority-public-key-identifier` s'agira du `KeyArn` de l'autorité de certification racine à l'étape 3, `WrappingKeyCertificate` du certificat feuille de l'étape 3 et `export-key-identifier` du `KeyArn` (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 (RSA Wrap)

Vue d'ensemble : La cryptographie des AWS paiements prend en charge l'encapsulation et le déballage RSA pour l'échange de clés lorsque le TR-34 n'est pas une option disponible par le partenaire. Semblable au TR-34, cette technique utilise la cryptographie asymétrique RSA pour chiffrer les 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. En outre, cette technique d'encapsulation RSA n'inclut pas les blocs clés utilisés pour préserver l'intégrité des métadonnées clés pendant le transport.

### Note

L'encapsulation RSA peut être utilisée pour exporter des clés TDES et AES-128.

### 1. Génération d'une clé et d'un certificat RSA sur le système de réception

Créez (ou identifiez) une clé RSA 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 l'ARN du 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": "LS0tLS1CRUdJTtBD...",
      "WrappingSpec": "RSA_OAEP_SHA_256"
    }
  }
}

```

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

```
{
  "WrappedKey": {
    "KeyMaterial":
"18874746731E9E1C4562E4116D1C2477063FCB08454D757D81854AEAE0A52B1F9D303FA29C02DC82AE7785353
    "WrappedKeyMaterialFormat": "KEY_CRYPTOGRAM"
  }
}
```

#### 4. Importer la clé dans le système de réception

De nombreux HSM et systèmes associés permettent d'importer des clés à l'aide de RSA unwrap (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, mode de remplissage = PKCS #1 v2.2 OAEP (avec SHA 256). La terminologie exacte peut varier selon le HSM.

##### Note

AWS Payment Cryptography génère la clé encapsulée dans 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 qu'un KEK est établi, vous pouvez utiliser cette clé pour transporter les clés suivantes (y compris d'autres KEK). AWS La cryptographie des paiements prend en charge ce type d'échange de clés à l'aide de la norme ANSI TR-31, largement utilisée et largement prise en charge par les fournisseurs de HSM.

##### 1. Clé de chiffrement Exchange Key (KEK)

On suppose que vous avez déjà échangé votre KEK et que vous avez le KeyArn (ou KeyAlias) 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 l'API, vous spécifiez la clé à exporter et la clé d'encapsulation 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/5rplquuwozodpwp
```

```
{
  "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 initiales DUKPT (IPEK/IK)

Lors de l'utilisation de [DUKPT](#), une seule clé de dérivation de base (BDK) peut être générée pour un parc de terminaux. Les terminaux n'ont toutefois jamais accès à ce BDK d'origine, mais reçoivent chacun une clé de terminal initiale unique appelée IPEK ou Initial Key (IK). Chaque IPEK est une clé dérivée du BDK et est destinée à être unique par terminal, mais elle est dérivée du BDK d'origine. Les données de dérivation utilisées pour ce calcul sont connues sous le nom de numéro de série clé (KSN). Selon X9.24, pour TDES, le KSN à 10 octets comprend généralement 24 bits pour l'ID du jeu de clés, 19 bits pour l'identifiant du terminal et 21 bits pour le compteur de transactions. Pour AES, le KSN à 12 octets comprend généralement 32 bits pour l'identifiant BDK, 32 bits pour l'identifiant de dérivation (ID) et 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 d'encapsulation TR-31, TR-34 et RSA. Les clés IPEK ne sont pas conservées et ne peuvent pas être utilisées pour des opérations ultérieures sur la cryptographie des paiements AWS

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

### Note

La partie compteur du KSN (32 bits pour AES DUKPT) n'est pas utilisée pour la dérivation IPEK/IK. Par conséquent, une entrée 12345678901234560001 et 12345678901234569999 produira le même IPEK.

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

## Exportation de clés asymétriques (RSA)

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

REMARQUE : Cette API n'est pas idempotente. Les appels suivants peuvent entraîner des certificats différents même si la clé sous-jacente est la même.



## Exemple

```
$ aws payment-cryptography get-public-key-certificate \  
    -key-identifiant 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/kwapwa6qai11w2h`.

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 chiffrement des paiements à l'aide de `CreateAliasAPI`.

L'API AWS de cryptographie des paiements fournit un contrôle total des alias dans chaque compte et région. L'API inclut des opérations permettant de créer un alias (CreateAlias), d'afficher les noms d'alias et le KeyARN lié (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](#)
- [API associées](#)

## 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 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 politique IAM, celle-ci fait référence à l'alias, et non à la clé de chiffrement des AWS paiements 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 `alias/sampleAlias1` 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/
kwapwa6qaif1lw2h"
    }
  ]
}
```

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/
kwapwa6qaif1lw2h"
    },
    {
      "AliasName": "alias/sampleAlias2",
      "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaif1lw2h"
    }
  ]
}
```

```
}
```

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/SampleAlias2 différent. 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` est associé à la clé de chiffrement des `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaiFl1w2h` 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 un ARN) 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/  
    kwapwa6qaif1lw2h
```

```
{  
  "Alias": {  
    "AliasName": "alias/alias/sampleAlias1",  
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
    kwapwa6qaif1lw2h"  
  }  
}
```

## 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 alias ARN.

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

Lorsque vous utilisez un alias ARN, n'oubliez pas que le mappage d'alias vers une clé de cryptographie de AWS paiement est défini dans le compte propriétaire de la clé de cryptographie de AWS paiement et peut varier 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 le nom de l'alias est sensible à la casse.

## API associées

### 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. L' GetKeys API prend une KeyIdentifier entrée et renvoie les attributs immuables et mutables de la clé mais ne contient aucun matériel cryptographique.

### Exemple

```
$ aws payment-cryptography get-key --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaif1lw2h
```

```
{
  "Key": {
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaif1lw2h",
    "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 résultat de l'API 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 appel d'API, même si la clé publique elle-même reste inchangée.

## Exemple

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

```
{
  "KeyCertificate":
  "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ01SQUo10Wd2VkpDd3d1Y1dMNldYZEpYY
  "KeyCertificateChain":
  "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUY0VENDQThZ0F3SUJBZ01SQUt1N2piaHFKZjJPd3FGUWI5c3VuO
}
```

## 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](#)
- [Afficher les tags clés dans la console](#)
- [Gestion des balises clés à l'aide des opérations d'API](#)
- [Contrôle de l'accès aux balises](#)
- [Utilisation de balises pour contrôler l'accès aux clés](#)



## 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'une 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 (BIN) spécifique.

```
[
  {
    "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 secret Amazon Elastic Block Store (Amazon EBS). AWS Secrets Manager 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, consultez [Autorisation 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 l'utilisateur IAM.

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 issue d'une politique IAM qui inclut la clé. Vous avez besoin de ces autorisations en plus des autorisations pour afficher les clés dans la console.

## Gestion des balises clés à l'aide des opérations d'API

Vous pouvez utiliser l'[API AWS Payment Cryptography](#) 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 : Obtenir 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'[CreateKey](#) opération.

Pour ajouter des balises lors de la création d'une clé, l'appelant doit disposer d'une `payment-cryptography:TagResource` autorisation dans le cadre d'une politique IAM. L'autorisation doit au minimum couvrir toutes les clés du compte et de la région. Pour plus de détails, veuillez consulter [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'opération [ListResourceTags](#).

## TagResource: ajouter ou modifier les balises d'une clé

L'[TagResource](#) opération 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/kwapwa6qaif1lw2h --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'opération [ListResourceTags](#).

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/kwapwa6qaif1lw2h \
  --tags '[{"Key":"Project","Value":"Noe"}]'
```

## ListResourceTags : Obtenir les tags d'une clé

L'opération [ListResourceTags](#) permet d'obtenir les balises d'une clé. Le paramètre ResourceArn (KeyArn ou KeyAlias) 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/kwapwa6qaif1lw2h
```

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

## UntagResource: Supprimer les tags d'une clé

L'[UntagResource](#) opération 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, utilisez l'opération [ListResourceTags](#).

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/
    kwapwa6qaif1lw2h --tag-keys Project
```

## Contrôle de l'accès aux balises

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

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 [TagResource](#) et [UntagResource](#).

Pour obtenir des exemples de politiques et plus d'informations, veuillez consulter [Contrôle de l'accès en fonction des clés de balises](#) dans le Guide de l'utilisateur IAM.

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 politique IAM 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 d'étiquetage dans une politique de clé ou une politique IAM. 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 principaux à baliser plusieurs clés, vous pouvez utiliser une politique IAM. Pour que cette politique soit efficace, la politique clé de chaque clé doit autoriser le compte à utiliser les politiques IAM pour contrôler l'accès à la clé.

Par exemple, la politique IAM 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'[CreateKey](#) opération 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 \(politiques IAM uniquement\)](#)
- [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 politique IAM 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 d'ensemble, veuillez consulter [Utiliser plusieurs clés et valeurs](#) dans le Guide de l'utilisateur IAM.

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



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

## 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 écrire une politique IAM qui permet aux principaux d'activer et de désactiver uniquement les clés dotées d'une balise particulière. Vous pouvez également utiliser une politique IAM 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 sert ABAC ? AWS](#) et [le contrôle de l'accès aux AWS ressources à l'aide de balises de ressources](#) dans le guide de l'utilisateur IAM.

 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 les politiques IAM. 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 [ListAliases](#)opérations [ListKeys](#)or.

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](#). Accordez aux responsables IAM 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. Les politiques IAM peuvent autoriser des

autorisations de balisage et de débaisage sur toutes les clés. Par exemple, la politique gérée par l'administrateur permet aux principaux de baliser, de débaisage 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.

Prenons l'exemple de la politique IAM 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": "IAMPolicyWithTag",
      "Effect": "Allow",
      "Action": [
        "payment-cryptography:DecryptData"
      ],
      "Resource": "arn:aws::us-east-1:111122223333:key/*",
      "Condition": {
        "StringEquals": {
          "aws:ResourceTag/Project": "Alpha"
        }
      }
    }
  ]
}
```

L'exemple de politique IAM 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

- TR31\_B0\_BASE\_DERIVATION\_KEY
  - Algorithmes clés autorisés : TDES\_2KEY, TDES\_3KEY, AES\_128, AES\_192, AES\_256
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31\_C0\_CARD\_VERIFICATION\_KEY
  - Algorithmes clés autorisés : TDES\_2KEY, TDES\_3KEY, AES\_128, AES\_192, AES\_256
  - 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\_3KEY, AES\_128, AES\_192, AES\_256
  - 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\_CRYPTGRAMS
  - Algorithmes clés autorisés : TDES\_2KEY, TDES\_3KEY, AES\_128, AES\_192, AES\_256

- Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31\_E1\_EMV\_MKEY\_CONFIDENTIALITY
  - Algorithmes clés autorisés : TDES\_2KEY, TDES\_3KEY, AES\_128, AES\_192, AES\_256
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31\_E2\_EMV\_MKEY\_INTEGRITY
  - Algorithmes clés autorisés : TDES\_2KEY, TDES\_3KEY, AES\_128, AES\_192, AES\_256
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31\_E4\_EMV\_MKEY\_DYNAMIC\_NUMBERS
  - Algorithmes clés autorisés : TDES\_2KEY, TDES\_3KEY, AES\_128, AES\_192, AES\_256
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31\_E5\_EMV\_MKEY\_CARD\_PERSONALIZATION
  - Algorithmes clés autorisés : TDES\_2KEY, TDES\_3KEY, AES\_128, AES\_192, AES\_256
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31\_E6\_EMV\_MKEY\_AUTRE
  - Algorithmes clés autorisés : TDES\_2KEY, TDES\_3KEY, AES\_128, AES\_192, AES\_256
  - Combinaison autorisée des principaux modes d'utilisation : { DeriveKey = true}, { NoRestrictions = true}
- TR31\_K0\_KEY\_ENCRYPTION\_KEY
  - Algorithmes clés autorisés : TDES\_2KEY, TDES\_3KEY, AES\_128, AES\_192, AES\_256
  - 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\_3KEY, AES\_128, AES\_192, AES\_256
  - 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\_3KEY
  - 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\_3KEY
  - 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\_3KEY, AES\_128, AES\_192, AES\_256
  - 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\_3KEY, AES\_128, AES\_192, AES\_256
  - Combinaison autorisée des principaux modes d'utilisation : {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}
- TR31\_P0\_PIN\_ENCRYPTION\_KEY
  - Algorithmes clés autorisés : TDES\_2KEY, TDES\_3KEY, AES\_128, AES\_192, AES\_256
  - 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\_3KEY, AES\_128, AES\_192, AES\_256
  - 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\_3KEY, AES\_128, AES\_192, AES\_256
  - Combinaison autorisée des principaux modes d'utilisation : {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, { NoRestrictions = true}

## Clés asymétriques

- TR31\_D1\_CLÉ\_ASYMÉTRIQUE POUR LE CHIFFREMENT DES DONNÉES
  - Algorithmes clés autorisés : RSA\_2048, RSA\_3072, RSA\_4096
  - Combinaison autorisée de modes d'utilisation clés : {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}
  - REMARQUE : {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\_CLÉ\_ASYMÉTRIQUE POUR SIGNATURE NUMÉRIQUE
  - Algorithmes clés autorisés : RSA\_2048, RSA\_3072, RSA\_4096
  - Combinaison autorisée des principaux modes d'utilisation : {Sign = true}, {Verify = true}
  - REMARQUE : {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](#)
- [Générer et vérifier les données de la carte](#)
- [Générez, traduisez et vérifiez les données PIN](#)
- [Cryptogramme de demande d'authentification \(ARQC\)](#)
- [Générer et vérifier MAC](#)
- [Clés valides pour les opérations cryptographiques](#)

## 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 remboursement	RSA_2048	RSA_3072	RSA_4096
OAEP SHA1	428	684	940

Type de remboursement	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'[CreateKey](#) opération ou importées à l'aide de l'[ImportKey](#) opération.

### Exemple

Dans cet exemple, nous chiffrerons les données en texte brut à l'aide d'une clé symétrique créée à l'aide de l'[CreateKey](#) opération 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-identifiant arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi --plain-text 31323334313233343132333431323334 --encryption-attributes 'Symmetric={Mode=CBC}'
```

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

## Chiffrer les données à l'aide de la clé DUKPT

### Exemple

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

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

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

### Exemple

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

### Exemple

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 31323334313233343132333431323334 --encryption-attributes
'Asymmetric={PaddingType=0AEP_SHA256}'
```

```
{
  "CipherText":
    "12DF6A2F64CC566D124900D68E8AFEEA794CA819876E258564D525001D00AC93047A83FB13 \
    E73F06329A100704FA484A15A49F06A7A2E55A241D276491AA91F6D2D8590C60CDE57A642BC64A897F4832A3930
    \
    0FAEC7981102CA0F7370BFBF757F271EF0BB2516007AB111060A9633D1736A9158042D30C5AE11F8C5473EC70F067
    \
    72590DEA1638E2B41FAE6FB1662258596072B13F8E2F62F5D9FAF92C12BB70F42F2ECDCF56AADF0E311D4118FE3591
    \
    FB672998CCE9D00FFFE05D2CD154E3120C5443C8CF9131C7A6A6C05F5723B8F5C07A4003A5A6173E1B425E2B5E42AD
    \
    7A2966734309387C9938B029AFB20828ACFC6D00CD1539234A4A8D9B94CDD4F23A",
  "KeyArn": "arn:aws:payment-cryptography:us-east-1:529027455495:key/5dza7xqd6soanjtb",
  "KeyCheckValue": "FF9DE9CE"
}
```

## Déchiffrer des données

[L'Decrypt DataAPI 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, 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

### Exemple

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.

## Exemple

Dans cet exemple, nous allons déchiffrer les données chiffrées à l'aide d'une clé [DUKPT](#) créée à l'aide de l'[CreateKey](#) opération 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 algorithm, 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

### Exemple

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'[CreateKey](#) opération. [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

### Exemple

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

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

# 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](#)
- [Vérifier les données de la carte](#)

## Générer des données de carte

L'Generate Card DataAPI 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 V8 utilisent le même algorithme cryptographique mais font varier les valeurs d'entrée. Par exemple, [CardVerificationValue1](#) contient des entrées telles que le numéro de ServiceCode carte et la date d'expiration. Bien que [CardVerificationValue2](#) ne possède que deux de ces entrées, cela est dû au fait que pour CVV2/CVC2, elles sont fixées à 000. ServiceCode 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 MMY Y ou YYMM) pour la génération et la validation afin de produire des résultats corrects.

## Générer un CVV2

### Exemple

Dans cet exemple, nous allons générer un CVV2 pour un PAN donné avec des entrées indiquant la date d'expiration de la carte [PAN](#) et la 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-
identifiant 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": "CADD1",
  "ValidationData": "801"
}
```

## Générer iCVV

### Exemple

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-
identifiant 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": "CADD1",
  "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 [VerifyCardValidationData](#) le 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

#### Exemple

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

### Exemple

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 999 [PAN](#), 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 generate-card-validation-data --key-
identifiant 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"
}
```

## 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

### Exemple

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-identif ier arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt --outgoing-key-identif ier 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

### Exemple

[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-identif ier arn:aws:payment-cryptography:us-east-2:111122223333:key/
ivi5ksfsuplneuyt --incoming-key-identif ier 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

#### Exemple

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 sont [PAN](#), 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

- [Générer un décalage de broche IBM3624 pour une broche](#)

### Générer un décalage de broche IBM3624 pour une broche

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

### Exemple

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-identifiant
arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2
--encryption-key-identifiant 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

#### Exemple

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

### Exemple

```

$ aws payment-cryptography-data generate-pin-data --generation-key-identifiant
arn:aws:payment-cryptography:us-east-2:111122223333:key/37y2tsl45p5zjbh2
--encryption-key-identifiant 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 ARQC 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](#)
- [Remboursement 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 remboursé :

00000000000000000000000008400080008000084016051700000000093800000B03011203 (74 caractères ou 37 octets)

Remboursé :

00000000000000000000000008400080008000084016051700000000093800000B03011203 000000 (80 caractères ou 40 octets)

### Rembourrage ISO 9797 Méthode 2

Non remboursé :

00000000000000000000000008400080008000084016051700000000093800000B1F220103000000 (80 caractères ou 40 octets)

Rembourné :

00000000000000000000000008400080008000084016051700000000093800000B1F220103000000  
8000000000000000 (88 caractères ou 44 octets)

## Exemples

### Visa CVN 10

#### Exemple

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-identifiant arn:aws:payment-cryptography:us-east-2:111122223333:key/  
pw3s6n162t5ushfk \  
--major-key-derivation-mode EMV_OPTION_A \  
--transaction-data  
0000000017000000000000000008400080008000084016051700000000093800000B03011203000000 \  
--session-key-derivation-attributes='{"Visa":{"PanSequenceNumber":"01" \  
, "PrimaryAccountNumber":"9137631040001422"}}'
```

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6n162t5ushfk",  
  "KeyCheckValue": "08D7B4"  
}
```

### Visa CVN18 et Visa CVN22

#### Exemple

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-identifiant arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6n162t5ushfk \
--major-key-derivation-mode EMV_OPTION_A
--transaction-data
0000000017000000000000000000008400080008000084016051700000000093800000B1F220103000000000000
\
00000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000000
--session-key-derivation-attributes='{"EmvCommon":
{"ApplicationTransactionCounter":"000B", \
"PanSequenceNumber":"01","PrimaryAccountNumber":"9137631040001422"}}'
```

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/pw3s6n162t5ushfk",
  "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.

AWS La 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](#)
- [Vérifiez le MAC](#)

## 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 surGenerate. La clé MAC peut être créée avec AWS Payment Cryptography en appelant [CreateKey](#) ou importée en appelant [ImportKey](#).

### Exemple

```
$ aws payment-cryptography-data generate-mac \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/  
qnobl5lghrzunce6 \  
  --message-data  
  "3b313038383439303031303733393431353d32343038323236303030373030303f33" \  
  --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 [CreateKey](#) ou 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 renvoyéHttp/200, sinon Http/400 avec un message l'indiquantMac 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 surVerify.

### Exemple

```
$ aws payment-cryptography-data verify-mac \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/  
qnobl5lghrzunce6 \  
  --message-data  
  "3b313038383439303031303733393431353d32343038323236303030373030303f33" \  
  --generation-attributes Algorithm=HMAC_SHA256
```

```
--key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
qno151ghrzunce6 \
--message-data
"3b343038383439303031303733393431353d32343038323236303030373030303f33" \
--verification-attributes='Algorithm=HMAC_SHA256' \
--mac ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C
```

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
qno151ghrzunce6,
  "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	<ul style="list-style-type: none"> <li>• AMEX_CARD_SECURITY_CODE_VERSION_1</li> <li>• AMEX_CARD_SECURITY_CODE_VERSION_2</li> </ul>	TR31_C0_CARD_VERIFICATION_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• TDES_3KEY</li> </ul>	{Générer = vrai}, {Générer = vrai, Vérifier = vrai}
GenerateCardDonnées	<ul style="list-style-type: none"> <li>• VALEUR_VÉRIFICATION_1 DE LA CARTE_1</li> <li>• VALEUR_VÉRIFICATION_2 DE LA CARTE_2</li> </ul>	TR31_C0_CARD_VERIFICATION_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> </ul>	{Générer = vrai}, {Générer = vrai, Vérifier = vrai}
GenerateCardDonnées	<ul style="list-style-type: none"> <li>• VALEUR_AUTHENTIFICATION_VÉRIFICATION_DU_TITUL</li> </ul>	TR31_E6_EMV_MKEY_AUTHRE	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> </ul>	{ 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	<ul style="list-style-type: none"> <li>CODE DE VÉRIFICATION DE LA CARTE DYNAMIQUE</li> </ul>	TR31_E4_E MV_MKEY_DYNAMIC_NUMBERS	<ul style="list-style-type: none"> <li>TDES_2KEY</li> </ul>	{ DeriveKey = vrai}
GenerateCardDonnées	<ul style="list-style-type: none"> <li>VALEUR_VÉRIFICATION_DYNAMIQUE DE LA CARTE</li> </ul>	TR31_E6_E MV_MKEY_AUTRE	<ul style="list-style-type: none"> <li>TDES_2KEY</li> </ul>	{ DeriveKey = vrai}

## VerifyCardDonnées

Opération ou algorithme cryptographique	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
<ul style="list-style-type: none"> <li>AMEX_CARD_SECURITY_CODE_VERSION_1</li> <li>AMEX_CARD_SECURITY_CODE_VERSION_2</li> </ul>	TR31_C0_CARD_VERIFICATION_KEY	<ul style="list-style-type: none"> <li>TDES_2KEY</li> <li>TDES_3KEY</li> </ul>	{Générer = vrai}, {Générer = vrai, Vérifier = vrai}

Opération ou algorithme cryptographique	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
<ul style="list-style-type: none"> <li>VALEUR_VÉRIFICATION_1 DE LA CARTE_1</li> <li>VALEUR_VÉRIFICATION_2 DE LA CARTE_2</li> </ul>	TR31_C0_CARD_VERIFICATION_KEY	<ul style="list-style-type: none"> <li>TDES_2KEY</li> </ul>	{Générer = vrai}, {Générer = vrai, Vérifier = vrai}
<ul style="list-style-type: none"> <li>VALEUR_AUTHENTIFICATION_VÉRIFICATION_DU_TITULAIRE DE LA CARTE</li> </ul>	TR31_E6_E MV_MKEY_AUTRE	<ul style="list-style-type: none"> <li>TDES_2KEY</li> </ul>	{ DeriveKey = vrai}
<ul style="list-style-type: none"> <li>CODE DE VÉRIFICATION DE LA CARTE DYNAMIQUE</li> </ul>	TR31_E4_E MV_MKEY_DYNAMIC_NUMBERS	<ul style="list-style-type: none"> <li>TDES_2KEY</li> </ul>	{ DeriveKey = vrai}
<ul style="list-style-type: none"> <li>VALEUR_VÉRIFICATION_DYNAMIQUE DE LA CARTE</li> </ul>	TR31_E6_E MV_MKEY_AUTRE	<ul style="list-style-type: none"> <li>TDES_2KEY</li> </ul>	{ 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_P IN_ENCRYPT TION_KEY	<ul style="list-style-type: none"> <li>TDES_2KEY</li> <li>TDES_3KEY</li> </ul>	<ul style="list-style-type: none"> <li>{Chiffrer = vrai, envelopper = vrai}</li> <li>{Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, débiller = vrai}</li> <li>{ NoRestrictions = vrai}</li> </ul>
Clé de génération de code PIN	TR31_V2_V ISA_PIN_VERIFICATI ON_KEY	<ul style="list-style-type: none"> <li>TDES_3KEY</li> </ul>	<ul style="list-style-type: none"> <li>{Générer = vrai}</li> <li>{Générer = vrai, vérifier = vrai}</li> </ul>

## 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_P IN_ENCRYPT TION_KEY	<ul style="list-style-type: none"> <li>TDES_2KEY</li> <li>TDES_3KEY</li> </ul>	Pour IBM3624_N ATURAL_PI N, IBM3624_R ANDOM_PIN , IBM3624_P IN_FROM_OFFSET

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
			<ul style="list-style-type: none"> <li>• {Chiffrer = vrai, envelopper = vrai}</li> <li>• {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai}</li> <li>• { NoRestrictions = vrai}</li> </ul> <p>Pour IBM3624_P IN_OFFSET</p> <ul style="list-style-type: none"> <li>• {Chiffrer = vrai, Décompresser = vrai}</li> <li>• {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai}</li> <li>• { NoRestrictions = vrai}</li> </ul>
Clé de génération de code PIN	TR31_V1_I BM3624_PI N_VERIFIC ATION_KEY	• TDES_3KEY	<ul style="list-style-type: none"> <li>• {Générer = vrai}</li> <li>• {Générer = vrai, vérifier = vrai}</li> </ul>

## 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_P IN_ENCRYPT TION_KEY	<ul style="list-style-type: none"> <li>TDES_2KEY</li> <li>TDES_3KEY</li> </ul>	<ul style="list-style-type: none"> <li>{Déchiffrer = vrai, Décompresser = vrai}</li> <li>{Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai}</li> <li>{ NoRestrictions = vrai}</li> </ul>
Clé de génération de code PIN	TR31_V2_V ISA_PIN_VERIFICATI ON_KEY	<ul style="list-style-type: none"> <li>TDES_3KEY</li> </ul>	<ul style="list-style-type: none"> <li>{Vérifier = vrai}</li> <li>{Générer = vrai, vérifier = vrai}</li> </ul>

## 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_P IN_ENCRYPT TION_KEY	<ul style="list-style-type: none"> <li>TDES_2KEY</li> <li>TDES_3KEY</li> </ul>	Pour IBM3624_N ATURAL_PI N, IBM3624_R ANDOM_PIN , IBM3624_P IN_FROM_OFFSET

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
			<ul style="list-style-type: none"> <li>• {Déchiffrer = vrai, Décompresser = vrai}</li> <li>• {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai}</li> <li>• { NoRestrictions = vrai}</li> </ul>
Clé de vérification du code PIN	TR31_V1_I BM3624_PI N_VERIFIC ATION_KEY	<ul style="list-style-type: none"> <li>• TDES_3KEY</li> </ul>	<ul style="list-style-type: none"> <li>• {Vérifier = vrai}</li> <li>• {Générer = vrai, vérifier = vrai}</li> </ul>

## Déchiffrer des données

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	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• AES_128</li> <li>• AES_192</li> <li>• AES_256</li> </ul>	<ul style="list-style-type: none"> <li>• { DeriveKey = vrai}</li> <li>• { NoRestrictions = vrai}</li> </ul>
EMV	TR31_E1_E MV_MKEY_C ONFIDENTIALITY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> </ul>	<ul style="list-style-type: none"> <li>• { DeriveKey = vrai}</li> </ul>

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	<ul style="list-style-type: none"> <li>• RSA_2048</li> <li>• RSA_3072</li> <li>• RSA_4096</li> </ul>	<ul style="list-style-type: none"> <li>• {Déchiffrer = vrai, unwrap=vrai}</li> <li>• {Encrypt=True, Wrap=True, Decrypt = true, Unwrap=True}</li> </ul>
Clés symétriques	TR31_D0_S YMMETRIC_ DATA_ENCR YPTION_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• TDES_3KEY</li> <li>• AES_128</li> <li>• AES_192</li> <li>• AES_256</li> </ul>	<ul style="list-style-type: none"> <li>• {Déchiffrer = vrai, unwrap=vrai}</li> <li>• {Encrypt=True, Wrap=True, Decrypt = true, Unwrap=True}</li> <li>• { NoRestrictions = vrai}</li> </ul>

## 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	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• AES_128</li> <li>• AES_192</li> <li>• AES_256</li> </ul>	<ul style="list-style-type: none"> <li>• { DeriveKey = vrai}</li> <li>• { NoRestrictions = vrai}</li> </ul>

Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
EMV	TR31_E1_E MV_MKEY_C CONFIDENTIALITY  TR31_E6_E MV_MKEY_AUTRE	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> </ul>	<ul style="list-style-type: none"> <li>• { DeriveKey = vrai }</li> </ul>
RSA	TR31_D1_C LÉ_ASYMÉTRIQUE POUR LE CHIFFREMENT DES DONNÉES	<ul style="list-style-type: none"> <li>• RSA_2048</li> <li>• RSA_3072</li> <li>• RSA_4096</li> </ul>	<ul style="list-style-type: none"> <li>• {Chiffrer = vrai, wrap=vrai}</li> <li>• {Encrypt=True, Wrap=True, Decrypt = true, Unwrap=True}</li> </ul>
Clés symétriques	TR31_D0_S SYMMETRIC_ DATA_ENCRYPTION_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• TDES_3KEY</li> <li>• AES_128</li> <li>• AES_192</li> <li>• AES_256</li> </ul>	<ul style="list-style-type: none"> <li>• {Chiffrer = vrai, wrap=vrai}</li> <li>• {Encrypt=True, Wrap=True, Decrypt = true, Unwrap=True}</li> <li>• { NoRestrictions = vrai }</li> </ul>

## 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_B ASE_DERIVATION_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• AES_128</li> <li>• AES_192</li> <li>• AES_256</li> </ul>	<ul style="list-style-type: none"> <li>• { DeriveKey = vrai }</li> <li>• { NoRestrictions = vrai }</li> </ul>
Source de données entrante	Non dopé (PEK, AWK, IWK, etc.)	TR31_P0_P IN_ENCRYPTION_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• TDES_3KEY</li> <li>• AES_128</li> <li>• AES_192</li> <li>• AES_256</li> </ul>	<ul style="list-style-type: none"> <li>• { Déchiffrer = vrai, Décompresser = vrai }</li> <li>• { Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai }</li> <li>• { NoRestrictions = vrai }</li> </ul>
Cible de données sortantes	DUKPT	TR31_B0_B ASE_DERIVATION_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• AES_128</li> <li>• AES_192</li> <li>• AES_256</li> </ul>	<ul style="list-style-type: none"> <li>• { DeriveKey = vrai }</li> <li>• { NoRestrictions = vrai }</li> </ul>
Cible de données sortantes	Non dopé (PEK, IWK, AWK, etc.)	TR31_P0_P IN_ENCRYPTION_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• TDES_3KEY</li> <li>• AES_128</li> <li>• AES_192</li> </ul>	<ul style="list-style-type: none"> <li>• { Chiffrer = vrai, envelopper = vrai }</li> </ul>

Direction	Type de clé	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
			<ul style="list-style-type: none"> <li>AES_256</li> </ul>	<ul style="list-style-type: none"> <li>{Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, débiller = vrai}</li> <li>{NoRestrictions = vrai}</li> </ul>

## 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_I SO_9797_1 _MAC_KEY	<ul style="list-style-type: none"> <li>TDES_2KEY</li> <li>TDES_3KEY</li> </ul>	<ul style="list-style-type: none"> <li>{Générer = vrai}</li> <li>{Générer = vrai, vérifier = vrai}</li> <li>{Vérifier = vrai}</li> <li>{Générer = vrai}</li> </ul>
Clé MAC (MAC pour le commerce de détail)	TR31_M1_I SO_9797_3 _MAC_KEY	<ul style="list-style-type: none"> <li>TDES_2KEY</li> <li>TDES_3KEY</li> </ul>	<ul style="list-style-type: none"> <li>{Générer = vrai}</li> </ul>

Utilisation autorisée des clés	Utilisation autorisée des clés	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
			<ul style="list-style-type: none"> <li>• {Générer = vrai, vérifier = vrai}</li> <li>• {Vérifier = vrai}</li> <li>• {Générer = vrai}</li> </ul>
Clé MAC (CMAC)	TR31_M6_I SO_9797_5 _CMAC_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• TDES_3KEY</li> <li>• AES_128</li> <li>• AES_192</li> <li>• AES_256</li> </ul>	<ul style="list-style-type: none"> <li>• {Générer = vrai}</li> <li>• {Générer = vrai, vérifier = vrai}</li> <li>• {Vérifier = vrai}</li> <li>• {Générer = vrai}</li> </ul>
Clé MAC (HMAC)	TR31_M7_H MAC_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• TDES_3KEY</li> <li>• AES_128</li> <li>• AES_192</li> <li>• AES_256</li> </ul>	<ul style="list-style-type: none"> <li>• {Générer = vrai}</li> <li>• {Générer = vrai, vérifier = vrai}</li> <li>• {Vérifier = vrai}</li> <li>• {Générer = vrai}</li> </ul>

## VerifyAuthRequestCryptogram

Utilisation autorisée des clés	Option EMV	Algorithme clé autorisé	Combinaison autorisée des principaux modes d'utilisation
<ul style="list-style-type: none"> <li>• OPTION A</li> <li>• OPTION B</li> </ul>	TR31_E0_E MV_MKEY_A PP_CRYPTOGRAMS	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> </ul>	<ul style="list-style-type: none"> <li>• { DeriveKey = vrai}</li> </ul>

## 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_K EY_BLOCK_ PROTECTION_KEY  TR31_K0_K EY_ENCRYP TION_KEY	<ul style="list-style-type: none"> <li>• TDES_2KEY</li> <li>• TDES_3KEY</li> <li>• AES_128</li> </ul>	<ul style="list-style-type: none"> <li>• {Encrypt = true, Wrap = true} (exportation uniquement)</li> <li>• {Decrypt = true, Unwrap = true} (importation uniquement)</li> <li>• {Chiffrer = vrai, Déchiffrer = vrai, envelopper = vrai, déballer = vrai}</li> </ul>
Importation d'une autorité de certification approuvée	TR31_S0_C LÉ_ASYMÉT RIQUE_POU R_SIGNATURE NUMÉRIQUE	<ul style="list-style-type: none"> <li>• RSA_2048</li> <li>• RSA_3072</li> <li>• RSA_4096</li> </ul>	<ul style="list-style-type: none"> <li>• {Vérifier = vrai}</li> </ul>
Importation d'un certificat à clé publique pour le chiffrement asymétrique	TR31_D1_C LÉ_ASYMÉT RIQUE POUR LE CHIFFREMENT DES DONNÉES	<ul style="list-style-type: none"> <li>• RSA_2048</li> <li>• RSA_3072</li> <li>• RSA_4096</li> </ul>	<ul style="list-style-type: none"> <li>• {Encrypt=TRUE, WRAP=TRUE}</li> </ul>

## 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\_ASYMMETRIC\_FOR\_KEY\_AGREEMENT

# 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, consultez la section [Services AWS concernés par programme de conformité](#) .
- 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](#)
- [Résilience dans la cryptographie des AWS paiements](#)
- [Sécurité de l'infrastructure dans AWS Payment Cryptography](#)
- [Connexion à la cryptographie des AWS paiements via un point de terminaison VPC](#)
- [Bonnes pratiques de sécurité pour la cryptographie des AWS paiements](#)

# 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 [Questions fréquentes \(FAQ\) sur la confidentialité des données](#). Pour en savoir plus sur la protection des données en Europe, consultez le billet de blog [Modèle de responsabilité partagée AWS et RGPD \(Règlement général sur la protection des données\)](#) sur le Blog de sécurité AWS .

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) avec chaque compte.
- Utilisez le protocole SSL/TLS pour communiquer avec les ressources. AWS Nous exigeons TLS 1.2 et recommandons TLS 1.3.
- Configurez l'API et la 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 modules cryptographiques validés par la norme FIPS 140-2 pour accéder AWS via une interface de ligne de commande ou une API, utilisez un point de terminaison FIPS. Pour plus d'informations sur les points de terminaison FIPS (Federal Information Processing Standard) disponibles, consultez [Federal Information Processing Standard \(FIPS\) 140-2](#) (Normes de traitement de l'information fédérale).

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 utilisez AWS Payment Cryptography ou une autre méthode à Services AWS l'aide de la console, de l'API ou des AWS SDK. AWS CLI 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 une adresse URL à un serveur externe, nous vous recommandons fortement de ne pas inclure d'informations d'identification dans l'adresse URL permettant de valider votre demande adressée à ce serveur.

AWS Payment Cryptography stocke et protège vos clés de chiffrement des paiements afin de les rendre hautement disponibles tout en vous fournissant un contrôle d'accès solide et flexible.

## Rubriques

- [Protection des éléments de clé](#)
- [Chiffrement des données](#)
- [Chiffrement au repos](#)
- [Chiffrement en transit](#)
- [Confidentialité du trafic inter-réseau](#)

## 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 les détails cryptographiques d'AWS Payment Cryptography.

## Chiffrement des données

Les données d'AWS Payment Cryptography comprennent les clés de chiffrement des paiements AWS, le contenu des clés de chiffrement qu'elles représentent et leurs attributs d'utilisation. Le contenu clé n'existe en texte clair que dans les modules matériels de sécurité (HSM) d'AWS Payment Cryptography 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.

Le contenu clé généré ou chargé par AWS Payment Cryptography pour les clés de paiement ne quitte jamais les limites des HSM d'AWS Payment Cryptography non chiffrés. Il peut être exporté chiffré par les opérations de l'API AWS Payment Cryptography.

## Chiffrement au repos

AWS Payment Cryptography génère des éléments clés pour les clés de paiement dans les HSM listés par PCI PTS HSM. Lorsqu'ils ne sont pas utilisés, les éléments de clé sont chiffrés par une clé HSM et écrits dans un stockage permanent et persistant. Le matériel clé pour les clés de cryptographie de paiement et les clés de chiffrement qui protègent le matériel clé ne quittent jamais les HSM 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 les détails cryptographiques d'AWS Key Management Service.

## Chiffrement en transit

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

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

Tous les appels de l'API AWS Payment Cryptography doivent être signés et transmis à l'aide du protocole TLS (Transport Layer Security). AWS Payment Cryptography nécessite des versions TLS et des suites de chiffrement définies par la norme PCI comme une « cryptographie forte ». Tous les points de terminaison de service prennent en charge le protocole TLS 1.0—1.3 et le protocole TLS post-quantique hybride.

Pour plus de détails, consultez les détails cryptographiques d'AWS Key Management Service.

## Confidentialité du trafic inter-réseau

AWS Payment Cryptography prend en charge une console de gestion AWS et un ensemble d'opérations d'API qui vous permettent de créer et de gérer des clés de paiement et de les utiliser dans des opérations cryptographiques.

AWS Payment Cryptography prend en charge deux options de connectivité réseau entre votre réseau privé et AWS.

- Une connexion VPN IPsec sur Internet.
- AWS Direct Connect, qui relie votre réseau interne à un site AWS Direct Connect via un câble Ethernet à fibre optique standard.

Tous les appels de l'API de cryptographie des paiements doivent être signés et transmis à l'aide du protocole TLS (Transport Layer Security). 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 (HSM) qui stockent les éléments clés pour les clés de paiement est autorisé uniquement à partir d'hôtes d'API AWS Payment Cryptography connus sur le réseau interne d'AWS.

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

AWS Payment Cryptography prend également en charge une option hybride d'échange de clés post-quantique pour le protocole de chiffrement réseau TLS (Transport Layer Security). Vous pouvez utiliser cette option avec le protocole TLS lorsque vous vous connectez aux points de terminaison de l'API AWS Payment Cryptography.

## 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

AWS Payment Cryptography est un service régional disponible dans plusieurs régions.

La conception isolée par région d'AWS Payment Cryptography garantit qu'un problème de disponibilité dans une région AWS ne peut affecter le fonctionnement d'AWS Payment Cryptography

dans aucune autre région. AWS Payment Cryptography est conçu 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 (SLA) d'AWS Payment Cryptography inclut un engagement de service de 99,99 % pour toutes les API de cryptographie des paiements. Pour respecter cet engagement, AWS Payment Cryptography garantit que toutes les données et informations d'autorisation requises pour exécuter une demande d'API sont disponibles sur tous les hôtes régionaux qui reçoivent la demande.

L'infrastructure de chiffrement des paiements AWS est répliquée dans au moins trois zones de disponibilité (AZ) dans chaque région. Pour garantir que les défaillances de plusieurs hôtes n'affectent pas les performances d'AWS Payment Cryptography, AWS Payment Cryptography est conçu pour gérer le trafic client en provenance de n'importe laquelle des AZ d'une 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 matériels de sécurité (HSM) AWS Payment Cryptography, chacun d'entre eux pouvant effectuer l'opération avec la clé de paiement.

## Conception à locataires multiples

La conception mutualisée d'AWS Payment Cryptography lui permet de respecter le SLA de disponibilité 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 dans le HSM dès sa création, et le matériel clé crypté est immédiatement transféré vers un stockage sécurisé. La clé chiffrée est récupérée et déchiffrée dans le HSM juste à temps pour être utilisée. La clé en texte clair reste dans la mémoire HSM uniquement pendant le temps nécessaire à la réalisation de l'opération cryptographique. Les éléments de clé en texte clair ne quittent jamais les HSM ; ils ne sont jamais écrits dans un stockage persistant.

Pour plus d'informations sur les mécanismes utilisés par AWS Payment Cryptography pour sécuriser vos clés, consultez les détails cryptographiques d'AWS Payment Cryptography.

## 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 des appels d'API AWS publiés pour accéder AWS Payment Cryptography via le réseau. Les clients doivent prendre en charge le protocole TLS (Transport Layer Security) 1.2 ou version ultérieure. Les clients doivent aussi prendre en charge les suites de chiffrement PFS (Perfect Forward Secrecy) comme Ephemeral Diffie-Hellman (DHE) ou Elliptic Curve Ephemeral Diffie-Hellman (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 ID de clé d'accès et d'une clé d'accès secrète associée à un principal IAM. 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.

La cryptographie des paiements AWS est prise en charge par des modules de sécurité matériels (HSM) dédiés certifiés commercial-off-the-shelf PCI PTS HSM. Le contenu clé des clés de cryptographie de paiement AWS est stocké uniquement dans la mémoire volatile des HSM, et uniquement lorsque la clé de cryptographie de paiement est utilisée. Les HSM se 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 des HSM de chiffrement des paiements AWS, consultez la section Détails du chiffrement des paiements AWS.

### Connexion à la cryptographie des AWS paiements via un point de terminaison VPC

Vous pouvez vous connecter directement à AWS Payment Cryptography via un point de terminaison d'interface privée dans votre cloud privé virtuel (VPC). Lorsque vous utilisez un point de terminaison



VPC d'interface, la communication entre votre VPC et AWS Payment Cryptography s'effectue entièrement au sein du réseau. AWS

AWS Payment Cryptography prend en charge les points de terminaison Amazon Virtual Private Cloud (Amazon VPC) alimentés par [AWS PrivateLink](#). Chaque point de terminaison d'un VPC est représenté par une ou plusieurs [interfaces réseau Elastic](#) (ENI) avec des adresses IP privées dans vos sous-réseaux VPC.

Le point de terminaison VPC de l'interface connecte votre VPC directement à AWS Payment Cryptography sans passerelle Internet, périphérique NAT, connexion VPN ou connexion AWS Direct Connect. Les instances de votre 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 points de terminaison VPC et les politiques des points de terminaison VPC dans toutes les régions AWS dans tous les cas où la cryptographie des paiements est prise [en charge](#).

## Rubriques

- [Considérations relatives aux points de terminaison VPC de cryptographie des AWS paiements](#)
- [Création d'un point de terminaison VPC pour AWS la cryptographie des paiements](#)
- [Connexion à un point de terminaison VPC de chiffrement des paiements](#)
- [Contrôle de l'accès à votre point de terminaison d'un VPC](#)
- [Utilisation d'un point de terminaison VPC dans une déclaration de politique](#)
- [Journalisation de votre point de terminaison d'un VPC](#)

## Considérations relatives aux points de terminaison VPC de cryptographie des AWS paiements

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

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

- Vous pouvez utiliser votre point de terminaison VPC pour appeler toutes les opérations du plan de [contrôle de cryptographie des AWS paiements et les opérations du plan](#) de données de [cryptographie des AWS paiements depuis](#) un VPC.
- Vous pouvez créer un point de terminaison VPC d'interface qui se connecte à un point de terminaison de région AWS de cryptographie des 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 point de terminaison VPC. Pour plus de détails, consultez [Journalisation de votre point de terminaison d'un VPC](#).

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

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

- Pour créer un point de terminaison VPC pour le chiffrement des AWS 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 point de terminaison de VPC, vous pouvez activer un [nom DNS privé](#) pour votre point de terminaison d'un VPC. Si vous sélectionnez l'option Activer le nom DNS,

le nom d'hôte DNS standard AWS de chiffrement des paiements correspond à votre point de terminaison VPC. Par exemple, `https://controlplane.payment-cryptography.us-west-2.amazonaws.com` serait résolu vers un point de terminaison d'un VPC connecté au nom du service `com.amazonaws.us-west-2.payment-cryptography.controlplane`.

Cette option facilite l'utilisation du point de terminaison d'un VPC. Les AWS SDK AWS CLI utilisent par défaut le nom d'hôte DNS standard de AWS Payment Cryptography. Vous n'avez donc pas besoin de spécifier l'URL du point de terminaison VPC 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 VPC de chiffrement des paiements

Vous pouvez vous connecter à AWS Payment Cryptography via le point de terminaison VPC à l'aide d'AWS un SDK, du ou. AWS CLI AWS Tools for PowerShell Pour spécifier le point de terminaison VPC, utilisez son nom DNS.

Par exemple, cette commande [list-keys](#) utilise le paramètre `endpoint-url` pour indiquer le point de terminaison VPC. Pour utiliser une commande comme celle-ci, remplacez l'exemple d'ID de point de terminaison VPC par celui de votre compte.

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

Si vous avez activé les noms d'hôte privés lorsque vous avez créé votre point de terminaison VPC, vous n'avez pas besoin de spécifier l'URL de point de terminaison VPC dans vos commandes de CLI ou dans la configuration de l'application. Le nom d'hôte DNS standard AWS de cryptographie des paiements correspond à votre point de terminaison VPC. Les SDK AWS CLI et utilisent ce nom d'hôte par défaut. Vous pouvez donc commencer à utiliser le point de terminaison VPC 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 attributs `enableDnsHostnames` et `enableDnsSupport` de votre VPC doivent avoir la valeur `true`. Pour définir ces attributs, utilisez l'opération [ModifyVpcAttribute](#). Pour plus d'informations, veuillez consulter [Afficher et mettre à jour les attributs DNS pour votre VPC](#) dans le Guide de l'utilisateur Amazon VPC.

## Contrôle de l'accès à votre point de terminaison d'un VPC

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

Vous pouvez créer une politique de point de terminaison VPC lorsque vous créez votre point de terminaison, et vous pouvez modifier la politique de point de terminaison d'un VPC à tout moment. Utilisez la console de gestion VPC ou les opérations [CreateVpcEndpoint](#) ou [ModifyVpcEndpoint](#). Vous pouvez également créer et modifier une politique de point de terminaison VPC à [l'aide d'un AWS CloudFormation modèle](#). Pour obtenir de l'aide sur l'utilisation de la console de gestion de VPC, veuillez consulter [Créer 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 de point de terminaison d'un VPC](#)
- [Politique de point de terminaison d'un VPC par défaut](#)
- [Création d'une stratégie de point de terminaison de VPC](#)
- [Affichage d'une politique de point de terminaison d'un VPC](#)

### À propos des politiques de point de terminaison d'un VPC

Pour qu'une demande AWS de cryptographie de paiement utilisant un point de terminaison VPC 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 point de terminaison d'un VPC doit accorder au principal l'autorisation d'utiliser le point de terminaison pour effectuer la demande.

Par exemple, une politique de clés peut autoriser un principal à appeler [Decrypt](#) sur une clé de cryptographie AWS de paiement particulière. Cependant, la politique du point de terminaison du VPC peut ne pas autoriser ce principal à faire appel à ces clés de cryptographie de AWS paiement en utilisant Decrypt le point de terminaison.

Une politique de point de terminaison VPC peut également autoriser un principal à utiliser le point de terminaison pour appeler [StopKeyUsage](#) sur certaines clés de cryptographie AWS de paiement. Mais si le principal ne dispose pas de ces autorisations dans le cadre d'une politique IAM, la demande échoue.

## Politique de point de terminaison d'un VPC par défaut

Chaque point de terminaison d'un VPC dispose d'une politique de point de terminaison d'un VPC, mais vous n'êtes pas tenu de spécifier la politique. 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 politique [IAM](#). 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 point de terminaison d'un VPC uniquement pour un sous-ensemble de leurs opérations autorisées, [créez ou mettez à jour la politique de point de terminaison d'un VPC](#).

## Création d'une stratégie de point de terminaison de VPC

Une politique de point de terminaison d'un VPC détermine si un principal a l'autorisation d'utiliser le point de terminaison d'un VPC pour effectuer des opérations sur une ressource. Pour les ressources de cryptographie des AWS paiements, le principal doit également être autorisé à effectuer les opérations conformément à une politique [IAM](#).

Chaque instruction de politique de point de terminaison d'un VPC 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.

L'instruction de politique ne spécifie pas le point de terminaison d'un VPC. Au lieu de cela, elle s'applique à tout point de terminaison d'un VPC auquel la politique est attachée. Pour plus d'informations, consultez [Contrôle de l'accès aux services avec points de terminaison d'un VPC](#) dans le Guide de l'utilisateur Amazon VPC.

Voici un exemple de politique de point de terminaison VPC pour la cryptographie des AWS paiements. Lorsqu'il est connecté à un point de terminaison VPC, cette politique permet d'ExampleUser d'utiliser le point de terminaison VPC pour appeler les opérations spécifiées sur les clés de cryptographie de paiement spécifiées AWS. 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/
        kwapwa6qaifl1lw2h"
    }
  ]
}
```

AWS CloudTrail enregistre toutes les opérations qui utilisent le point de terminaison VPC. 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 souhaitez peut-être créer une politique de point de terminaison VPC qui empêche les principaux titulaires de comptes externes d'utiliser le point de terminaison VPC pour appeler 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"
        }
      }
    }
  ]
}
```

## Affichage d'une politique de point de terminaison d'un VPC

[Pour consulter la politique de point de terminaison VPC d'un point de terminaison, utilisez la console de gestion VPC ou l'opération Endpoints. DescribeVpc](#)

La AWS CLI commande suivante obtient la politique du point de terminaison avec l'ID de point de terminaison VPC 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==`'].[PolicyDocument]'
--output text
```

## Utilisation d'un point de terminaison VPC dans une déclaration de politique

Vous pouvez contrôler l'accès aux ressources et aux opérations de cryptographie des AWS paiements lorsque la demande provient d'un VPC ou utilise un point de terminaison VPC. Pour ce faire, utilisez une politique [IAM](#)

- Utilisez la clé de condition `aws:sourceVpce` pour accorder ou restreindre l'accès en fonction du point de terminaison d'un VPC.
- Utilisez la clé de condition `aws:sourceVpc` pour accorder ou restreindre l'accès en fonction du VPC 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 [terminaison Amazon VPC](#). Pour restreindre les requêtes à un point de terminaison VPC, utilisez les clés de condition `aws:sourceVpce` ou `aws:sourceVpc`. Pour de plus amples informations, veuillez consulter [Gestion des identités et des accès pour les points de terminaison d'un VPC et les services de points de terminaison d'un VPC](#) dans le Guide AWS PrivateLink .

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 [CreateKey](#) ce type qui ne dépend 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 terminaison VPC 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'ID du point de terminaison VPC figurant dans la demande est comparé à la valeur de `aws:sourceVpce` la clé de 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 réservé et les identifiants de point de terminaison VPC 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": ""
      }
    }
  }
]
```

Vous pouvez également utiliser la clé de `aws:sourceVpce` condition pour restreindre l'accès à vos clés de cryptographie de AWS paiement en fonction du VPC dans lequel réside le point de terminaison du VPC.

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 politique comme celle-ci si une application est en cours d'exécution dans un VPC, mais que vous utilisez un second VPC isolé pour les fonctions de gestion.

Pour utiliser une politique comme celle-ci, remplacez l'identifiant de compte AWS réservé et les identifiants de point de terminaison VPC 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": "*",
      }
    ]
  }
}
```

## Journalisation de votre point de terminaison d'un VPC

AWS CloudTrail enregistre toutes les opérations qui utilisent le point de terminaison VPC. Lorsqu'une demande adressée à AWS Payment Cryptography utilise un point de terminaison VPC, l'ID du point de terminaison VPC apparaît dans [AWS CloudTrail l'entrée du journal qui enregistre la](#) demande. Vous pouvez utiliser l'identifiant du point de terminaison pour vérifier l'utilisation de votre point de terminaison VPC AWS Payment Cryptography.

Pour protéger votre VPC, les demandes refusées par une [politique de point de terminaison du VPC](#), mais qui auraient autrement été autorisées, ne sont pas enregistrées dans [AWS CloudTrail](#)

Par exemple, cet exemple d'entrée de journal enregistre une [GenerateMac](#) demande utilisant le point de terminaison VPC. Le champ `vpcEndpointId` apparaît à la fin de l'entrée de journal.

```
{
  "eventVersion": "1.08",
  "userIdentity": {
    "principalId": "TESTXECZ5U9M4LGF2N6Y5:",
    "arn": "arn:aws:sts::111122223333:assumed-role//",
    "accountId": "111122223333",
    "accessKeyId": "TESTXECZ5U2ZULLHJM",
    "sessionContext": {
      "sessionIssuer": {
        "type": "Role",
        "principalId": "TESTXECZ5U9M4LGF2N6Y5",
        "arn": "arn:aws:iam::111122223333:role/",
        "accountId": "111122223333",
        "userName": ""
      },
      "webIdFederationData": {},
      "attributes": {
        "creationDate": "2024-05-27T19:34:10Z",
        "mfaAuthenticated": "false"
      },
      "ec2RoleDelivery": "2.0"
    }
  },
}
```

```
"eventTime": "2024-05-27T19:49:54Z",
"eventSource": "payment-cryptography.amazonaws.com",
"eventName": "CreateKey",
"awsRegion": "us-east-1",
"sourceIPAddress": "172.31.85.253",
"userAgent": "aws-cli/2.14.5 Python/3.9.16 Linux/6.1.79-99.167.amzn2023.x86_64
source/x86_64.amzn.2023 prompt/off command/payment-cryptography.create-key",
"requestParameters": {
  "keyAttributes": {
    "keyUsage": "TR31_M1_ISO_9797_1_MAC_KEY",
    "keyClass": "SYMMETRIC_KEY",
    "keyAlgorithm": "TDES_2KEY",
    "keyModesOfUse": {
      "encrypt": false,
      "decrypt": false,
      "wrap": false,
      "unwrap": false,
      "generate": true,
      "sign": false,
      "verify": true,
      "deriveKey": false,
      "noRestrictions": false
    }
  },
  "exportable": true
},
"responseElements": {
  "key": {
    "keyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaiifllw2h",
    "keyAttributes": {
      "keyUsage": "TR31_M1_ISO_9797_1_MAC_KEY",
      "keyClass": "SYMMETRIC_KEY",
      "keyAlgorithm": "TDES_2KEY",
      "keyModesOfUse": {
        "encrypt": false,
        "decrypt": false,
        "wrap": false,
        "unwrap": false,
        "generate": true,
        "sign": false,
        "verify": true,
        "deriveKey": false,
        "noRestrictions": false
      }
    }
  }
}
```

```
    }
  },
  "keyCheckValue": "A486ED",
  "keyCheckValueAlgorithm": "ANSI_X9_24",
  "enabled": true,
  "exportable": true,
  "keyState": "CREATE_COMPLETE",
  "keyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
  "createTimestamp": "May 27, 2024, 7:49:54 PM",
  "usageStartTimestamp": "May 27, 2024, 7:49:54 PM"
}
},
"requestID": "f3020b3c-4e86-47f5-808f-14c7a4a99161",
"eventID": "b87c3d30-f3ab-4131-87e8-bc54cfef9d29",
"readOnly": false,
"eventType": "AwsApiCall",
"managementEvent": true,
"recipientAccountId": "111122223333",
"vpcEndpointId": "",
"eventCategory": "Management",
"tlsDetails": {
  "tlsVersion": "TLSv1.3",
  "cipherSuite": "TLS_AES_128_GCM_SHA256",
  "clientProvidedHostHeader": "-oo28vrivr.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 aux fins prévues, notamment des [politiques IAM](#), un ensemble complet de clés de conditions de politique pour affiner vos politiques clés et vos politiques IAM et l'application intégrée des règles PCI PIN 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 interopérables sécurisés ANSI X9 TR 31-2018 et conformément à l'exigence de sécurité PCI PIN 18-3. 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, comme Alias/bin\_12345\_CVK pour désigner une clé de vérification de carte associée au BIN 12345. 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 :** l'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. L'IAM peut également être utilisé pour limiter l'accès aux commandes et aux touches susceptibles de ne pas s'appliquer 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 de sécurité dans IAM](#) dans le Guide de l'utilisateur IAM

# Validation de la conformité pour AWS Cryptographie des paiements

Des auditeurs tiers évaluent la sécurité et la conformité des AWS Cryptographie des paiements dans le cadre de plusieurs AWS programmes de 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 (PCI PIN) et du chiffrement point à point (P2PE) PCI. Veuillez consulter AWS Artifact pour les attestations et les guides de conformité disponibles.

Pour obtenir la liste des services AWS relevant de programmes de conformité spécifiques, consultez les [Services AWS relevant de programmes de conformité](#). Pour obtenir des renseignements généraux, consultez [Programmes de conformité AWS](#).

Vous pouvez télécharger les rapports de l'audit externe avec AWS Artifact. Pour de plus amples informations, veuillez consulter [Téléchargement de rapports dans AWS Artifact](#).

Votre responsabilité en matière de conformité lors de l'utilisation AWS La cryptographie des paiements est déterminée par la sensibilité de vos données, les objectifs de conformité de votre entreprise et les 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](#)—Ces guides de déploiement abordent les considérations architecturales et fournissent les étapes à suivre pour déployer des environnements de base axés sur la sécurité et la conformité sur AWS.
- [AWS Ressources de conformité](#)—Cette collection de cahiers d'exercices et de guides peut s'appliquer à votre secteur d'activité et à votre région.
- [Évaluation des ressources à l'aide de règles](#) dans le AWS Config Guide du développeur—AWS Config; évalue dans quelle mesure vos configurations de ressources sont conformes aux pratiques internes, aux directives du secteur et aux réglementations.
- [AWS Security Hub](#)—Ceci AWS le service fournit une vue complète de l'état de votre sécurité dans AWS qui vous aide à 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. Les administrateurs IAM contrôlent qui peut être authentifié (connecté) et autorisé (autorisé) à utiliser les ressources de cryptographie des AWS paiements. IAM est un Service AWS outil que vous pouvez utiliser sans frais supplémentaires.

## Rubriques

- [Public ciblé](#)
- [Authentification par des identités](#)
- [Gestion des accès à l'aide de politiques](#)
- [Comment fonctionne la cryptographie des AWS paiements avec IAM](#)
- [AWS Exemples de politiques basées sur l'identité en matière de cryptographie des paiements](#)
- [Résolution des problèmes d'identité et d'accès liés à la cryptographie des AWS paiements](#)

## 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, consultez [Résolution 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 soumettre les demandes à votre administrateur IAM pour modifier les autorisations des utilisateurs de votre service. Consultez les informations sur cette page pour comprendre les concepts de base d'IAM. Pour en savoir plus sur la manière dont votre entreprise peut utiliser l'IAM avec le chiffrement des AWS paiements, consultez. [Comment fonctionne la cryptographie des AWS paiements avec IAM](#)

**Administrateur IAM** — Si vous êtes administrateur IAM, vous souhaitez 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 dans IAM, consultez. [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 qu'utilisateur IAM ou en assumant un rôle IAM. Utilisateur racine d'un compte AWS

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 (IAM Identity 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 avec une identité fédérée, votre administrateur aura précédemment configuré une fédération d'identités avec des rôles IAM. 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 [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 demandes AWS d'API](#) dans le guide de l'utilisateur IAM.

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 l'utilisateur AWS IAM Identity Center et [Utilisation de l'authentification multifactorielle \(MFA\) dans l'interface AWS](#) dans le Guide de l'utilisateur IAM.

## 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 vous imposent de vous connecter en tant qu'utilisateur racine, consultez [Tâches nécessitant les informations d'identification de l'utilisateur racine](#) dans le Guide de l'utilisateur IAM.

## Utilisateurs et groupes IAM

Un [utilisateur IAM](#) est une identité au sein de votre Compte AWS qui possède des autorisations spécifiques pour une seule personne ou application. Dans la mesure du possible, nous vous recommandons de vous appuyer sur des informations d'identification temporaires plutôt que de créer des utilisateurs IAM ayant des informations d'identification à long terme tels que les clés d'accès. Toutefois, si certains cas d'utilisation spécifiques nécessitent des informations d'identification à long terme avec les utilisateurs IAM, nous vous recommandons de faire pivoter les clés d'accès. Pour plus d'informations, consultez [Rotation régulière des clés d'accès pour les cas d'utilisation nécessitant des informations d'identification](#) dans le Guide de l'utilisateur IAM.

Un [groupe IAM](#) est une identité qui concerne un ensemble d'utilisateurs IAM. 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 avoir un groupe nommé IAMAdmins et accorder à ce groupe les autorisations d'administrer des ressources IAM.

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, consultez [Quand créer un utilisateur IAM \(au lieu d'un rôle\)](#) dans le Guide de l'utilisateur IAM.

## Rôles IAM

Un [rôle IAM](#) est une identité au sein de votre Compte AWS dotée d'autorisations spécifiques. Le concept ressemble à celui d'utilisateur IAM, mais le rôle IAM n'est pas associé à une personne en particulier. Vous pouvez assumer temporairement un rôle IAM dans le en AWS Management Console [changeant de rôle](#). Vous pouvez assumer un rôle en appelant une opération d' AWS API AWS CLI ou en utilisant une URL personnalisée. Pour plus d'informations sur les méthodes d'utilisation des rôles, consultez [Utilisation de rôles IAM](#) dans le Guide de l'utilisateur IAM.

Les rôles IAM avec des informations d'identification temporaires sont utiles dans les cas suivants :

- 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 obtenir des informations sur les rôles pour la fédération, consultez [Création d'un rôle pour un fournisseur d'identité tiers \(fédération\)](#) dans le Guide de l'utilisateur IAM. Si vous utilisez IAM Identity Center, vous configurez un jeu d'autorisations. IAM Identity Center met en corrélation le jeu d'autorisations avec un rôle dans IAM afin de contrôler à quoi vos identités peuvent accéder après leur authentification. 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 d'utilisateur IAM temporaires : un rôle ou un utilisateur IAM peut endosser un rôle IAM pour profiter temporairement d'autorisations différentes pour une tâche spécifique.
- Accès intercompte : vous pouvez utiliser un rôle IAM pour permettre à un utilisateur (principal de confiance) d'un compte différent d'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 en savoir plus sur la différence entre les rôles et les politiques basées sur les ressources pour l'accès intercompte, consultez [Différence entre les rôles IAM et les politiques basées sur les ressources](#) dans le Guide de l'utilisateur IAM.
- 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 direct (FAS) :** lorsque vous utilisez un utilisateur ou un rôle IAM 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. FAS utilise les autorisations du principal appelant et Service AWS, associées Service AWS à la demande, pour adresser des demandes aux services en aval. Les demandes FAS 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 la politique relative à la transmission de demandes FAS, consultez [Sessions de transmission d'accès](#).
- **Rôle de service :** il s'agit d'un [rôle IAM](#) attribué à un service afin de réaliser des actions en votre nom. Un administrateur IAM peut créer, modifier et supprimer une fonction du service à partir d'IAM. Pour plus d'informations, consultez [Création d'un rôle pour la délégation d'autorisations à un Service AWS](#) dans le Guide de l'utilisateur IAM.
- **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 administrateur IAM peut consulter, mais ne peut pas modifier, les autorisations concernant les rôles liés à un service.
- **Applications exécutées sur Amazon EC2 :** vous pouvez utiliser un rôle IAM pour gérer les informations d'identification temporaires pour les applications qui s'exécutent sur une instance EC2 et qui envoient des demandes d'API. AWS CLI AWS Cette solution est préférable au stockage des clés d'accès au sein de l'instance EC2. Pour attribuer un AWS rôle à une instance EC2 et le mettre à la disposition de 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 qui s'exécutent sur l'instance EC2 d'obtenir des informations d'identification temporaires. Pour plus d'informations, consultez [Utilisation d'un rôle IAM pour accorder des autorisations à des applications s'exécutant sur des instances Amazon EC2](#) dans le Guide de l'utilisateur IAM.

Pour savoir dans quel cas utiliser des rôles ou des utilisateurs IAM, consultez [Quand créer un rôle IAM \(au lieu d'un utilisateur\)](#) dans le Guide de l'utilisateur IAM.

## 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 documents JSON. Pour plus d'informations sur la structure et le contenu des documents de politique JSON, consultez [Vue d'ensemble des politiques JSON](#) dans le Guide de l'utilisateur IAM.

Les administrateurs peuvent utiliser les politiques AWS JSON 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 octroyer aux utilisateurs des autorisations d'effectuer des actions sur les ressources dont ils ont besoin, un administrateur IAM peut créer des politiques IAM. L'administrateur peut ensuite ajouter les politiques IAM aux rôles et les utilisateurs peuvent assumer les rôles.

Les politiques IAM définissent les autorisations d'une action, quelle que soit la méthode que vous utilisez pour exécuter 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 à partir de AWS Management Console AWS CLI, de ou de l' AWS API.

## Politiques basées sur l'identité

Les politiques basées sur l'identité sont des documents de politique d'autorisations JSON que vous pouvez attacher à une identité telle qu'un utilisateur, un groupe d'utilisateurs ou un rôle IAM. Ces politiques contrôlent quel type d'actions des utilisateurs et des rôles peuvent exécuter, sur quelles ressources et dans quelles conditions. Pour découvrir comment créer une politique basée sur l'identité, consultez [Création de politiques IAM](#) dans le Guide de l'utilisateur IAM.

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 découvrir comment choisir entre une politique gérée et une politique en ligne, consultez [Choix entre les politiques gérées et les politiques en ligne](#) dans le Guide de l'utilisateur IAM.

## politiques basées sur les ressources

Les politiques basées sur les ressources sont des documents de politique JSON que vous attachez à une ressource. Des politiques basées sur les ressources sont, par exemple, les politiques de

confiance de rôle IAM et des politiques de compartiment. 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 les politiques AWS gérées par IAM dans une stratégie basée sur les ressources.

## Listes de contrôle d'accès (ACL)

Les listes de contrôle d'accès (ACL) vérifie quels principaux (membres de compte, utilisateurs ou rôles) ont l'autorisation d'accéder à une ressource. Les listes de contrôle d'accès sont similaires aux politiques basées sur les ressources, bien qu'elles n'utilisent pas le format de document de politique JSON.

Amazon S3 et Amazon VPC sont des exemples de services qui prennent en charge les ACL. AWS WAF Pour en savoir plus sur les listes de contrôle d'accès, consultez [Vue d'ensemble des listes 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.

- **Limite d'autorisations** : une limite d'autorisations est une fonctionnalité avancée dans laquelle vous définissez le nombre maximal d'autorisations qu'une politique basée sur l'identité peut accorder à une entité IAM (utilisateur ou rôle IAM). 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, consultez [Limites d'autorisations pour des entités IAM](#) dans le Guide de l'utilisateur IAM.

- **Politiques de contrôle des services (SCP)** — Les SCP sont des politiques JSON 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 les multiples comptes AWS de votre entreprise. Si vous activez toutes les fonctionnalités d'une organisation, vous pouvez appliquer les politiques de contrôle des services (SCP) à l'un ou à l'ensemble de vos comptes. Le SCP limite les autorisations pour les entités figurant dans les comptes des membres, y compris chaque utilisateur racine d'un compte AWS d'entre elles. Pour plus d'informations sur les organisations et les SCP, consultez [Fonctionnement des SCP](#) dans le Guide de l'utilisateur AWS Organizations.
- **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 [politiques de séance](#) dans le Guide de l'utilisateur IAM.

## 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 détermine 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 l'utilisateur IAM.

## Comment fonctionne la cryptographie des AWS paiements avec IAM

Avant d'utiliser IAM pour gérer l'accès à la cryptographie des AWS paiements, vous devez comprendre quelles fonctionnalités IAM peuvent être utilisées avec AWS la cryptographie des paiements. Pour obtenir une vue d'ensemble du fonctionnement de la cryptographie des AWS paiements et AWS des autres services avec IAM, consultez la section [AWS Services qui fonctionnent avec IAM dans le guide de l'utilisateur d'IAM](#).

### 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 des actions et 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 politique JSON, consultez [Références des éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

### Actions

Les administrateurs peuvent utiliser les politiques AWS JSON 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 `Action` d'une politique JSON décrit les actions que vous pouvez utiliser pour autoriser ou refuser l'accès à une politique. Les actions de stratégie portent généralement le même nom que l'opération AWS d'API associée. Il existe quelques exceptions, telles que les actions avec autorisations uniquement qui n'ont pas d'opération API correspondante. 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 opération d'`VerifyCardDataAPI` de chiffrement 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'utilisateur IAM.

## Ressources

Les administrateurs peuvent utiliser les politiques AWS JSON 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 de politique JSON Resource indique le ou les objets auxquels l'action s'applique. Les instructions doivent inclure un élément Resource ou NotResource. Il est recommandé de définir une ressource à l'aide de 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 de paiement possède l'ARN suivant :

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

Pour plus d'informations sur le format des ARN, consultez [Amazon Resource Names \(ARN\) et AWS Service Namespaces](#).

Par exemple, pour spécifier l'instance arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaiif1lw2h dans votre instruction, utilisez l'ARN suivant :

```
"Resource": "arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaiif1lw2h"
```

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, les utilisateurs et les rôles IAM ne sont pas autorisés à créer ou à modifier des ressources de chiffrement des AWS paiements. Ils ne peuvent pas non plus effectuer de tâches à l'aide de l'AWS API AWS Management Console AWS CLI, ou. Un administrateur IAM doit créer des politiques IAM autorisant les utilisateurs et les rôles à exécuter des opérations d'API spécifiques sur les ressources spécifiées dont ils ont besoin. Il doit ensuite attacher ces politiques aux utilisateurs ou aux groupes IAM ayant besoin de ces autorisations.

Pour savoir comment créer une stratégie IAM basée sur l'identité à l'aide de ces exemples de documents de stratégie JSON, veuillez consulter [Création de stratégies dans l'onglet JSON](#) dans le Guide de l'utilisateur IAM.

## 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 des API à l'aide de 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 [politiques gérées par AWS](#) ou [politiques gérées par AWS pour les activités professionnelles](#) dans le Guide de l'utilisateur IAM.
- Accorder les autorisations de moindre privilège : lorsque vous définissez des autorisations avec des politiques IAM, accordez uniquement les autorisations nécessaires à l'exécution d'une seule 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 de IAM pour appliquer des autorisations, consultez [politiques et autorisations dans IAM](#) dans le Guide de l'utilisateur IAM.
- Utiliser des conditions dans les politiques IAM pour restreindre davantage l'accès : vous pouvez ajouter une condition à vos politiques afin de limiter l'accès aux actions et aux ressources. Par exemple, vous pouvez écrire une condition de politique pour spécifier que toutes les demandes doivent être envoyées via SSL. 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, consultez [Conditions pour éléments de politique JSON IAM](#) dans le Guide de l'utilisateur IAM.

- Utilisez IAM Access Analyzer pour valider vos politiques IAM afin de garantir des autorisations sécurisées et fonctionnelles : IAM Access Analyzer valide les politiques nouvelles et existantes de manière à ce que les politiques IAM respectent le langage de politique IAM (JSON) et les bonnes pratiques IAM. IAM Access Analyzer fournit plus de 100 vérifications de politiques et des recommandations exploitables pour vous aider à créer des politiques sécurisées et fonctionnelles. Pour plus d'informations, consultez [Validation de politique IAM Access Analyzer](#) dans le Guide de l'utilisateur IAM.
- Exiger l'authentification multifactorielle (MFA) : si vous avez un scénario qui nécessite des utilisateurs IAM ou un utilisateur root, activez l'authentification MFA pour une sécurité accrue. Compte AWS Pour exiger le MFA lorsque des opérations d'API sont appelées, ajoutez des conditions MFA à vos politiques. Pour plus d'informations, consultez [Configuration de l'accès aux API protégé par MFA](#) dans le Guide de l'utilisateur IAM.

Pour plus d'informations sur les bonnes pratiques dans IAM, consultez [Bonnes pratiques de sécurité dans IAM](#) dans le Guide de l'utilisateur IAM.

## 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 (utilisateurs et rôles IAM) tributaires 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 [Ajout d'autorisations à un utilisateur](#) dans le Guide de l'utilisateur IAM.

Il n'est pas nécessaire d'accorder des autorisations de console minimales aux utilisateurs qui appellent uniquement l'API AWS CLI ou l' AWS API. Autorisez plutôt l'accès à uniquement aux actions qui correspondent à l'opération d'API que vous tentez d'effectuer.

## Autorisation accordée aux utilisateurs pour afficher leurs propres autorisations

Cet exemple montre comment créer une politique qui permet aux utilisateurs IAM d'afficher les politiques en ligne et gérées attachées à leur identité d'utilisateur. Cette politique inclut les

autorisations permettant d'effectuer cette action sur la console ou par programmation à l'aide de l'API AWS CLI or AWS .

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "ViewOwnUserInfo",
      "Effect": "Allow",
      "Action": [
        "iam:GetUserPolicy",
        "iam:ListGroupForUser",
        "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 utilisateur IAM 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 les deux ControlPlane opérations. DataPlane

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

## Possibilité d'appeler des API à l'aide de touches spécifiées

Dans cet exemple, vous souhaitez autoriser un utilisateur IAM 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/kwapwa6qaif1lw2h` puis utiliser cette ressource dans deux API, `GenerateCardData` et `VerifyCardData`. À l'inverse, l'utilisateur IAM 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/
kwapwa6qaiif1lw2h"
      ]
    }
  ]
}
```

## 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 utilisateur IAM 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/
kwapwa6qaif1lw2h"
        ]
    }
]
}
```

## Résolution des problèmes d'identité et d'accès liés à la cryptographie des AWS paiements

Des sujets seront ajoutés à cette section au fur et à mesure que les problèmes liés à l'IAM spécifiques à la cryptographie des AWS paiements seront identifiés. Pour obtenir du contenu général sur le dépannage sur des sujets liés à l'IAM, reportez-vous à la [section de résolution](#) des problèmes du guide de l'utilisateur IAM.

# Surveillance de la cryptographie des AWS paiements

La surveillance constitue une part importante de la gestion de la fiabilité, la disponibilité et les performances de la AWS technologie de paiement et de vos autres solutions AWS. AWS met à votre disposition les outils de surveillance suivants pour surveiller le chiffrement des AWS paiements, signaler les incidents et prendre des mesures automatiques le cas échéant :

- Amazon CloudWatch surveille vos AWS ressources et les applications que vous exécutez sur 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 connaître l'utilisation d'UC pour une CloudWatch piste ou d'autres métriques de vos instances Amazon EC2 et démarre automatiquement de nouvelles instances lorsque cela est nécessaire. Pour de plus amples informations, consultez le [Guide de CloudWatch l'utilisateur Amazon](#).
- Amazon CloudWatch Logs vous permet de surveiller, stocker et accéder à vos fichiers journaux à partir d'instances Amazon EC2CloudTrail, et d'autres sources. CloudWatchLes 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 de plus amples informations, consultez le [Guide de l'utilisateur Amazon CloudWatch Logs](#).
- Amazon EventBridge peut être utilisé pour automatiser vos AWS services et répondre automatiquement aux événements système tels que les problèmes de disponibilité des applications ou les changements de ressources. Les événements des services AWS sont fournis à EventBridge presque en temps réel. Vous pouvez écrire des règles simples pour préciser les événements qui vous intéressent et les actions automatisées à effectuer quand un événement correspond à une règle. Pour de plus amples informations, consultez le [Guide de EventBridge l'utilisateur Amazon](#).
- AWS CloudTrail capture les appels d'API et les événements associés créés par ou au nom de votre compte AWS et envoie les fichiers journaux à un compartiment Amazon S3 que vous spécifiez. Vous pouvez identifier les utilisateurs et les comptes qui ont appelé AWS, l'adresse IP source à partir de laquelle les appels ont été émis, ainsi que le moment où les appels ont eu lieu. Pour de plus amples informations, veuillez consulter le [Guide de l'utilisateur AWS CloudTrail](#).

**Note**

AWS CloudTrailles journaux sont pris en charge pour les opérations du plan de contrôle, telles queCreateKey, mais pas pour les opérations du plan de données, telles que la génération de données de carte

## Enregistrement des appels d'API AWS de cryptographie des paiements à l'aide de AWS CloudTrail

AWSLa cryptographie des paiements est intégrée àAWS CloudTrail, un service qui enregistre les actions effectuées par un utilisateur, un rôle ou un AWS service dans le domaine de la cryptographie des AWS paiements. CloudTrailcapture les appels d'API pour la cryptographie des AWS paiements. Les appels de la console AWS Payment Cryptography et les appels de code aux opérations d'API de AWS paiement. Si vous créez un journal de suivi, vous pouvez activer la livraison continue des CloudTrail événements dans un compartiment Amazon S3, y compris les événements pour la cryptographie des AWS paiements. Si vous ne configurez pas de journal d'activité, vous pouvez toujours afficher les événements les plus récents dans la console CloudTrail dans Historique des événements. Avec les informations collectées parCloudTrail, vous pouvez déterminer la demande qui a été envoyée à AWS Payment Cryptography, l'adresse IP à partir de laquelle la demande a été effectuée, l'auteur et la date de la demande, ainsi que d'autres détails.

Pour en savoir plus sur CloudTrail, veuillez consulter le [Guide de l'utilisateur AWS CloudTrail](#).

**Note**

L'intégration de Cloudtrail est actuellement prise en charge uniquement pour les opérations du plan de contrôle.

## AWSInformations sur la cryptographie des paiements dans CloudTrail

CloudTrail est activé sur votre compte AWS lorsque vous créez le compte. Quand une activité a lieu dans AWS Payment Cryptography, cette activité est enregistrée dans un CloudTrail événement. AWS Vous pouvez afficher, rechercher et télécharger les événements récents dans votre compte AWS.

Pour plus d'informations, consultez [Affichage des événements avec l'historique des événements CloudTrail](#).

Pour un enregistrement continu des événements dans votre AWS compte, y compris les événements concernant la cryptographie des AWS paiements, créez un journal de suivi. Un journal CloudTrail de suivi permet de livrer des fichiers journaux dans un compartiment Amazon S3. Par défaut, lorsque vous créez un journal de suivi dans la console, il s'applique à toutes les régions AWS. Le journal de suivi consigne les événements de toutes les Régions dans la partition AWS et livre les fichiers journaux dans le compartiment Amazon S3 de votre choix. En outre, vous pouvez configurer d'autres services AWS pour analyser plus en profondeur les données d'événement collectées dans les journaux CloudTrail et agir sur celles-ci. Pour en savoir plus, consultez les ressources suivantes :

- [Présentation de la création d'un journal d'activité](#)
- [Intégrations et services pris en charge par CloudTrail](#)
- [Configuration des notifications Amazon SNS pour CloudTrail](#)
- [Réception des fichiers journaux CloudTrail de plusieurs régions](#)
- [Réception des fichiers journaux CloudTrail de plusieurs comptes](#)

CloudTrail enregistre les opérations de cryptographie des AWS paiements, telles que [CreateKey](#), [ImportKey](#), [DeleteKey](#), [ListKeysTagResource](#), et toutes les autres opérations du plan de contrôle.

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 effectuée avec les informations d'identification utilisateur racine 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 requête a été effectuée par un autre service AWS.

Pour de plus amples informations, veuillez consulter l'[élément userIdentity CloudTrail](#).

## Comprendre les AWS entrées du fichier journal de cryptographie des paiements

Un journal d'activité est une configuration qui permet d'envoyer des événements sous forme de fichiers journaux à un compartiment Simple Storage Service (Amazon S3) que vous spécifiez. Les fichiers journaux CloudTrail contiennent une ou plusieurs entrées de journal. Un événement représente une demande unique provenant de n'importe quelle source et comprend des informations sur l'action demandée, la date et l'heure de l'action, les paramètres de la requête, etc. Les fichiers journaux CloudTrail ne constituent pas une série ordonnée retraçant les appels d'API publics. Ils ne suivent aucun ordre précis.

L'exemple présente une entrée de CloudTrail journal qui illustre l'CreateKey action de cryptographie des AWS paiements.

```
{
  CloudTrailEvent: {
    tlsDetails= {
      TlsDetails: {
        cipherSuite=TLS_AES_128_GCM_SHA256,
        tlsVersion=TLSv1.3,
        clientProvidedHostHeader=pdx80.controlplane.paymentcryptography.us-
west-2.amazonaws.com
      }
    },
    requestParameters=CreateKeyInput (
      keyAttributes=KeyAttributes(
        KeyUsage=TR31_B0_BASE_DERIVATION_KEY,
        keyClass=SYMMETRIC_KEY,
        keyAlgorithm=AES_128,
        keyModesOfUse=KeyModesOfUse(
          encrypt=false,
          decrypt=false,
          wrap=false
          unwrap=false,
          generate=false,
          sign=false,
          verify=false,
          deriveKey=true,
          noRestrictions=false)
        ),
  ),
```

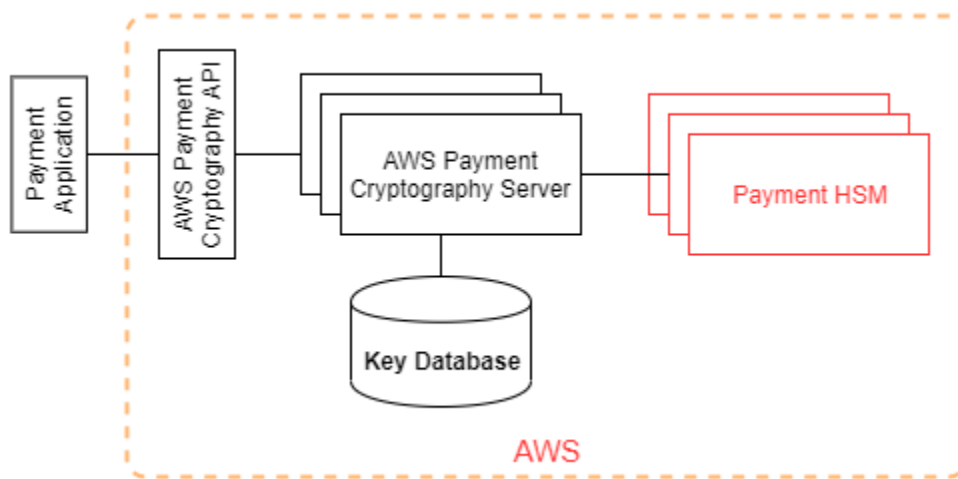
```
keyCheckValueAlgorithm=null,
exportable=true,
enabled=true,
tags=null),
eventName=CreateKey,
userAgent=Coral/Apache-HttpClient5,
responseElements=CreateKeyOutput(
  key=Key(
    keyArn=arn:aws:payment-cryptography:us-
east-2:111122223333:key/5rplquuwozodpwp,
    keyAttributes=KeyAttributes(
      KeyUsage=TR31_B0_BASE_DERIVATION_KEY,
      keyClass=SYMMETRIC_KEY,
      keyAlgorithm=AES_128,
      keyModesOfUse=KeyModesOfUse(
        encrypt=false,
        decrypt=false,
        wrap=false,
        unwrap=false,
        generate=false,
        sign=false,
        verify=false,
        deriveKey=true,
        noRestrictions=false)
      ),
    keyCheckValue=FE23D3,
    keyCheckValueAlgorithm=ANSI_X9_24,
    enabled=true,
    exportable=true,
    keyState=CREATE_COMPLETE,
    keyOrigin=AWS_PAYMENT_CRYPTOGRAPHY,
    createTimestamp=Sun May 21 18:58:32 UTC 2023,
    usageStartTimestamp=Sun May 21 18:58:32 UTC 2023,
    usageStopTimestamp=null,
    deletePendingTimestamp=null,
    deleteTimestamp=null)
  ),
sourceIPAddress=192.158.1.38,
userIdentity={
  UserIdentity: {
    arn=arn:aws:sts::111122223333:assumed-role/TestAssumeRole-us-west-2-PDX80/
ControlPlane-IntegTest-68211a2a-3e9d-42b7-86ac-c682520e0410,
    invokedBy=null,
    accessKeyId=,
```

```
    type=AssumedRole,
    sessionContext={
      SessionContext: {
        sessionIssuer={
          SessionIssuer: {arn=arn:aws:iam::111122223333:role/TestAssumeRole-us-
west-2-PDX80,
          type=Role,
          accountId=111122223333,
          userName=TestAssumeRole-us-west-2-PDX80,
          principalId=}
        },
        attributes={
          SessionContextAttributes: {
            creationDate=Sun May 21 18:58:31 UTC 2023,
            mfaAuthenticated=false
          }
        },
        webIdFederationData=null
      }
    },
    username=null,
    principalId=:ControlPlane-User,
    accountId=111122223333,
    identityProvider=null
  }
},
eventTime=Sun May 21 18:58:32 UTC 2023,
managementEvent=true,
recipientAccountId=111122223333,
awsRegion=us-west-2,
requestID=151cdd67-4321-1234-9999-dce10d45c92e,
eventVersion=1.08, eventType=AwsApiCall,
readOnly=false,
eventID=c69e3101-eac2-1b4d-b942-019919ad2faf,
eventSource=payment-cryptography.amazonaws.com,
eventCategory=Management,
additionalEventData={
}
}
}
```

## Détails cryptographiques

AWS La cryptographie des paiements fournit une interface Web permettant de générer et de gérer des clés cryptographiques pour les transactions de paiement. AWS La cryptographie des paiements 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 pour vous aider à évaluer les fonctions offertes par le service. AWS

[AWS La cryptographie des paiements contient plusieurs interfaces \(y compris une API RESTful, via l'interface de ligne de commande AWS, le kit SDK AWS, etc. AWS Management Console\) permettant de demander les opérations cryptographiques d'un parc distribué de modules de sécurité matériels validés par la norme PCI PTS HSM.](#)



AWS La cryptographie des paiements est un service à plusieurs niveaux, composé d'hôtes orientés vers le Web AWS et d'un niveau de HSM. Le regroupement de ces hôtes à plusieurs niveaux forme la pile cryptographique pour les AWS paiements. Toutes les demandes adressées à AWS Payment cryptography doivent être effectuées via le protocole TLS (Transport Layer Security) et se terminer vers un hôte pour le cryptographie pour les AWS paiements. Les hôtes du service n'autorisent le protocole TLS qu'avec une suite de chiffrement garantissant une [parfaite confidentialité ultérieure](#). Le service authentifie et autorise vos demandes à l'aide des 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 de service et le [HSM](#) sont sécurisées par le protocole TLS mutuel (MTLS) pour l'authentification et le chiffrement.



## Objectifs de conception

AWS La 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](#)
- [Entropie et génération de nombres aléatoires](#)
- [Opérations clés symétriques](#)
- [Opérations clés asymétriques](#)
- [Rangement des clés](#)
- [Importation de clés à l'aide de clés symétriques](#)
- [Importation de clés à l'aide de clés asymétriques](#)
- [Exportation de clés](#)
- [Protocole DUKPT \(clé unique dérivée par transaction\)](#)
- [Hiérarchie des clés](#)

## 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.

## 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/encapsulation 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.

## 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

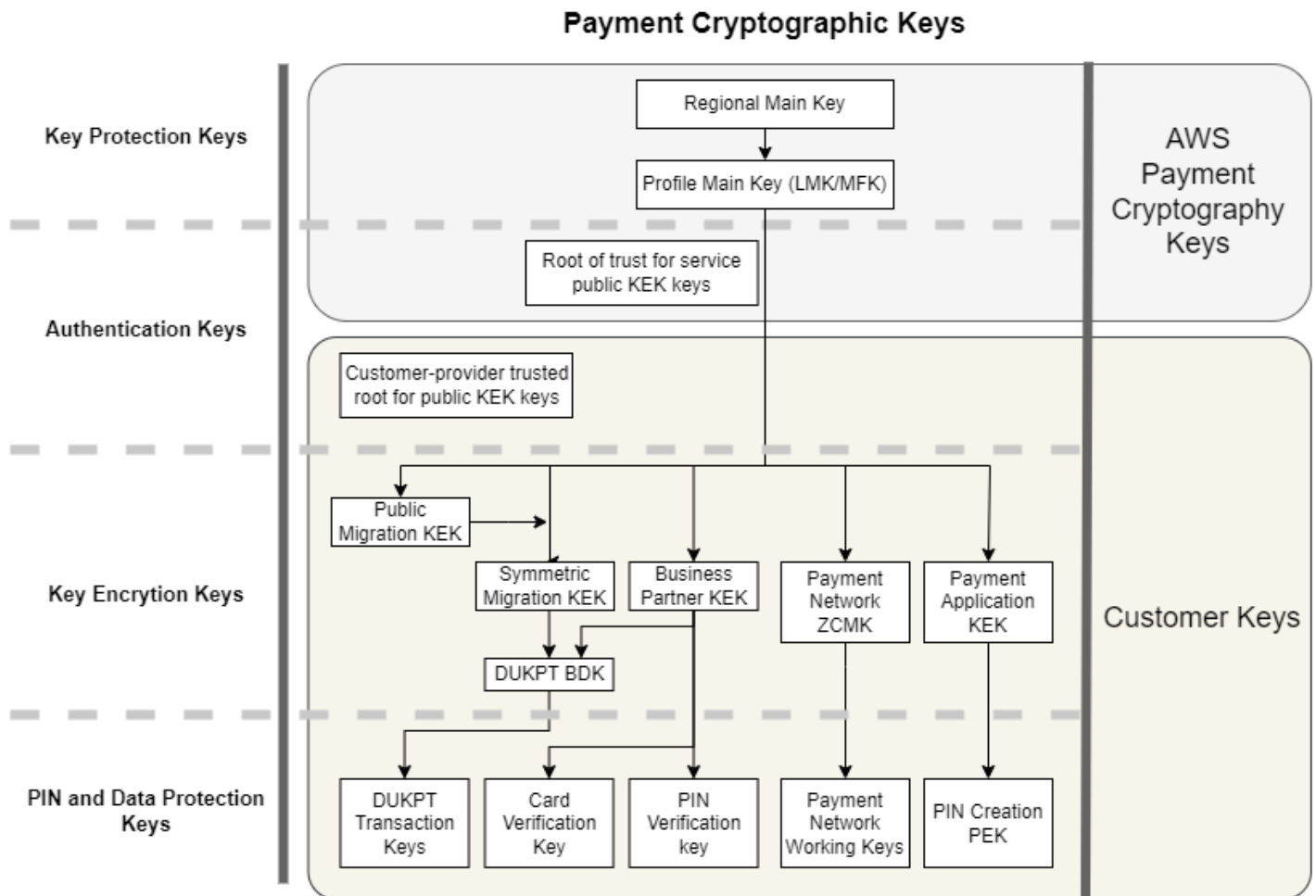
Les clés peuvent être exportées et protégées par des clés dotées de la valeur appropriée KeyUsage et qui sont aussi fortes ou plus fortes que la clé à exporter.

## 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	Description
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 partenaires commerciaux, des réseaux de paiement ou différentes applications au sein de votre organisation.

Clé	Description
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 vérification, 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 vérification, 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'émetteur 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.

## 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 niveau 3 et utilisent également les versions du microprogramme et la politique de sécurité répertoriée sur la [liste des périphériques PCI PTS approuvée par le PCI Security Standards Council en tant que conformité 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 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

- [Génération](#)
- [Synchronisation des touches principales de la région](#)
- [Rotation des clés principales de la région](#)
- [Synchronisation des touches principales du profil](#)
- [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 équilibrateurs 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.

## Partage de clés

Les clés peuvent être partagées avec d'autres comptes au sein ou en dehors de votre organisation à l'aide d'AWS Resource Access Manager (<https://docs.aws.amazon.com/ARG/index.html>). Les clés peuvent être regroupées dans un partage de ressources puis partagées avec un compte ou avec des utilisateurs et des rôles IAM spécifiques au sein d'un compte. Vous spécifiez les autorisations d'utilisation pour chaque partage de ressources. Les autorisations de partage sont limitées par une politique de ressources clés. Une clé partagée n'autorisera pas une action limitée par sa propre politique. L'autorisation de partage peut être retirée à tout moment.

## 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	Ajusté	Description
Alias	Chaque région prise en charge : 2 000	<a href="#">Oui</a>	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	<a href="#">Oui</a>	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	<a href="#">Oui</a>	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	Ajuste	Description
Taux combiné de demandes de plan de données (symétrique)	Chaque Région prise en charge : 500 par seconde	<a href="#">Oui</a>	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	<a href="#">Oui</a>	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
<a href="#">Publication de fonctionnalités</a>	Ajout d'informations sur les points de terminaison VPC (PrivateLink) et exemples d'iCVV.	30 mai 2024
<a href="#">Publication de fonctionnalités</a>	Informations ajoutées sur les nouvelles fonctionnalités relatives à l'importation/exportation de clés à l'aide de RSA et à l'exportation de clés DUKPT IPEK/IK.	15 janvier 2024
<a href="#">Première version</a>	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.