



Guida per l'utente

AWS Crittografia dei pagamenti



AWS Crittografia dei pagamenti: Guida per l'utente

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

I marchi e l'immagine commerciale di Amazon non possono essere utilizzati in relazione a prodotti o servizi che non siano di Amazon, in una qualsiasi modalità che possa causare confusione tra i clienti o in una qualsiasi modalità che denigri o discrediti Amazon. Tutti gli altri marchi non di proprietà di Amazon sono di proprietà delle rispettive aziende, che possono o meno essere associate, collegate o sponsorizzate da Amazon.

Table of Contents

Cosa è AWS Crittografia dei pagamenti?	1
Concetti	2
Terminologia di settore	4
Tipi di chiavi comuni	4
Altri termini	7
Servizi correlati	10
Ulteriori informazioni	11
Endpoints	11
Endpoint del piano di controllo	11
Endpoint del piano dati	12
Nozioni di base	13
Prerequisiti	13
Fase 1: Creare una chiave	13
Passaggio 2: generare un valore CVV2 utilizzando la chiave	15
Fase 3: Verificare il valore generato nel passaggio 2	15
Fase 4: Eseguire un test negativo	16
Fase 5: (Facoltativo) Pulizia	16
Gestione delle chiavi	18
Generazione delle chiavi	18
Generazione di una chiave TDES 2KEY	19
Generazione di una chiave di crittografia PIN	20
Crea una chiave asimmetrica (RSA)	21
Generazione di una chiave PVV (PVV)	22
Elenca le chiavi	23
Abilitazione e disabilitazione delle chiavi	24
Inizia l'utilizzo della chiave	25
Interrompi l'uso delle chiavi	26
Eliminazione delle chiavi	27
Informazioni sul periodo di attesa	28
Importa ed esporta chiavi	31
Chiavi di importazione	32
Chiavi di esportazione	42
Utilizzo di alias	50
Informazioni sugli alias	51

Utilizzo di alias nelle applicazioni	54
API correlate	55
Procurati le chiavi	55
Ottieni la chiave/certificato pubblico associato a una coppia di chiavi	56
Chiavi di tagging	57
Informazioni sui tag nella crittografia dei pagamenti AWS	57
Visualizzazione dei tag chiave nella console	59
Gestione dei tag chiave con operazioni API	59
Controllo degli accessi ai tag	62
Utilizzo dei tag per controllare l'accesso alle chiavi	66
Comprensione degli attributi chiave	69
Chiavi simmetriche	69
Chiavi asimmetriche	72
Operazioni sui dati	73
Crittografa, decrittografa e ricrittografa i dati	73
Crittografare i dati	74
Decrittare i dati	78
Genera e verifica i dati della carta	81
Genera i dati delle carte	82
Verifica i dati della carta	83
Generazione, traduzione e verifica dei dati PIN	85
Traduci i dati del PIN	85
Genera dati PIN	87
Verifica i dati del PIN	90
Crittogramma Verify Auth Request (ARQC)	92
Creazione di dati sulle transazioni	93
Riempimento dei dati delle transazioni	94
Esempi	94
Genera e verifica MAC	96
Genera MAC	97
Verifica MAC	98
Tipi di chiave per operazioni specifiche sui dati	98
GenerateCardDati	99
VerifyCardDati	101
GeneratePinData (per schemi VISA/ABA)	102
GeneratePinData (perIBM3624)	102

VerifyPinData (per schemi VISA/ABA)	104
VerifyPinData (perIBM3624)	104
Decrittografia dei dati	105
Encrypt Data (Crittografa dati)	107
Traduci PIN Data	108
Genera/verifica MAC	109
VerifyAuthRequestCryptogram	110
Chiave Import/Export	111
Tipi di chiavi non utilizzati	111
Sicurezza	112
Protezione dei dati	112
Protezione del materiale della chiave	114
Crittografia dei dati	114
Crittografia a riposo	114
Crittografia in transito	115
Riservatezza del traffico Internet	115
Resilienza	116
Isolamento regionale	116
Design multi-tenant	117
Sicurezza dell'infrastruttura	117
Isolamento degli host fisici	118
Usa Amazon VPC e AWS PrivateLink	118
Considerazioni sugli endpoint AWS VPC per la crittografia dei pagamenti	119
Creazione di un endpoint VPC per la crittografia dei pagamenti AWS	119
Connessione a un endpoint VPC	120
Controllo dell'accesso all'endpoint VPC	121
Utilizzo di un endpoint VPC in un'istruzione di policy	125
Registrazione dell'endpoint VPC	128
Best practice di sicurezza	130
Convalida della conformità	132
Gestione dell'identità e degli accessi	133
Destinatari	133
Autenticazione con identità	134
Account AWS utente root	135
Utenti e gruppi IAM	135
Ruoli IAM	135

Gestione dell'accesso con policy	137
Policy basate su identità	138
Policy basate su risorse	138
Liste di controllo degli accessi (ACL)	138
Altri tipi di policy	139
Più tipi di policy	139
Come funziona la crittografia dei AWS pagamenti con IAM	140
AWS Politiche basate sull'identità della crittografia dei pagamenti	140
Autorizzazione basata sui tag di crittografia dei pagamenti AWS	142
Esempi di policy basate su identità	142
Best practice delle policy	143
Utilizzo della console	144
Consentire agli utenti di visualizzare le loro autorizzazioni	144
Capacità di accedere a tutti gli aspetti della crittografia dei pagamenti AWS	145
Possibilità di chiamare le API utilizzando chiavi specificate	146
Capacità di negare specificamente una risorsa	147
Risoluzione dei problemi	148
Monitoraggio	149
Log CloudTrail	150
AWS Informazioni sulla crittografia dei pagamenti in CloudTrail	150
Informazioni sulle voci dei AWS file di registro di Payment Cryptography	151
Dettagli crittografici	155
Obiettivi di progettazione di	156
Fondazioni	157
Primitive di crittografia	157
Entropia e generazione di numeri casuali	157
Operazioni chiave simmetriche	158
Operazioni con chiavi asimmetriche	158
Archiviazione delle chiavi	158
Importazione di chiavi tramite chiavi simmetriche	159
Importazione di chiavi tramite chiavi asimmetriche	159
Esportazione di chiavi	159
Protocollo DUKPT (Derived Unique Key Per Transaction)	159
Gerarchia delle chiavi	159
Operazioni interne	163
Specifiche e ciclo di vita HSM	164

Sicurezza fisica dei dispositivi HSM	164
Inizializzazione HSM	165
Assistenza e riparazione HSM	165
Disattivazione HSM	165
Aggiornamento del firmware HSM	165
Accesso dell'operatore	166
Gestione delle chiavi	166
Operazioni con i clienti	173
Generazione delle chiavi	174
Importazione delle chiavi	174
Esportazione delle chiavi	175
Eliminazione delle chiavi	175
Rotazione delle chiavi	176
Quote	177
Cronologia dei documenti	179
.....	clxxx

Cosa è AWS Crittografia dei pagamenti?

La crittografia dei pagamenti è gestita da un servizio AWS che fornisce l'accesso alle funzioni crittografiche e alla gestione delle chiavi utilizzate nell'elaborazione dei pagamenti in conformità con gli standard del settore delle carte di pagamento (PCI) senza la necessità di procurarsi istanze HSM di pagamento dedicate. La crittografia dei pagamenti offre ai clienti che svolgono funzioni di pagamento come acquirenti, facilitatori di pagamento, reti, switch, processori e banche la possibilità di avvicinare le loro operazioni crittografiche di pagamento alle applicazioni nel cloud e ridurre al minimo la dipendenza dai data center ausiliari o dalle strutture di colocation contenenti HSM di pagamento dedicati.

Il servizio è progettato per soddisfare le norme di settore applicabili, tra cui PCI PIN, PCI P2PE e PCI DSS, e sfrutta l'hardware che è [Certificazione PCI PTS HSM V3 e FIPS 140-2 di livello 3](#). È progettato per supportare una bassa latenza e [elevati livelli di operatività e resilienza](#). La crittografia dei pagamenti è completamente elastica ed elimina molti dei requisiti operativi degli HSM locali, come la necessità di fornire hardware, gestire in modo sicuro il materiale chiave e mantenere i backup di emergenza in strutture sicure. AWS Payment Cryptography ti offre anche la possibilità di condividere le chiavi con i tuoi partner elettronicamente, eliminando la necessità di condividere componenti cartacei in testo non crittografato.

Puoi usare il [AWS API Payment Cryptography Control Plane](#) per creare e gestire le chiavi.

Puoi usare il [AWS API Data Plane per la crittografia dei pagamenti](#) utilizzare chiavi di crittografia per l'elaborazione delle transazioni relative ai pagamenti e le operazioni crittografiche associate.

La crittografia dei pagamenti offre importanti funzionalità che puoi utilizzare per gestire le tue chiavi:

- Crea e gestisci sistemi simmetrici e asimmetrici. Chiavi di crittografia dei pagamenti, comprese le chiavi TDES, AES e RSA, con indicazione dello scopo previsto, ad esempio per la generazione di CVV o la derivazione di chiavi DUKPT.
- Memorizza automaticamente il tuo. Chiavi di crittografia dei pagamenti in modo sicuro, protette da moduli di sicurezza hardware (HSM), che garantisce al contempo la separazione delle chiavi tra i casi d'uso.
- Crea, elimina, elenca e aggiorna gli alias, che sono «nomi amichevoli» che possono essere usati per accedere o controllare l'accesso ai tuoi. Chiavi crittografiche di pagamento.

- Tagga il tuoAWSChiavi crittografiche di pagamento per l'identificazione, il raggruppamento, l'automazione, il controllo degli accessi e il monitoraggio dei costi.
- Importa ed esporta chiavi simmetriche traAWSCrittografia dei pagamenti e HSM (o di terze parti) utilizzando Key Encryption Keys (KEK) secondo TR-31 (Interoperable Secure Key Exchange Key Block Specification).
- Importa ed esporta chiavi di crittografia a chiave simmetrica (KEK) traAWSCrittografia dei pagamenti e altri sistemi che utilizzano coppie di chiavi asimmetriche, seguiti da mezzi elettronici come TR-34 (Metodo per la distribuzione di chiavi simmetriche mediante tecniche asimmetriche).

Puoi usare il tuoAWSChiavi di crittografia dei pagamenti nelle operazioni crittografiche, come:

- Crittografa, decrittografa e ricrittografa i dati in modo simmetrico o asimmetricoAWSChiavi crittografiche di pagamento.
- Traduci in modo sicuro i dati sensibili (come i pin del titolare della carta) tra le chiavi di crittografia senza esporre il testo non crittografato in conformità con le regole PCI PIN.
- Genera o convalida i dati del titolare della carta come CVV, CVV2 o ARQC.
- Genera e convalida i pin del titolare della carta.
- Genera o convalida le firme MAC.

Concetti

Scopri i termini e i concetti di base utilizzati nella crittografia dei AWS pagamenti e come utilizzarli per proteggere i tuoi dati.

Alias

Un nome intuitivo associato a una chiave di crittografia dei AWS pagamenti. L'alias può essere utilizzato in modo intercambiabile con la chiave [ARN](#) in molte operazioni dell'API Payment Cryptography. AWS Gli alias consentono di ruotare o modificare in altro modo le chiavi senza influire sul codice dell'applicazione. Il nome alias è una stringa di massimo 256 caratteri. Identifica in modo univoco una chiave di crittografia dei AWS pagamenti associata all'interno di un account e di una regione. In AWS Payment Cryptography, gli alias iniziano sempre con. `alias/`

Il formato di un nome alias è il seguente:

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

Per esempio:

```
alias/sampleAlias2
```

ARN della chiave

L'ARN chiave è l'Amazon Resource Name (ARN) di una voce chiave in Payment Cryptography. AWS È un identificatore unico e completamente qualificato per la chiave Payment Cryptography. AWS Un ARN chiave include una regione Account AWS, e un ID generato casualmente. L'ARN non è correlato o derivato dal materiale chiave. Poiché vengono assegnati automaticamente durante le operazioni di creazione o importazione, questi valori non sono idempotenti. L'importazione della stessa chiave più volte comporterà la creazione di più ARN chiave con il proprio ciclo di vita.

Il formato di un ARN della chiave è il seguente:

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

Di seguito è riportato un esempio di codice ARN:

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

Identificatore chiave

Un identificatore di chiave è un riferimento a una chiave e uno (o più) di essi sono input tipici delle operazioni di crittografia dei AWS pagamenti. [Gli identificatori di chiave validi possono essere un Key Arn un Key Alias.](#)

AWS Chiavi di crittografia dei pagamenti

AWS Le chiavi (chiavi) di crittografia dei pagamenti vengono utilizzate per tutte le funzioni crittografiche. Le chiavi vengono generate direttamente dall'utente utilizzando il comando `create key` o aggiunte al sistema chiamando `key import`. L'origine di una chiave può essere determinata esaminando l'attributo `KeyOrigin`. AWS La crittografia dei pagamenti supporta anche chiavi derivate o intermedie utilizzate durante le operazioni crittografiche come quelle utilizzate da DUKPT.

Queste chiavi hanno attributi immutabili e mutabili definiti al momento della creazione. Gli attributi, come algoritmo, lunghezza e utilizzo, vengono definiti al momento della creazione e non possono

essere modificati. Altri, come la data di validità o la data di scadenza, possono essere modificati. Consulta il [riferimento all'API AWS Payment Cryptography](#) per un elenco completo degli attributi delle chiavi di crittografia dei AWS pagamenti.

AWS Le chiavi di crittografia dei pagamenti hanno tipi di chiave, definiti principalmente da [ANSI X9 TR 31](#), che ne limitano l'uso allo scopo previsto, come specificato nel requisito 19 del PCI PIN v3.1.

Gli attributi sono associati alle chiavi mediante blocchi chiave quando vengono archiviati, condivisi con altri account o esportati come specificato nel Requisito 18-3 del PCI PIN v3.1.

Le chiavi vengono identificate nella piattaforma AWS Payment Cryptography utilizzando un valore univoco noto come chiave Amazon Resource Name (ARN).

Note

ARNLa chiave viene generata quando una chiave viene inizialmente creata o importata nel servizio AWS Payment Cryptography. Pertanto, se si aggiunge lo stesso materiale chiave più volte utilizzando la funzionalità di importazione delle chiavi, lo stesso materiale chiave verrà posizionato in più chiavi ma ognuna con un ciclo di vita della chiave diverso.

Terminologia di settore

Argomenti

- [Tipi di chiavi comuni](#)
- [Altri termini](#)

Tipi di chiavi comuni

AWK

Una chiave di lavoro dell'acquirer (AWK) è una chiave tipicamente utilizzata per lo scambio di dati tra un processore acquirer/acquirer e una rete (come Visa o Mastercard). Storicamente AWK utilizza 3DES per la crittografia e sarebbe rappresentato come TR31_P0_PIN_ENCRYPTION_KEY.

BDK

Una chiave di derivazione di base (BDK) è una chiave di lavoro utilizzata per derivare chiavi successive ed è comunemente usata come parte del processo PCI PIN e PCI P2PE DUKPT. È indicata come TR31_B0_BASE_DERIVATION_KEY.

CMK

Una card master key (CMK) è una o più chiavi specifiche per una scheda tipicamente derivate da una chiave master dell'emittente, PAN e PSN e sono in genere chiavi 3DES. Queste chiavi vengono memorizzate sul chip EMV durante la personalizzazione. Esempi di CMK includono le chiavi AC, SMI e SMC.

CMK-AC

Una chiave di crittografia dell'applicazione (AC) viene utilizzata come parte delle transazioni EMV per generare il crittogramma della transazione ed è un tipo di chiave master della carta.

CMK-SMI

Una chiave SMI (Secure Messaging Integrity) viene utilizzata come parte di EMV per verificare l'integrità dei payload inviati alla scheda tramite MAC, ad esempio gli script di aggiornamento dei pin. È un tipo di chiave master della carta.

CMK-SMC

Una chiave SMC (Secure Messaging Reservatezza) viene utilizzata come parte di EMV per crittografare i dati inviati alla scheda, come gli aggiornamenti dei pin. È un tipo di chiave master della carta.

CVK

Una chiave di verifica della carta (CVK) è una chiave utilizzata per generare CVV, CVV2 e valori simili utilizzando un algoritmo definito e per convalidare un input. È indicata come TR31_C0_CARD_VERIFICATION_KEY.

iCVV

iCVV è un valore simile a CVV2 ma incorporato con i dati equivalenti a track2 su una scheda EMV (Chip). Questo valore viene calcolato utilizzando un codice di servizio 999 ed è diverso dal CVV1/ CVV2 per evitare che le informazioni rubate vengano utilizzate per creare nuove credenziali di pagamento di tipo diverso. Ad esempio, se sono stati ottenuti dati sulle transazioni con chip, non è possibile utilizzare questi dati per generare una banda magnetica (CVV1) o per acquisti online (CVV2).

Utilizza una chiave [???](#)

INCHIOSTRO

Una chiave master dell'emittente (IMK) è una chiave master utilizzata come parte della personalizzazione delle chip card EMV. In genere ci saranno 3 IMK, uno ciascuno per le chiavi AC (crittogramma), SMI (chiave master dello script per integrità/firma) e SMC (chiave master dello script per la riservatezza/crittografia).

cinematica inversa

[Una chiave iniziale \(IK\) è la prima chiave utilizzata nel processo DUKPT e deriva dalla Base Derivation Key \(BDK\).](#) Nessuna transazione viene elaborata su questa chiave, ma viene utilizzata per derivare chiavi future che verranno utilizzate per le transazioni. Il metodo di derivazione per creare un IK è stato definito in X9. 24-1:2017. Quando viene utilizzato un TDES BDK, X9. 24-1:2009 è lo standard applicabile e IK viene sostituito dalla Initial Pin Encryption Key (IPEK).

IPEK

[Una chiave di crittografia PIN iniziale \(IPEK\) è la chiave iniziale utilizzata nel processo DUKPT e deriva dalla Base Derivation Key \(BDK\).](#) Nessuna transazione viene elaborata su questa chiave, ma viene utilizzata per derivare chiavi future che verranno utilizzate per le transazioni. IPEK è un termine improprio in quanto questa chiave può essere utilizzata anche per derivare la crittografia dei dati e le chiavi mac. Il metodo di derivazione per creare un IPEK è stato definito in X9. 24-1:2009. [Quando viene utilizzato un BDK AES, X9. 24-1:2017 è lo standard applicabile e IPEK viene sostituito da Initial Key \(IK\).](#)

WIK

Una chiave di lavoro dell'emittente (IWK) è una chiave generalmente utilizzata per lo scambio di dati tra un processore emittente/emittente e una rete (come Visa o Mastercard). Storicamente IWK utilizza 3DES per la crittografia ed è rappresentato come TR31_P0_PIN_ENCRYPTION_KEY.

KEK

Una chiave di crittografia (KEK) è una chiave utilizzata per crittografare altre chiavi per la trasmissione o l'archiviazione. Le chiavi destinate a proteggere altre chiavi in genere hanno un valore di KeyUsage TR31_K0_KEY_ENCRYPTION_KEY secondo lo standard. [TR-31](#)

PEK

Una chiave di crittografia PIN (PEK) è un tipo di chiave di lavoro utilizzata per crittografare i PIN per l'archiviazione o la trasmissione tra due parti. IWK e AWK sono due esempi di

usi specifici delle chiavi di crittografia dei pin. Queste chiavi sono rappresentate come TR31_P0_PIN_ENCRYPTION_KEY.

PVK

Una chiave di verifica PIN (PVK) è un tipo di chiave funzionante utilizzata per generare valori di verifica del PIN come PVV. I due tipi più comuni sono TR31_V1_IBM3624_PIN_VERIFICATION_KEY utilizzato per generare valori di offset IBM3624 e TR31_V2_VISA_PIN_VERIFICATION_KEY utilizzato per i valori di verifica VISA/ABA.

Altri termini

ARQC

Authorization Request Cryptogram (ARQC) è un crittogramma generato al momento della transazione da una chip card standard EMV (o un'implementazione contactless equivalente). In genere, un ARQC viene generato da una chip card e inoltrato a un emittente o al suo agente per la verifica al momento della transazione.

DUPPT

Derived Unique Key Per Transaction (DUKPT) è uno standard di gestione delle chiavi tipicamente utilizzato per definire l'uso di chiavi di crittografia monouso su POS/POI fisici. Storicamente, DUKPT utilizza 3DES per la crittografia. Lo standard di settore per DUKPT è definito in ANSI X9.24-3-2017.

EMV

[EMV](#) (originariamente Europay, Mastercard, Visa) è un organismo tecnico che collabora con le parti interessate ai pagamenti per creare standard e tecnologie di pagamento interoperabili. Un esempio di standard riguarda le carte chip/contactless e i terminali di pagamento con cui interagiscono, inclusa la crittografia utilizzata. La derivazione delle chiavi EMV si riferisce ai metodi di generazione di chiavi univoche per ogni carta di pagamento sulla base di un set iniziale di chiavi come [IMK](#)

HSM (HSM)

Un Hardware Security Module (HSM) è un dispositivo fisico che protegge le operazioni crittografiche (ad esempio, crittografia, decrittografia e firme digitali) nonché le chiavi sottostanti utilizzate per tali operazioni.

KCV

Key Check Value (KCV) si riferisce a una varietà di metodi di checksum utilizzati principalmente per confrontare le chiavi tra loro senza avere accesso al materiale chiave effettivo. I KCV sono stati utilizzati anche per la convalida dell'integrità (specialmente durante lo scambio di chiavi), sebbene questo ruolo sia ora incluso come parte di formati di blocchi chiave come [TR-31](#). Per le chiavi TDES, il KCV viene calcolato crittografando 8 byte, ciascuno con valore zero, con la chiave da controllare e conservando i 3 byte di ordine più alto del risultato crittografato. Per le chiavi AES, il KCV viene calcolato utilizzando un algoritmo CMAC in cui i dati di input sono pari a 16 byte pari a zero e conservano i 3 byte di ordine più alto del risultato crittografato.

KDH

[Un Key Distribution Host \(KDH\) è un dispositivo o un sistema che invia chiavi in un processo di scambio di chiavi come TR-34.](#) Quando si inviano chiavi da AWS Payment Cryptography, viene considerato KDH.

KIF

Un Key Injection Facility (KIF) è una struttura sicura utilizzata per inizializzare i terminali di pagamento, incluso il loro caricamento con chiavi di crittografia.

KRD

[Un dispositivo di ricezione delle chiavi \(KRD\) è un dispositivo che riceve chiavi in un processo di scambio di chiavi come TR-34.](#) Quando si inviano chiavi a AWS Payment Cryptography, questa viene considerata KRD.

KSN

Un Key Serial Number (KSN) è un valore utilizzato come input per la crittografia/decrittografia DUKPT per creare chiavi di crittografia uniche per transazione. Il KSN è in genere costituito da un identificatore BDK, un ID terminale semiunivoco e un contatore di transazioni che incrementa ogni transazione elaborata su un determinato terminale di pagamento.

PADELLA

Un numero di conto primario (PAN) è un identificatore univoco per un conto, ad esempio una carta di credito o di debito. In genere, la lunghezza è di 13-19 cifre. Le prime 6-8 cifre identificano la rete e la banca emittente.

Blocco PIN

Un blocco di dati contenente un PIN durante l'elaborazione o la trasmissione e altri elementi di dati. I formati dei blocchi PIN standardizzano il contenuto del blocco PIN e il modo in cui può

essere elaborato per recuperare il PIN. La maggior parte dei blocchi PIN è composta dal PIN e dalla lunghezza del PIN e spesso contiene parte o tutto il PAN. AWS Payment Cryptography supporta i formati ISO 9564-1 0, 1, 3 e 4. Il formato 4 è richiesto per le chiavi AES. Durante la verifica o la traduzione dei PIN, è necessario specificare il blocco PIN dei dati in entrata o in uscita.

POI

Point of Interaction (POI), spesso utilizzato anche come sinonimo di Point of Sale (POS), è il dispositivo hardware con cui il titolare della carta interagisce per presentare le proprie credenziali di pagamento. Un esempio di POI è il terminale fisico in una sede commerciale. [Per l'elenco dei terminali POI PCI PTS certificati, consulta il sito Web PCI.](#)

PSN

[Il PAN Sequence Number \(PSN\) è un valore numerico utilizzato per differenziare più carte emesse con lo stesso PAN.](#)

Chiavi pubbliche

Quando si utilizzano cifrari asimmetrici (RSA), la chiave pubblica è il componente pubblico di una coppia di chiavi pubblica-privata. La chiave pubblica può essere condivisa e distribuita alle entità che devono crittografare i dati per il proprietario della coppia di chiavi pubblica-privata. Per le operazioni di firma digitale, la chiave pubblica viene utilizzata per verificare la firma.

Chiave privata

Quando si utilizzano cifrari asimmetrici (RSA), la chiave privata è il componente privato di una coppia di chiavi pubblica-privata. La chiave privata viene utilizzata per decrittografare i dati o creare firme digitali. Analogamente alle chiavi simmetriche di crittografia dei AWS pagamenti, le chiavi private vengono create in modo sicuro dagli HSM. Vengono decrittografate solo nella memoria volatile dell'HSM e solo per il tempo necessario all'elaborazione della richiesta crittografica.

PVV

Il valore di verifica del pin (PVV) è un valore derivato algebricamente da una serie di input come il [numero della carta](#) e il PIN, che genera un valore che può essere utilizzato per la successiva convalida. Uno di questi schemi è noto come Visa PVV (noto anche come metodo ABA) sebbene venga utilizzato per i PIN su qualsiasi rete.

Wrap/Unwrap con RSA

RSA wrap utilizza una chiave asimmetrica per avvolgere una chiave simmetrica (ad esempio una chiave TDES) per la trasmissione a un altro sistema. Solo il sistema con la chiave privata corrispondente può decrittografare il payload e caricare la chiave simmetrica. Al contrario, RSA unwrap decrittograferà in modo sicuro una chiave crittografata utilizzando RSA e quindi caricherà la chiave nella crittografia di pagamento. AWS RSA wrap è un metodo di scambio di chiavi di basso livello che non trasmette chiavi in formato di blocco di chiavi e non utilizza la firma del payload da parte della parte mittente. È necessario prendere in considerazione controlli alternativi per accertare la provvidenza e che gli attributi chiave non siano mutati.

TR-34 utilizza inoltre RSA internamente, ma è un formato separato e non è interoperabile.

TR-31

TR-31 (definito formalmente come ANSI X9 TR 31) è un formato di blocco chiave definito dall'American National Standards Institute (ANSI) per supportare la definizione degli attributi chiave nella stessa struttura di dati dei dati chiave stessi. Il formato del blocco chiave TR-31 definisce un insieme di attributi chiave collegati alla chiave in modo che siano tenuti insieme. AWS Payment Cryptography utilizza termini standardizzati TR-31 laddove possibile per garantire la corretta separazione delle chiavi e lo scopo delle chiavi. [TR-31 è stato sostituito da ANSI X9.143-2022.](#)

TR-34

TR-34 è un'implementazione di ANSI X9.24-2 che descrive un protocollo per distribuire in modo sicuro chiavi simmetriche (come 3DES e AES) utilizzando tecniche asimmetriche (come RSA). AWS La crittografia dei pagamenti utilizza i metodi TR-34 per consentire l'importazione e l'esportazione sicure delle chiavi.

Servizi correlati

[AWS Key Management Service](#)

AWSServizio di gestione delle chiavi (AWSKMS) è un servizio gestito che semplifica la creazione e il controllo delle chiavi crittografiche utilizzate per proteggere i dati. AWS KMS utilizza moduli di sicurezza hardware (HSM) per proteggere e convalidare i tuoiAWSChiavi KMS.

AWS CloudHSM

AWS CloudHSM fornisce ai clienti istanze HSM dedicate per uso generico nel AWS Nuvola. AWS CloudHSM può fornire una varietà di funzioni crittografiche come la creazione di chiavi, la firma dei dati o la crittografia e la decrittografia dei dati.

Ulteriori informazioni

- Per conoscere i termini e i concetti utilizzati in AWS Crittografia dei pagamenti, vedi [AWS Concetti relativi alla crittografia dei pagamenti](#).
- Per informazioni su AWS API Payment Cryptography Control Plane, vedi [AWS Riferimento all'API Payment Cryptography Control Plane](#).
- Per informazioni su AWS API Data Plane per la crittografia dei pagamenti, vedi [AWS Riferimento all'API Data Plane per la crittografia dei pagamenti](#).
- Per informazioni tecniche dettagliate su come AWS La crittografia dei pagamenti utilizza la crittografia e protegge AWS Chiavi di crittografia dei pagamenti, vedi [Dettagli crittografici](#).

Endpoint per AWS Payment Cryptography

Per connetterti a livello di codice AWS Payment Cryptography, usi un endpoint, l'URL del punto di ingresso del servizio. Gli AWS SDK e gli strumenti della riga di comando utilizzano automaticamente l'endpoint predefinito per il servizio in Regione AWS base al contesto regionale di una richiesta, quindi in genere non è necessario impostare in modo esplicito questi valori. Se necessario, puoi specificare un endpoint diverso per le tue richieste API.

Endpoint del piano di controllo

Nome regione	Regione	Endpoint	Protocollo
Stati Uniti orientali (Virginia settentrionale)	us-east-1	piano di controllo. payment-cryptography.us-east-1.amazonaws.com	HTTPS
Stati Uniti orientali (Ohio)	us-east-2	piano di controllo. payment-cryptography.us-east-2.amazonaws.com	HTTPS

Nome regione	Regione	Endpoint	Protocollo
US West (Oregon)	us-west-2	piano di controllo. payment-cryptography.us-west-2.amazonaws.com	HTTPS

Endpoint del piano dati

Nome regione	Regione	Endpoint	Protocollo
Stati Uniti orientali (Virginia settentrionale)	us-east-1	piano dati. payment-cryptography.us-east-1.amazonaws.com	HTTPS
Stati Uniti orientali (Ohio)	us-east-2	piano dati. payment-cryptography.us-east-2.amazonaws.com	HTTPS
US West (Oregon)	us-west-2	piano dati. payment-cryptography.us-west-2.amazonaws.com	HTTPS

Guida introduttiva alla crittografia AWS dei pagamenti

Per iniziare con la crittografia dei AWS pagamenti, dovrai prima creare delle chiavi e poi utilizzarle in varie operazioni crittografiche. Il tutorial seguente fornisce un semplice esempio di generazione di una chiave da utilizzare per generare/verificare i valori CVV2. [Per provare altri esempi ed esplorare i modelli di distribuzione all'interno di AWS, prova il seguente Workshop sulla crittografia dei AWS pagamenti o esplora il nostro progetto di esempio disponibile su Github](#)

Questo tutorial ti guida nella creazione di una singola chiave e nell'esecuzione di operazioni crittografiche utilizzando la chiave. Successivamente, elimini la chiave se non la desideri più, il che completa il ciclo di vita della chiave.

Argomenti

- [Prerequisiti](#)
- [Fase 1: Creare una chiave](#)
- [Passaggio 2: generare un valore CVV2 utilizzando la chiave](#)
- [Fase 3: Verificare il valore generato nel passaggio 2](#)
- [Fase 4: Eseguire un test negativo](#)
- [Fase 5: \(Facoltativo\) Pulizia](#)

Prerequisiti

Prima di iniziare, assicurati che:

- Hai il permesso di accedere al servizio. Per ulteriori informazioni, consulta [le politiche IAM](#).
- Hai [AWS CLI](#) installato. Puoi anche utilizzare [AWSSDK](#) o [AWSAPI](#) per accedere alla crittografia dei AWS pagamenti, ma le istruzioni di questo tutorial utilizzano il. AWS CLI

Fase 1: Creare una chiave

Il primo passo è creare una chiave. Per questo tutorial, crei una chiave [CVK](#) 3DES (2KEY TDES) a doppia lunghezza per generare e verificare i valori CVV/CVV2.

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

```

--exportable
--key-attributes KeyAlgorithm=TDDES_2KEY,KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,\
  KeyClass=SYMMETRIC_KEY,\
  KeyModesOfUse='{Generate=true,Verify=true}'

```

La risposta richiama i parametri della richiesta, tra cui un ARN per le chiamate successive e un Key Check Value (KCV).

```

{
  "Key": {
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
    tqv5yij6wtxx64pi",
    "KeyAttributes": {
      "KeyUsage": "TR31_C0_CARD_VERIFICATION_KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyAlgorithm": "TDDES_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"
}

```

Prendi nota di *KeyArn* ciò che rappresenta la chiave, ad esempio `arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi`. Ne hai bisogno nel passaggio successivo.

Passaggio 2: generare un valore CVV2 utilizzando la chiave

In questo passaggio, si genera un CVV2 per una determinata data di scadenza utilizzando la [PAN](#) chiave del passaggio 1.

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

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

Prendi nota del numero 144cardDataValue, in questo caso a 3 cifre. Ne hai bisogno nella fase successiva.

Fase 3: Verificare il valore generato nel passaggio 2

In questo esempio, convalidi il CVV2 del passaggio 2 utilizzando la chiave creata nel passaggio 1.

Esegui il comando seguente per convalidare il CVV2.

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

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

```
}
```

Il servizio restituisce una risposta HTTP di 200 per indicare che ha convalidato il CVV2.

Fase 4: Eseguire un test negativo

In questo passaggio, si crea un test negativo in cui il CVV2 non è corretto e non viene convalidato. Si tenta di convalidare un CVV2 errato utilizzando la chiave creata nel passaggio 1. Si tratta di un'operazione prevista, ad esempio se il titolare della carta ha inserito un codice CVV2 errato al momento del pagamento.

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

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

Il servizio restituisce una risposta HTTP di 400 con il messaggio «Verifica dei dati di convalida della carta non riuscita» e un motivo di INVALID_VALIDATION_DATA.

Fase 5: (Facoltativo) Pulizia

Ora puoi eliminare la chiave che hai creato nel passaggio 1. Per ridurre al minimo le modifiche irrecuperabili, il periodo di eliminazione della chiave predefinito è di sette giorni.

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

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

```
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
    tqv5yij6wtxx64pi",
    "KeyAttributes": {
      "KeyAlgorithm": "TDES_3KEY",
      "KeyClass": "SYMMETRIC_KEY",
      "KeyModesOfUse": {
        "Decrypt": true,
        "DeriveKey": false,
        "Encrypt": true,
        "Generate": false,
        "NoRestrictions": false,
        "Sign": false,
        "Unwrap": true,
        "Verify": false,
        "Wrap": true
      },
      "KeyUsage": "TR31_P0_PIN_ENCRYPTION_KEY"
    },
    "KeyCheckValue": "CADD1",
    "KeyCheckValueAlgorithm": "ANSI_X9_24",
    "KeyOrigin": "AWS_PAYMENT_CRYPTOGRAPHY",
    "KeyState": "DELETE_PENDING",
    "UsageStartTimestamp": "2022-10-27T08:27:51.753000-07:00"
  }
}
```

Prendi nota di due campi nell'output. Per impostazione predefinita, `deletePendingTimestamp` è impostato su sette giorni nel futuro. `KeyState` è impostato su `DELETE_PENDING`. È possibile annullare questa eliminazione in qualsiasi momento prima dell'orario di eliminazione pianificato [restore-key](#) chiamando.

Gestione delle chiavi

Per iniziare con la crittografia dei AWS pagamenti, ti consigliamo di creare una chiave di crittografia dei AWS pagamenti.

Gli argomenti di questa sezione spiegano come creare e gestire diversi tipi di chiavi di crittografia dei AWS pagamenti, dalla creazione all'eliminazione. Include argomenti sulla creazione, la modifica e la visualizzazione delle chiavi, l'etichettatura delle chiavi, la creazione di alias chiave e l'attivazione e la disabilitazione delle chiavi.

Argomenti

- [Generazione delle chiavi](#)
- [Elenca le chiavi](#)
- [Abilitazione e disabilitazione delle chiavi](#)
- [Eliminazione delle chiavi](#)
- [Importa ed esporta chiavi](#)
- [Utilizzo di alias](#)
- [Procurati le chiavi](#)
- [Chiavi di tagging](#)
- [Comprensione degli attributi chiave della chiave Payment Cryptography AWS](#)

Generazione delle chiavi

È possibile creare chiavi AWS di crittografia dei pagamenti utilizzando l'operazione API. CreateKey Durante questo processo, specificherete vari attributi della chiave o dell'output risultante, come l'algoritmo chiave (ad esempio, TDES_3KEY), il (ad esempio TR31_P0_PIN_ENCRYPTION_KEY), le operazioni consentite KeyUsage (ad esempio, crittografia, firma) e se è esportabile. Non è possibile modificare queste proprietà dopo la creazione della chiave Payment Cryptography. AWS

Esempi

- [Generazione di una chiave TDES 2KEY](#)
- [Generazione di una chiave di crittografia PIN](#)
- [Crea una chiave asimmetrica \(RSA\)](#)

- [Generazione di una chiave PVV \(PVV\)](#)

Generazione di una chiave TDES 2KEY

Example

Questo comando genera una chiave TDES 2KEY allo scopo di generare e verificare i valori CVV/ CVV2. La risposta riporta i parametri della richiesta, incluso un ARN per le chiamate successive e un KCV (Key Check Value).

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

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

```

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

```

Generazione di una chiave di crittografia PIN

Example Generazione di una chiave di crittografia PIN (PEK)

Questo comando genera una chiave TDES 3KEY allo scopo di crittografare i valori PIN (nota come chiave di crittografia PIN). Questa chiave può essere utilizzata per proteggere i PIN archiviati o per decrittografare i PIN forniti durante un tentativo di verifica, ad esempio durante una transazione. La risposta riporta i parametri della richiesta, incluso un ARN per le chiamate successive e 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/
kwapwa6qaiflw2h",
    "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"
}
}

```

Crea una chiave asimmetrica (RSA)

Example

In questo esempio, genereremo una nuova key pair asimmetrica RSA a 2048 bit. Verrà generata una nuova chiave privata e la chiave pubblica corrispondente. La chiave pubblica può essere recuperata utilizzando 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"
}
}

```

Generazione di una chiave PVV (PVV)

Example

Questo comando genera una chiave TDES 3KEY allo scopo di generare valori PVV (noti come PIN Verification Value). È possibile utilizzare questa chiave per generare un valore PVV che può essere confrontato con un PVV calcolato successivamente. La risposta riporta i parametri della richiesta, incluso un ARN per le chiamate successive e 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"
  }
}

```

Elenca le chiavi

List Keys presenta un elenco di tasti accessibili al chiamante in questo account e in questa regione.

Example

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

```

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

```

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

Abilitazione e disabilitazione delle chiavi

Puoi disabilitare e riattivare le chiavi di crittografia dei AWS pagamenti. Quando si crea una chiave, questa è abilitata per impostazione predefinita. Se si disabilita una chiave, questa non può essere utilizzata in alcuna [operazione di crittografia](#) finché non viene riattivata. I comandi di avvio/ interruzione dell'utilizzo hanno effetto immediato, quindi si consiglia di esaminarne l'utilizzo prima di apportare tali modifiche. È inoltre possibile impostare una modifica (avvio o interruzione dell'utilizzo) in modo che abbia effetto in futuro utilizzando il `timestamp` parametro opzionale.

Poiché è temporanea e facilmente annullabile, la disattivazione di una chiave di crittografia dei AWS pagamenti è un'alternativa più sicura all'eliminazione di una chiave di crittografia dei AWS pagamenti, un'azione distruttiva e irreversibile. Se stai pensando di eliminare una chiave AWS di crittografia dei pagamenti, disattivala prima e assicurati di non dover utilizzare la chiave per crittografare o decrittografare i dati in futuro.

Argomenti

- [Inizia l'utilizzo della chiave](#)
- [Interrompi l'uso delle chiavi](#)

Inizia l'utilizzo della chiave

L'utilizzo della chiave deve essere abilitato per poter utilizzare una chiave per le operazioni crittografiche. Se una chiave non è abilitata, è possibile utilizzare questa operazione per renderla utilizzabile. Il campo `UsageStartTimestamp` rappresenterà quando la chiave è diventata/diventerà attiva. Ciò avverrà in passato per un token abilitato e in futuro se in attesa di attivazione.

Example

In questo esempio, viene richiesta l'attivazione di una chiave per l'utilizzo della chiave. La risposta include le informazioni chiave e il flag `enable` è stato convertito a `true`. Ciò si rifletterà anche nell'oggetto di risposta `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"
  }
}
```

Interrompi l'uso delle chiavi

Se non intendi più utilizzare una chiave, puoi interromperne l'utilizzo per impedire ulteriori operazioni crittografiche. Questa operazione non è permanente, quindi è possibile invertirla [avviando l'utilizzo della chiave](#). Puoi anche impostare una chiave in modo che venga disattivata in futuro. Il campo `UsageStopTimestamp` rappresenterà quando la chiave è diventata/diventerà disabilitata.

Example

In questo esempio, viene richiesto di interrompere l'utilizzo delle chiavi in futuro. Dopo l'esecuzione, questa chiave non può essere utilizzata per operazioni crittografiche a meno che non venga riattivata tramite l'[utilizzo della chiave di avvio](#). La risposta include le informazioni sulla chiave e il flag di abilitazione è stato spostato su `false`. Ciò si rifletterà anche nell'oggetto di risposta `list-keys`.

```
$ aws payment-cryptography stop-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": false,
    "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",  
"UsageStopTimestamp": "2022-10-27T14:09:59.468000-07:00"  
}  
}
```

Eliminazione delle chiavi

L'eliminazione di una chiave AWS di crittografia di pagamento elimina il materiale chiave e tutti i metadati associati alla chiave ed è irreversibile a meno che una copia della chiave non sia disponibile al di fuori di Payment Cryptography. AWS Dopo l'eliminazione di una chiave, non è più possibile decrittografare i dati crittografati con quella chiave, il che significa che i dati potrebbero diventare irrecuperabili. È consigliabile eliminare una chiave solo quando si è certi di non averne più bisogno e che nessun'altra parte stia utilizzando questa chiave. Se non sei sicuro, valuta la possibilità di disabilitare la chiave anziché eliminarla. Puoi riattivare una chiave disabilitata se devi riutilizzarla in un secondo momento, ma non puoi recuperare una chiave di crittografia dei AWS pagamenti eliminata a meno che non sia possibile reimportarla da un'altra fonte.

Prima di eliminare una chiave, assicurati di non averne più bisogno. AWS La crittografia dei pagamenti non memorizza i risultati di operazioni crittografiche come CVV2 e non è in grado di determinare se è necessaria una chiave per qualsiasi materiale crittografico persistente.

AWS Payment Cryptography non elimina mai le chiavi appartenenti agli AWS account attivi, a meno che non ne venga esplicitamente pianificata l'eliminazione e che il periodo di attesa obbligatorio scada.

Tuttavia, potresti scegliere di eliminare una chiave di crittografia dei AWS pagamenti per uno o più dei seguenti motivi:

- Per completare il ciclo di vita di una chiave che non ti serve più

- Per evitare il sovraccarico di gestione associato alla manutenzione delle chiavi di crittografia dei pagamenti non utilizzate AWS

Note

Se [chiudi o elimini la tua Account AWS](#) chiave di crittografia dei AWS pagamenti diventa inaccessibile. Non è necessario pianificare l'eliminazione della chiave di crittografia dei AWS pagamenti separatamente dalla chiusura dell'account.

AWS Payment Cryptography registra una voce nel [AWS CloudTrail](#) registro quando pianifichi l'eliminazione della chiave di crittografia dei AWS pagamenti e quando la chiave di crittografia dei AWS pagamenti viene effettivamente eliminata.

Informazioni sul periodo di attesa

Poiché l'eliminazione di una chiave è irreversibile, AWS Payment Cryptography richiede di impostare un periodo di attesa compreso tra 3 e 180 giorni. Il periodo di attesa predefinito è di sette giorni.

Tuttavia, il periodo di attesa effettivo potrebbe essere fino a 24 ore più lungo di quello pianificato. Per ottenere la data e l'ora effettive in cui la chiave AWS di crittografia dei pagamenti verrà eliminata, utilizza le GetKey operazioni. Assicurati di segnare il fuso orario.

Durante il periodo di attesa, lo stato e lo stato della chiave AWS Payment Cryptography sono In attesa di eliminazione.

Note

[Una chiave AWS di crittografia dei pagamenti in attesa di eliminazione non può essere utilizzata in alcuna operazione crittografica.](#)

Al termine del periodo di attesa, AWS Payment Cryptography elimina la chiave Payment Cryptography, i AWS relativi alias e tutti i relativi metadati di crittografia dei pagamenti. AWS

Utilizza il periodo di attesa per assicurarti di non aver bisogno della chiave AWS di crittografia dei pagamenti ora o in futuro. Se ritieni di aver bisogno della chiave durante il periodo di attesa, puoi annullare l'eliminazione della chiave prima della fine del periodo di attesa. Al termine del periodo di attesa, non è possibile annullare l'eliminazione della chiave e il servizio elimina la chiave.

Example

In questo esempio, viene richiesta l'eliminazione di una chiave. Oltre alle informazioni chiave di base, due campi rilevanti sono che lo stato della chiave è stato modificato in `DELETE_PENDING` e `deletePendingTimestamp` indica quando è attualmente pianificata l'eliminazione della chiave.

```
$ 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"  
  }  
}
```

```
}
```

Example

In questo esempio, un'eliminazione in sospeso viene annullata. Una volta completata con successo, una chiave non verrà più eliminata secondo la pianificazione precedente. La risposta contiene le informazioni chiave di base; inoltre, sono stati modificati due campi pertinenti: `KeyState` ed `deletePendingTimestamp`. `KeyState` viene restituito al valore `CREATE_COMPLETE`, mentre `DeletePendingTimestamp` viene rimosso.

```
$ aws payment-cryptography restore-key --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qai1lw2h
```

```
{
  "Key": {
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qai1lw2h",
    "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"
  }
}
```

```
}  
}
```

Importa ed esporta chiavi

AWS Le chiavi di crittografia dei pagamenti possono essere importate da altre soluzioni o esportate in altre soluzioni (come altri HSM). È un caso d'uso comune lo scambio di chiavi con i fornitori di servizi utilizzando funzionalità di importazione ed esportazione. In quanto servizio cloud, AWS Payment Cryptography adotta un approccio elettronico moderno alla gestione delle chiavi, aiutandovi al contempo a mantenere la conformità e i controlli applicabili. L'obiettivo a lungo termine è quello di abbandonare i componenti chiave cartacei per passare a strumenti elettronici di scambio di chiavi basati su standard.

Scambio di chiavi di crittografia (KEK)

AWS [La crittografia dei pagamenti incoraggia l'uso della crittografia a chiave pubblica \(RSA\) per lo scambio iniziale di chiavi utilizzando la consolidata norma ANSI X9.24 TR-34](#). I nomi comuni per questo tipo di chiave iniziale includono Key Encryption Key (KEK), Zone Master Key (ZMK) e Zone Control Master Key (ZCMK). [Se i tuoi sistemi o partner non sono ancora in grado di supportare TR-34, puoi anche prendere in considerazione l'utilizzo di RSA Wrap/Unwrap](#).

Se hai bisogno di continuare a elaborare i componenti chiave cartacei fino a quando tutti i partner non supporteranno lo scambio di chiavi elettroniche, puoi prendere in considerazione la possibilità di mantenere un HSM offline per questo scopo.

Note

[Se desideri importare le tue chiavi di test, dai un'occhiata al progetto di esempio su Github](#). Per istruzioni su come importare/esportare chiavi da altre piattaforme, consulta la guida per l'utente di tali piattaforme.

Working Key (WK) Exchange

AWS La crittografia dei pagamenti utilizza la norma di settore pertinente ([ANSI X9.24 TR 31-2018](#)) per lo scambio di chiavi funzionanti. TR-31 presuppone che una KEK sia stata precedentemente scambiata. Ciò è coerente con il requisito del PIN PCI di associare crittograficamente il materiale chiave al tipo di chiave e al suo utilizzo in ogni momento. Le chiavi di lavoro hanno nomi diversi, tra cui chiavi di lavoro dell'acquirente, chiavi di lavoro dell'emittente, BDK, IPEK, ecc.

Argomenti

- [Chiavi di importazione](#)
- [Chiavi di esportazione](#)

Chiavi di importazione

Important

Gli esempi possono richiedere la versione più recente di AWS CLI V2. [Prima di iniziare, assicurati di aver effettuato l'aggiornamento alla versione più recente.](#)

Argomenti

- [Importazione di chiavi simmetriche](#)
- [Importazione di chiavi asimmetriche \(RSA\)](#)

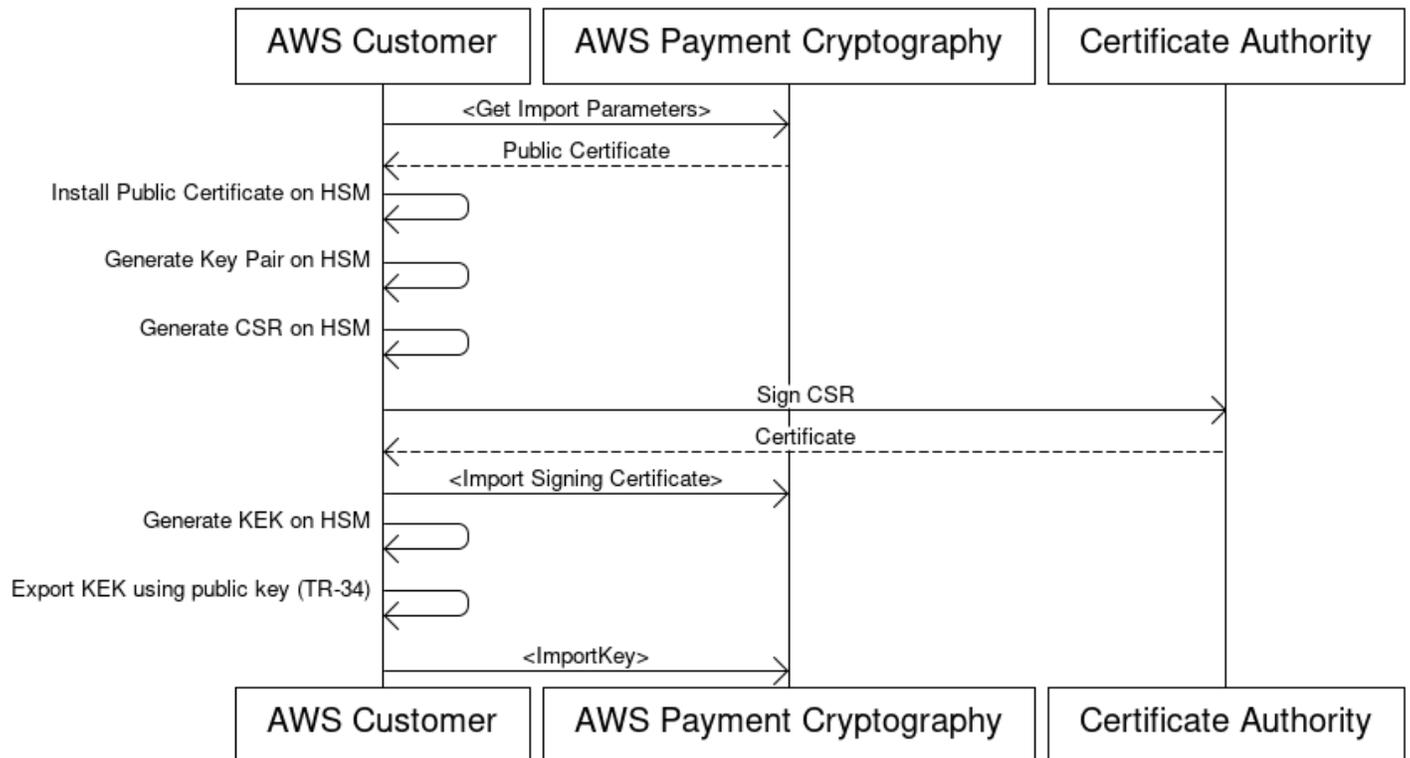
Importazione di chiavi simmetriche

Argomenti

- [Importazione di chiavi utilizzando tecniche asimmetriche \(TR-34\)](#)
- [Importa le chiavi utilizzando tecniche asimmetriche \(RSA Unwrap\)](#)
- [Importa chiavi simmetriche utilizzando una chiave di scambio di chiavi prestabilita \(TR-31\)](#)

Importazione di chiavi utilizzando tecniche asimmetriche (TR-34)

Key Encryption Key(KEK) Import Process



Panoramica: TR-34 utilizza la crittografia asimmetrica RSA per crittografare le chiavi simmetriche per lo scambio e garantire l'origine dei dati (firma). Ciò garantisce sia la riservatezza (crittografia) che l'integrità (firma) della chiave incapsulata.

Se desideri importare le tue chiavi, dai un'occhiata al progetto di esempio su [Github](#). Per istruzioni su come importare/esportare chiavi da altre piattaforme, consulta la guida per l'utente di tali piattaforme.

1. Chiamate il comando `initialize import`

Chiama `get-parameters-for-import` per inizializzare il processo di importazione. Questa API genererà una coppia di chiavi ai fini dell'importazione delle chiavi, firmerà la chiave e restituirà il certificato e la radice del certificato. In definitiva, la chiave da esportare dovrebbe essere crittografata utilizzando questa chiave. Nella terminologia TR-34, questo è noto come KRD Cert. Si noti che questi certificati sono di breve durata e sono destinati esclusivamente a questo scopo.

2. Installa il certificato pubblico sul sistema di origine delle chiavi

Con molti HSM, potrebbe essere necessario installare/caricare/considerare attendibile il certificato pubblico generato nel passaggio 1 per esportare le chiavi utilizzandolo.

3. Genera la chiave pubblica e fornisci la radice del certificato a Payment Cryptography AWS

Per garantire l'integrità del payload trasmesso, questo viene firmato dalla parte mittente (nota come Key Distribution Host o KDH). La parte mittente vorrà generare una chiave pubblica per questo scopo e quindi creare un certificato a chiave pubblica (X509) che possa essere restituito a Payment Cryptography. AWS Private CA è un'opzione per generare certificati, ma non ci sono restrizioni sull'autorità di certificazione utilizzata.

Una volta ottenuto il certificato, ti consigliamo di caricare il certificato principale su AWS Payment Cryptography usando il `importKey` comando and `KeyMaterialType` of `ROOT_PUBLIC_KEY_CERTIFICATE` e `KeyUsageType` of `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`.

4. Esporta la chiave dal sistema sorgente

Molti HSM e sistemi correlati supportano la possibilità di esportare le chiavi utilizzando la norma TR-34. Ti consigliamo di specificare la chiave pubblica del passaggio 1 come certificato KRD (crittografia) e la chiave del passaggio 3 come certificato KDH (firma). Per effettuare l'importazione in AWS Payment Cryptography, dovrai specificare che il formato sia il formato a due passaggi TR-34.2012 non CMS, che può anche essere chiamato formato TR-34 Diebold.

5. Chiave di importazione delle chiamate

Come ultimo passaggio, chiamerai l'API `ImportKey` con un `KeyMaterialType` di `TR34_KEY_BLOCK_certificate-authority-public-key-identifier`. Sarà il `KeyArn` della CA principale importata nella fase 3, `key-material` sarà il materiale chiave racchiuso nel passaggio 4 ed `signing-key-certificate` è il certificato foglia del passaggio 3. Sarà inoltre necessario fornire il token di importazione del passaggio 1.

6. Utilizza la chiave importata per le operazioni crittografiche o la successiva importazione

Se la chiave importata `KeyUsage` era `TR31_K0_KEY_ENCRYPTION_KEY`, questa chiave può essere utilizzata per le successive importazioni di chiavi utilizzando TR-31. Se il tipo di chiave era di qualsiasi altro tipo (ad esempio `TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`), la chiave può essere utilizzata direttamente per operazioni crittografiche.

Importa le chiavi utilizzando tecniche asimmetriche (RSA Unwrap)

Panoramica: AWS Payment Cryptography supporta RSA wrap/unwrap per lo scambio di chiavi quando TR-34 non è possibile. Simile a TR-34, questa tecnica utilizza la crittografia asimmetrica RSA per crittografare le chiavi simmetriche per lo scambio. Tuttavia, a differenza di TR-34, questo metodo

non ha il payload firmato dalla parte mittente. Inoltre, questa tecnica di wrap RSA non mantiene l'integrità dei metadati chiave durante il trasferimento, in quanto non include i blocchi chiave.

Note

Il wrap RSA può essere utilizzato per importare o esportare chiavi TDES e AES-128.

1. Chiama il comando di inizializzazione dell'importazione

Chiama `get-parameters-for-import` per inizializzare il processo di importazione con un tipo di materiale chiave `KEY_CRYPTOGRAM`. `WrappingKeyAlgorithm` può essere `RSA_2048` quando si scambiano chiavi TDES. È possibile utilizzare `RSA_3072` o `RSA_4096` per lo scambio di chiavi TDES o AES-128. Questa API genererà una coppia di chiavi ai fini dell'importazione delle chiavi, firmerà la chiave utilizzando una radice del certificato e restituirà sia il certificato che la radice del certificato. In definitiva, la chiave da esportare dovrebbe essere crittografata utilizzando questa chiave. Tieni presente che questi certificati sono di breve durata e sono destinati esclusivamente a questo scopo.

```
$ 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. Installa il certificato pubblico sul sistema di origine delle chiavi

Con molti HSM, potrebbe essere necessario installare/caricare/considerare attendibile il certificato pubblico (e/o la relativa radice) generato nel passaggio 1 per esportare le chiavi utilizzandolo.

3. Esporta la chiave dal sistema sorgente

Molti HSM e sistemi correlati supportano la possibilità di esportare le chiavi utilizzando RSA wrap. Ti consigliamo di specificare la chiave pubblica del passaggio 1 come certificato (di crittografia)

(WrappingKeycertificato). Se hai bisogno della catena di fiducia, questa è contenuta nel campo di risposta WrappingKeyCertificateChain nel passaggio #1. Quando esporti la chiave dal tuo HSM, ti consigliamo di specificare che il formato sia RSA, Padding Mode = PKCS #1 v2.2 OAEP (con SHA 256 o SHA 512).

4. Chiave di importazione delle chiamate

Come ultimo passaggio, chiamerai l'API ImportKey con un KeyMaterialType di KeyMaterial. Avrai bisogno del token di importazione del passaggio 1 e del key-material (materiale chiave incartato) del passaggio 3. Dovrai fornire i parametri chiave (come l'utilizzo delle chiavi) poiché RSA wrap non utilizza blocchi chiave.

```
$ 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
```

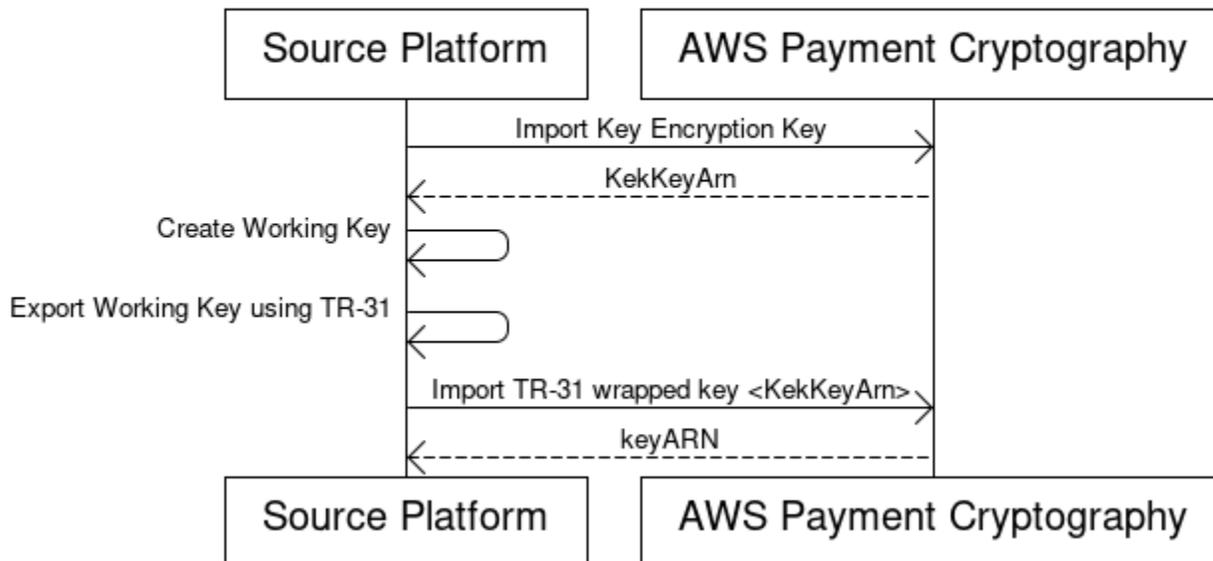
```
{
  "Key": {
    "KeyOrigin": "EXTERNAL",
    "Exportable": true,
    "KeyCheckValue": "DA1ACF",
    "UsageStartTimestamp": 1697643478.92,
    "Enabled": true,
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaiifllw2h",
    "CreateTimestamp": 1697643478.92,
    "KeyState": "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. Utilizza la chiave importata per le operazioni crittografiche o l'importazione successiva

Se la chiave importata `KeyUsage` era `TR31_K0_KEY_ENCRYPTION_KEY`, questa chiave può essere utilizzata per le successive importazioni di chiavi utilizzando `TR-31`. Se il tipo di chiave era di qualsiasi altro tipo (ad esempio `TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY`), la chiave può essere utilizzata direttamente per operazioni crittografiche.

Importa chiavi simmetriche utilizzando una chiave di scambio di chiavi prestabilita (TR-31)

Import symmetric keys using a pre-established key exchange key (TR-31)



Quando i partner si scambiano più chiavi (o supportano la rotazione delle chiavi), è normale scambiare prima una chiave di crittografia a chiave iniziale (KEK) utilizzando tecniche come componenti chiave cartacei o, nel caso della crittografia dei AWS pagamenti, utilizzando TR-34.

Una volta stabilita una KEK, è possibile utilizzare questa chiave per trasportare le chiavi successive (incluse altre KEK). AWS Payment Cryptography supporta questo tipo di scambio di chiavi utilizzando ANSI TR-31, ampiamente utilizzato e ampiamente supportato dai fornitori di HSM.

1. Chiave di crittografia a chiave di importazione (KEK)

Si presume che abbiate già importato la vostra KEK e che abbiate KeyArn (o KeyAlias) a vostra disposizione.

2. Crea una chiave sulla piattaforma di origine

Se la chiave non esiste già, crea la chiave sulla piattaforma di origine. Al contrario, puoi creare la chiave su AWS Payment Cryptography e utilizzare invece il `export` comando.

3. Esporta la chiave dalla piattaforma di origine

Durante l'esportazione, assicuratevi di specificare il formato di esportazione come TR-31. La piattaforma di origine ti chiederà anche la chiave da esportare e la chiave di crittografia da utilizzare.

4. Importazione in AWS Payment Cryptography

Quando si chiama il comando `ImportKey`, `WrappingKeyId` deve essere il `keyArn` (o alias) della chiave di crittografia della chiave ed `WrappedKeyBlock` è l'output della piattaforma di origine.

Example

```
$ aws payment-cryptography import-key \
    --key-material="Tr31KeyBlock={WrappingKeyId="arn:aws:payment-
cryptography:us-east-2:111122223333:key/ov6icy4ryas4zcza"},\
    WrappedKeyBlock="D0112B0AX00E00002E0A3D58252CB67564853373D1EBCC1E23B2ADE7B15E967CC27B85D599"
```

```
{
  "Key": {
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
kwapwa6qaiifllw2h",
    "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"
```

```
}
}
```

Importazione di chiavi asimmetriche (RSA)

Importazione di chiavi pubbliche RSA

AWS Payment Cryptography supporta l'importazione di chiavi RSA pubbliche sotto forma di certificati X.509. Per importare un certificato, è necessario prima importare il relativo certificato radice. Tutti i certificati non devono essere scaduti al momento dell'importazione. Il certificato deve essere in formato PEM e deve essere codificato in base 64.

1. Importazione in Root Certificate in Payment Cryptography AWS

Example

```
$ aws payment-cryptography import-key \
  --key-material='{"RootCertificatePublicKey":{"KeyAttributes":
{"KeyAlgorithm":"RSA_2048", \
  "KeyClass":"PUBLIC_KEY", "KeyModesOfUse":{"Verify":
true},"KeyUsage":"TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"}, \
  "PublicKeyCertificate":"LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSURKVENDQWcyZ0F3SUJBZ01CWkR"
```

```
{
  "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. Importa il certificato a chiave pubblica nella crittografia AWS dei pagamenti

Ora puoi importare una chiave pubblica. Esistono due opzioni per importare le chiavi pubbliche. TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE può essere utilizzato se lo scopo della chiave è verificare le firme (ad esempio quando si importa utilizzando TR-34). TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION può essere utilizzato per crittografare dati destinati all'uso con un altro sistema.

Example

```

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

```

```

{
  "Key": {
    "CreateTimestamp": "2023-08-08T18:55:46.815000+00:00",
    "Enabled": true,
    "KeyArn": "arn:aws:payment-cryptography:us-
east-2:111122223333:key/4kd6xud22e64wcbk",
    "KeyAttributes": {
      "KeyAlgorithm": "RSA_4096",
      "KeyClass": "PUBLIC_KEY",
      "KeyModesOfUse": {

```

```
        "Decrypt": false,  
        "DeriveKey": false,  
        "Encrypt": false,  
        "Generate": false,  
        "NoRestrictions": false,  
        "Sign": false,  
        "Unwrap": false,  
        "Verify": true,  
        "Wrap": false  
    },  
    "KeyUsage": "TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE"  
},  
"KeyOrigin": "EXTERNAL",  
"KeyState": "CREATE_COMPLETE",  
"UsageStartTimestamp": "2023-08-08T18:55:46.815000+00:00"  
}  
}
```

Chiavi di esportazione

Argomenti

- [Esportazione di chiavi simmetriche](#)
- [Esportazione di chiavi asimmetriche \(RSA\)](#)

Esportazione di chiavi simmetriche

Important

Gli esempi possono richiedere la versione più recente di AWS CLI V2. [Prima di iniziare, assicurati di aver effettuato l'aggiornamento alla versione più recente.](#)

Argomenti

- [Esportazione delle chiavi utilizzando tecniche asimmetriche \(TR-34\)](#)
- [Esporta le chiavi utilizzando tecniche asimmetriche \(RSA Wrap\)](#)
- [Esporta le chiavi simmetriche utilizzando una chiave di scambio di chiavi prestabilita \(TR-31\)](#)
- [Esporta le chiavi iniziali DUKPT \(IPEK/IK\)](#)

Esportazione delle chiavi utilizzando tecniche asimmetriche (TR-34)

Panoramica: TR-34 utilizza la crittografia asimmetrica RSA per crittografare le chiavi simmetriche per lo scambio e garantire l'origine dei dati (firma). Ciò garantisce sia la riservatezza (crittografia) che l'integrità (firma) della chiave incapsulata. Durante l'esportazione, AWS Payment Cryptography diventa l'host di distribuzione delle chiavi (KDH) e il sistema di destinazione diventa il dispositivo di ricezione delle chiavi (KRD).

1. Chiama il comando di inizializzazione dell'esportazione

Chiama `get-parameters-for-export` per inizializzare il processo di esportazione. Questa API genererà una coppia di chiavi ai fini dell'esportazione delle chiavi, firmerà la chiave e restituirà il certificato e la radice del certificato. In definitiva, la chiave privata generata da questo comando viene utilizzata per firmare il payload di esportazione. Nella terminologia TR-34, questo è noto come certificato di firma KDH. Si noti che questi certificati sono di breve durata e sono destinati esclusivamente a questo scopo. Il parametro `ParametersValidUntilTimestamp` specifica la loro durata.

NOTA: tutti i certificati vengono restituiti in un formato codificato base64

Example

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

```
{
  "SigningKeyCertificate":
  "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUV2RENDQXFTZ0F3SUJBZ0lRZFAzSzMNHNEFKT0I4WTNpTmUvY1
  "SigningKeyCertificateChain":
  "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURS0tLS0tCk1JSUY0VENDQThZ0F3SUJBZ0lSQUt1N2piaHFKZjJPd3FGUWI5c3
  "SigningKeyAlgorithm": "RSA_2048",
  "ExportToken": "export-token-au7pvkbsq4mbup6i",
  "ParametersValidUntilTimestamp": "2023-06-13T15:40:24.036000-07:00"
}
```

2. Importa il certificato AWS di crittografia dei pagamenti nel sistema di ricezione

Se necessario, importate la catena di certificati fornita nel passaggio 1 nel sistema di ricezione.

3. Genera una key pair, crea un certificato pubblico e fornisci la radice del certificato a AWS Payment Cryptography

Per garantire la riservatezza del payload trasmesso, questo viene crittografato dalla parte mittente (nota come Key Distribution Host o KDH). La parte ricevente (in genere il tuo HSM o l'HSM dei tuoi partner) vorrà generare una chiave pubblica a tale scopo e quindi creare un certificato a chiave pubblica (x.509) che può essere restituito a Payment Cryptography. AWS Private CA è un'opzione per la generazione di certificati, ma non ci sono restrizioni sull'autorità di certificazione utilizzata.

Una volta ottenuto il certificato, ti consigliamo di caricare il certificato principale su AWS Payment Cryptography usando il `ImportKey` comando and `KeyMaterialType` of `ROOT_PUBLIC_KEY_CERTIFICATE` e `KeyUsageType` of `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE`.

Il `KeyUsageType` certificato è `TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE` perché è la chiave principale e viene utilizzata per firmare il certificato foglia. I certificati Leaf per l'importazione/esportazione non vengono importati in Payment AWS Cryptography ma vengono passati in linea.

Note

Se il certificato radice è stato importato in precedenza, questo passaggio può essere saltato.

4. Chiama la chiave Export

Come ultimo passaggio, chiamerai l' `ExportKey` API con un `KeyMaterialType` di `TR34_KEY_BLOCK`. `certificate-authority-public-key-identifier` Sarà il keyARN dell'importazione della CA principale nel passaggio 3, `WrappingKeyCertificate` sarà il certificato leaf del passaggio 3 ed `export-key-identifier` è il keyARN (o alias) da esportare. Dovrai inoltre fornire il token di esportazione del passaggio 1.

Esporta le chiavi utilizzando tecniche asimmetriche (RSA Wrap)

Panoramica: AWS Payment Cryptography supporta RSA wrap/unwrap per lo scambio di chiavi quando TR-34 non è un'opzione disponibile dalla controparte. Simile a TR-34, questa tecnica utilizza la crittografia asimmetrica RSA per crittografare le chiavi simmetriche per lo scambio. Tuttavia, a

differenza di TR-34, questo metodo non ha il payload firmato dalla parte mittente. Inoltre, questa tecnica di wrap RSA non include i blocchi chiave utilizzati per mantenere l'integrità dei metadati chiave durante il trasporto.

Note

Il wrap RSA può essere utilizzato per esportare le chiavi TDES e AES-128.

1. Genera una chiave RSA e un certificato sul sistema ricevente

Crea (o identifica) una chiave RSA che verrà utilizzata per ricevere la chiave incapsulata. AWS Payment Cryptography richiede chiavi in formato certificato X.509. Il certificato deve essere firmato da un certificato radice importato (o che può essere importato) in Payment Cryptography. [AWS](#)

2. Installa il certificato pubblico principale su AWS Payment Cryptography

```
$ 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. Chiave di esportazione delle chiamate

Successivamente desideri indicare a AWS Payment Cryptography di esportare la tua chiave utilizzando il tuo certificato leaf. Specificherai l'ARN per il certificato radice importato in precedenza, il certificato foglia da utilizzare per l'esportazione e la chiave simmetrica da esportare. L'output sarà una versione (crittografata) con codifica esadecimale binaria della chiave simmetrica.

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

```

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

```

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

```

{
  "WrappedKey": {
    "KeyMaterial":
    "18874746731E9E1C4562E4116D1C2477063FCB08454D757D81854AEAEE0A52B1F9D303FA29C02DC82AE7785353

```

```
    "WrappedKeyMaterialFormat": "KEY_CRYPTOGRAM"  
  }  
}
```

4. Importa la chiave nel sistema ricevente

Molti HSM e sistemi correlati supportano la possibilità di importare chiavi utilizzando RSA unwrap (inclusa AWS Payment Cryptography). A tale scopo, specifica la chiave pubblica del passaggio 1 come certificato (di crittografia) e il formato deve essere specificato come RSA, Padding Mode = PKCS #1 v2.2 OAEP (con SHA 256). La terminologia esatta può variare in base all'HSM.

Note

AWS Payment Cryptography emette la chiave incapsulata in HexBinary. Potrebbe essere necessario convertire il formato prima dell'importazione se il sistema richiede una rappresentazione binaria diversa come base64.

Esporta le chiavi simmetriche utilizzando una chiave di scambio di chiavi prestabilita (TR-31)

[Quando i partner si scambiano più chiavi \(o supportano la rotazione delle chiavi\), è normale scambiare prima una chiave di crittografia a chiave iniziale \(KEK\) utilizzando tecniche come componenti chiave cartacei o, nel caso della crittografia dei AWS pagamenti, utilizzando TR-34.](#)

Una volta stabilita una KEK, è possibile utilizzare questa chiave per trasportare le chiavi successive (incluse altre KEK). AWS Payment Cryptography supporta questo tipo di scambio di chiavi utilizzando ANSI TR-31, ampiamente utilizzato e ampiamente supportato dai fornitori di HSM.

1. Chiave di crittografia con chiave di scambio (KEK)

Si presume che abbiate già scambiato la vostra KEK e che abbiate KeyArn (o KeyAlias) a vostra disposizione.

2. Crea una chiave sulla crittografia dei pagamenti AWS

Se la chiave non esiste già, crea la chiave. Al contrario, è possibile creare la chiave sull'altro sistema e utilizzare invece il comando [import](#).

3. Esporta la chiave da AWS Payment Cryptography

Durante l'esportazione, il formato sarà TR-31. Quando si chiama l'API, si specifica la chiave da esportare e la chiave di wrapping da utilizzare.

```
$ 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. Importa nel tuo sistema

Tu o il tuo partner utilizzerete l'implementazione della chiave di importazione sul vostro sistema per importare la chiave.

Esporta le chiavi iniziali DUKPT (IPEK/IK)

Quando si utilizza [DUKPT](#), è possibile generare una singola Base Derivation Key (BDK) per una flotta di terminali. I terminali, tuttavia, non hanno mai accesso al BDK originale, ma a ciascuno viene iniettata una chiave terminale iniziale unica nota come IPEK o Initial Key (IK). Ogni IPEK è una chiave derivata dal BDK ed è pensata per essere unica per terminale, ma deriva dal BDK originale. I dati di derivazione per questo calcolo sono noti come Key Serial Number (KSN). Per X9.24, per TDES il KSN a 10 byte è in genere composto da 24 bit per il Key Set ID, 19 bit per l'ID del terminale e 21 bit per il contatore delle transazioni. Per AES, il KSN a 12 byte è in genere composto da 32 bit per l'ID BDK, 32 bit per l'identificatore di derivazione (ID) e 32 bit per il contatore delle transazioni.

AWS La crittografia dei pagamenti fornisce un meccanismo per generare ed esportare queste chiavi iniziali. Una volta generate, queste chiavi possono essere esportate utilizzando i metodi wrap TR-31, TR-34 e RSA. Le chiavi IPEK non sono persistenti e non possono essere utilizzate per operazioni successive sulla crittografia dei pagamenti AWS

AWS La crittografia dei pagamenti non impone la suddivisione tra le prime due parti del KSN. Se desideri memorizzare l'identificatore di derivazione insieme al BDK, puoi utilizzare la funzione dei tag a questo scopo. AWS

Note

La parte del contatore del KSN (32 bit per AES DUKPT) non viene utilizzata per la derivazione IPEK/IK. Pertanto, un input di 12345678901234560001 e 12345678901234569999 produrrà lo stesso IPEK.

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

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

Esportazione di chiavi asimmetriche (RSA)

Chiamata `get-public-key-certificate` per esportare una chiave pubblica sotto forma di certificato. Questa API esporterà il certificato e il relativo certificato radice codificato in formato base64.

NOTA: questa API non è idempotente: le chiamate successive possono generare certificati diversi anche se la chiave sottostante è la stessa.

Example

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

```
{
```

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

Utilizzo di alias

Un alias è un nome descrittivo per una AWS chiave di crittografia dei pagamenti. Ad esempio, un alias consente di fare riferimento a una chiave come `alias/test-key` anziché.

```
arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaiflw2h
```

È possibile utilizzare un alias per identificare una chiave nella maggior parte delle operazioni di gestione delle chiavi (piano di controllo) e nelle operazioni [crittografiche \(piano dati\)](#).

Puoi anche consentire e negare l'accesso alla chiave AWS Payment Cryptography in base ai relativi alias senza modificare le politiche o gestire le concessioni. Questa funzionalità fa parte del supporto del servizio per il controllo degli accessi basato sugli [attributi \(ABAC\)](#).

Gran parte della potenza degli alias deriva dalla possibilità di modificare la chiave associata a un alias in qualsiasi momento. Gli alias possono rendere il tuo codice più facile da scrivere e gestire. Ad esempio, supponiamo di utilizzare un alias per fare riferimento a una particolare chiave di crittografia dei AWS pagamenti e di voler modificare la chiave di crittografia dei pagamenti. AWS In tal caso, basta associare l'alias a una chiave diversa. Non è necessario modificare il codice o la configurazione dell'applicazione.

Gli alias semplificano anche il riutilizzo dello stesso codice in diverse Regioni AWS. Crea alias con lo stesso nome in più regioni e associa ogni alias a una chiave di crittografia dei AWS pagamenti nella relativa regione. Quando il codice viene eseguito in ciascuna regione, l'alias fa riferimento alla chiave di crittografia dei AWS pagamenti associata in quella regione.

Puoi creare un alias per una chiave AWS di crittografia dei pagamenti utilizzando l'API.

CreateAlias

L'API AWS Payment Cryptography fornisce il controllo completo degli alias in ogni account e regione. L'API include operazioni per creare un alias (`CreateAlias`), visualizzare i nomi degli alias e il `keyArn` collegato (`list-aliases`), modificare la chiave AWS Payment Cryptography associata a un alias (`update-alias`) ed eliminare un alias (`delete-alias`).

Argomenti

- [Informazioni sugli alias](#)

- [Utilizzo di alias nelle applicazioni](#)
- [API correlate](#)

Informazioni sugli alias

Scopri come funzionano gli alias nella crittografia dei pagamenti. AWS

Un alias è una risorsa indipendente AWS

Un alias non è una proprietà di una chiave di crittografia dei AWS pagamenti. Le azioni che esegui sull'alias non influiscono sulla chiave associata. Puoi creare un alias per una chiave di crittografia dei AWS pagamenti e quindi aggiornare l'alias in modo che sia associato a una chiave di crittografia dei pagamenti diversa AWS . Puoi anche eliminare l'alias senza alcun effetto sulla chiave di crittografia dei pagamenti associata. AWS Se elimini una chiave AWS di crittografia dei pagamenti, tutti gli alias associati a tale chiave non verranno assegnati.

Se specifichi un alias come risorsa in una policy IAM, la policy si riferisce all'alias, non alla chiave Payment Cryptography associata. AWS

Ogni alias ha un nome descrittivo

Quando si crea un alias, si specifica il nome dell'alias preceduto da `alias/` Ad esempio `alias/test_1234`

Ogni alias è associato a una chiave AWS di crittografia dei pagamenti alla volta

L'alias e la relativa chiave AWS di crittografia dei pagamenti devono trovarsi nello stesso account e nella stessa regione.

Una chiave AWS di crittografia dei pagamenti può essere associata a più di un alias contemporaneamente, ma ogni alias può essere mappato su una sola chiave

Ad esempio, questo `list-aliases` output mostra che l'`alias/sampleAlias1alias` è associato esattamente a una chiave di crittografia dei AWS pagamenti di destinazione, rappresentata dalla proprietà `KeyArn`

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

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

È possibile associare più alias alla stessa chiave di crittografia dei pagamenti AWS

Ad esempio, puoi associare gli alias/sampleAlias2 alias alias/sampleAlias1; and alla stessa chiave.

```
$ 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 deve essere univoco per un determinato account e regione

Ad esempio, è possibile avere un solo alias alias/sampleAlias1 in ogni account e regione. Gli alias fanno distinzione tra maiuscole e minuscole, ma si consiglia di non utilizzare alias che differiscono solo nelle lettere maiuscole, in quanto possono essere soggetti a errori. Non è possibile modificare un nome alias. Tuttavia, puoi eliminare l'alias e creare un nuovo alias con il nome desiderato.

Puoi creare un alias con lo stesso nome in diverse regioni

Ad esempio, è possibile avere alias `alias/sampleAlias2` negli Stati Uniti orientali (Virginia settentrionale) e `alias` negli Stati Uniti occidentali (Oregon). `alias/sampleAlias2` Ogni alias verrebbe associato a una chiave di crittografia dei AWS pagamenti nella relativa regione. Se il tuo codice fa riferimento a un nome alias come `alias/finance-key`, puoi eseguirlo in più regioni. In ogni regione, utilizza un `alias/SampleAlias2` diverso. Per informazioni dettagliate, vedi [Utilizzo di alias nelle applicazioni](#).

È possibile modificare la chiave di crittografia dei pagamenti associata a un AWS alias

È possibile utilizzare l'`UpdateAlias` operazione per associare un alias a una chiave di crittografia dei AWS pagamenti diversa. Ad esempio, se l'`alias/sampleAlias2` alias è associato alla chiave `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qai f1lw2h` AWS Payment Cryptography, è possibile aggiornarlo in modo che sia associato alla chiave `arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi`

Warning

AWS La crittografia dei pagamenti non verifica che la vecchia e la nuova chiave abbiano tutti gli stessi attributi, come l'utilizzo delle chiavi. L'aggiornamento con un tipo di chiave diverso può causare problemi nell'applicazione.

Alcune chiavi non hanno alias

Un alias è una funzionalità opzionale e non tutte le chiavi avranno alias a meno che non si scelga di utilizzare l'ambiente in questo modo. Le chiavi possono essere associate agli alias utilizzando il comando `create-alias`. Inoltre, è possibile utilizzare l'operazione `update-alias` per modificare la chiave AWS Payment Cryptography associata a un alias e l'operazione `delete-alias` per eliminare un alias. Di conseguenza, alcune chiavi di AWS Payment Cryptography potrebbero avere diversi alias e altre potrebbero non averne nessuno.

Mappatura di una chiave su un alias

È possibile mappare una chiave (rappresentata da un ARN) a uno o più alias utilizzando il comando `create-alias`. Questo comando non è idempotente: per aggiornare un alias, usa il comando `update-alias`.

```
$ aws payment-cryptography create-alias --alias-name alias/sampleAlias1 \
```

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

```
{  
  "Alias": {  
    "AliasName": "alias/alias/sampleAlias1",  
    "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
kwapwa6qaiif1lw2h"  
  }  
}
```

Utilizzo di alias nelle applicazioni

È possibile utilizzare un alias per rappresentare una chiave di crittografia dei pagamenti nel codice dell'applicazione. AWS Il `key-identifier` parametro nelle [operazioni sui dati AWS di crittografia dei pagamenti e in altre operazioni](#) come List Keys accetta un nome alias o 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}
```

Quando usi un alias ARN, ricorda che la mappatura degli alias su AWS una chiave di crittografia dei pagamenti è definita nell'account che possiede la chiave di crittografia AWS dei pagamenti e potrebbe differire in ogni regione.

Uno degli usi più efficaci degli alias è nelle applicazioni eseguite in più Regioni AWS.

È possibile creare una versione diversa dell'applicazione in ciascuna regione o utilizzare un dizionario, una configurazione o un'istruzione switch per selezionare la chiave di crittografia dei AWS pagamenti corretta per ciascuna regione. Ma potrebbe essere più semplice creare un alias con lo stesso nome alias in ogni regione. Tieni presente che il nome alias rispetta la distinzione tra maiuscole e minuscole.

API correlate

[Tag](#)

I tag sono coppie di chiavi e valori che fungono da metadati per organizzare le chiavi di crittografia dei AWS pagamenti. Possono essere utilizzati per identificare in modo flessibile le chiavi o raggruppare una o più chiavi insieme.

Procurati le chiavi

Una chiave AWS di crittografia dei pagamenti rappresenta una singola unità di materiale crittografico e può essere utilizzata solo per operazioni crittografiche per questo servizio. L' `GetKeys` API accetta un `KeyIdentifier` input e restituisce gli attributi immutabili e mutabili della chiave, ma non contiene materiale crittografico.

Example

```
$ 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"
  }
}
```

Ottieni la chiave/certificato pubblico associato a una coppia di chiavi

Get Public Key/Certificate restituisce la chiave pubblica indicata da KeyArn. Può trattarsi della parte di chiave pubblica di una coppia di chiavi generata su AWS Payment Cryptography o di una chiave pubblica importata in precedenza. Il caso d'uso più comune consiste nel fornire la chiave pubblica a un servizio esterno che crittograferà i dati. Tali dati possono quindi essere trasferiti a un'applicazione che sfrutta la crittografia dei pagamenti AWS e i dati possono essere decrittografati utilizzando la chiave privata protetta all'interno di Payment Cryptography.

Il servizio restituisce le chiavi pubbliche come certificato pubblico. Il risultato dell'API contiene la CA e il certificato a chiave pubblica. Entrambi gli elementi di dati sono codificati in base 64.

Note

Il certificato pubblico restituito è destinato a essere di breve durata e non è destinato a essere idempotente. È possibile ricevere un certificato diverso per ogni chiamata API, anche se la chiave pubblica stessa rimane invariata.

Example

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

```
{
  "KeyCertificate":
  "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUV2VENDQXFXZ0F3SUJBZ01SQUo10Wd2VkpDd3d1Y1dMN1dYZEpYY
  "KeyCertificateChain":
  "LS0tLS1CRUdJTiBDRVJUSUZJQ0FURSB0tLS0tCk1JSUY0VENDQThZ0F3SUJBZ01SQUt1N2piaHFKZjJPd3FGUWI5c3VuO
}
```

Chiavi di tagging

In AWS Payment Cryptography, puoi aggiungere tag a una chiave di crittografia dei AWS pagamenti quando [crei una chiave](#) e taggare o rimuovere tag alle chiavi esistenti a meno che non siano in attesa di eliminazione. I tag sono opzionali, ma possono essere molto utili.

[Per informazioni generali sui tag, incluse le migliori pratiche, le strategie di etichettatura e il formato e la sintassi dei tag, consulta Tagging resources in. AWS](#)[Riferimenti generali di Amazon Web Services](#)

Argomenti

- [Informazioni sui tag nella crittografia dei pagamenti AWS](#)
- [Visualizzazione dei tag chiave nella console](#)
- [Gestione dei tag chiave con operazioni API](#)
- [Controllo degli accessi ai tag](#)
- [Utilizzo dei tag per controllare l'accesso alle chiavi](#)

Informazioni sui tag nella crittografia dei pagamenti AWS

Un tag è un'etichetta di metadati opzionale che puoi assegnare (o AWS assegnare) a una risorsa. AWS Ogni tag è costituito da una chiave di tag e da un valore di tag, entrambe le stringhe fanno distinzione tra maiuscole e minuscole. Il valore di tag può essere una stringa vuota (null). Ogni tag su una risorsa deve avere una chiave di tag diversa, ma puoi aggiungere lo stesso tag a più risorse. AWS Ogni risorsa può avere fino a 50 tag creati dall'utente.

Non includere informazioni riservate o sensibili nella chiave o nel valore del tag. I tag sono accessibili a molti Servizi AWS, inclusa la fatturazione.

In AWS Payment Cryptography, puoi aggiungere tag a una chiave al momento [della creazione della chiave](#) e taggare o rimuovere tag alle chiavi esistenti a meno che non siano in attesa di eliminazione. Non è possibile etichettare gli alias. I tag sono opzionali, ma possono essere molto utili.

Ad esempio, puoi aggiungere un "Project"="Alpha" tag a tutte le chiavi di crittografia dei AWS pagamenti e ai bucket Amazon S3 che usi per il progetto Alpha. Un altro esempio consiste nell'aggiungere un "BIN"="20130622" tag a tutte le chiavi associate a uno specifico numero di identificazione bancaria (BIN).

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

Per informazioni generali sui tag, inclusi il formato e la sintassi, vedere [Tagging AWS resources](#) in. Riferimenti generali di Amazon Web Services

I tag consentono di:

- Identifica e organizza le tue risorse. AWS Molti AWS servizi supportano l'etichettatura, quindi puoi assegnare lo stesso tag a risorse di servizi diversi per indicare che le risorse sono correlate. Ad esempio, puoi assegnare lo stesso tag a una chiave di crittografia dei AWS pagamenti e a un volume o segreto Amazon Elastic Block Store (Amazon EBS). AWS Secrets Manager Puoi anche utilizzare i tag per identificare le chiavi per l'automazione.
- Tieni traccia AWS dei costi. Quando aggiungi tag alle tue AWS risorse, AWS genera un rapporto sull'allocazione dei costi con utilizzo e costi aggregati per tag. È possibile utilizzare questa funzionalità per tenere traccia dei costi della crittografia dei AWS pagamenti per un progetto, un'applicazione o un centro di costo.

Per ulteriori informazioni sull'utilizzo dei tag per l'allocazione dei costi, consulta [Uso dei tag per l'allocazione dei costi](#) nella Guida per l'utente di AWS Billing . Per informazioni sulle regole che si applicano alle chiavi dei tag e ai valori dei tag, consulta [Limitazioni per i tag definiti dall'utente](#) nella Guida per l'utente di AWS Billing .

- Controlla l'accesso alle tue AWS risorse. Consentire e negare l'accesso alle chiavi in base ai relativi tag fa parte del supporto di AWS Payment Cryptography per il controllo degli accessi basato sugli attributi (ABAC). Per informazioni sul controllo dell'accesso alla crittografia dei pagamenti AWS in base ai relativi tag, consulta [Autorizzazione basata sui tag di crittografia dei pagamenti AWS](#) Per informazioni più generali sull'uso dei tag per controllare l'accesso alle AWS risorse, consulta [Controlling Access to AWS Resources Using Resource Tags](#) nella IAM User Guide.

AWS Payment Cryptography scrive una voce nel AWS CloudTrail registro quando si utilizzano le ListTagsForResource operazioni TagResource UntagResource, o.

Visualizzazione dei tag chiave nella console

Per visualizzare i tag nella console, è necessario il permesso di etichettare la chiave in base a una policy IAM che includa la chiave. Sono necessarie queste autorizzazioni oltre alle autorizzazioni per visualizzare le chiavi nella console.

Gestione dei tag chiave con operazioni API

Puoi utilizzare l'[API AWS Payment Cryptography](#) per aggiungere, eliminare ed elencare i tag per le chiavi che gestisci. Questi esempi utilizzano la [AWS Command Line Interface \(AWS CLI\)](#), ma puoi usare anche qualsiasi linguaggio di programmazione supportato. Non puoi taggare Chiavi gestite da AWS.

Per aggiungere, modificare, visualizzare ed eliminare i tag per una chiave, è necessario disporre delle autorizzazioni necessarie. Per informazioni dettagliate, vedi [Controllo degli accessi ai tag](#) .

Argomenti

- [CreateKey: aggiungi tag a una nuova chiave](#)
- [TagResource: aggiungi o modifica i tag per una chiave](#)
- [ListResourceTag: ottieni i tag per una chiave](#)
- [UntagResource: elimina i tag da una chiave](#)

CreateKey: aggiungi tag a una nuova chiave

Puoi aggiungere tag quando crei una chiave. Per specificare i tag, utilizzate il `Tags` parametro dell'[CreateKey](#) operazione.

Per aggiungere tag durante la creazione di una chiave, il chiamante deve disporre dell'`payment-cryptography:TagResource` autorizzazione in una policy IAM. Come minimo, l'autorizzazione deve coprire tutte le chiavi dell'account e della regione. Per informazioni dettagliate, vedi [Controllo degli accessi ai tag](#).

Il valore del parametro `Tags` di `CreateKey` è una raccolta di coppie di chiave di tag e valore di tag per cui si applica la distinzione tra maiuscole e minuscole. Ogni tag su una chiave deve avere un nome di tag diverso. Il valore di tag può essere una stringa nulla o vuota.

Ad esempio, il AWS CLI comando seguente crea una chiave di crittografia simmetrica con un `Project:Alpha` tag. Quando si specificano più coppie chiave-valore, utilizzare uno spazio per separare ciascuna coppia.

```
$ aws payment-cryptography create-key --exportable --key-attributes
KeyAlgorithm=TDDES_2KEY, \
    KeyUsage=TR31_C0_CARD_VERIFICATION_KEY,KeyClass=SYMMETRIC_KEY, \
    KeyModesOfUse='{Generate=true,Verify=true}' \
    --tags '[{"Key":"Project","Value":"Alpha"}, {"Key":"BIN","Value":"123456"}]'
```

Quando questo comando ha esito positivo, restituisce un `Key` oggetto con informazioni sulla nuova chiave. Tuttavia, `Key` non include tag. Per ottenere i tag, utilizzate l'operazione [ListResourceTags](#).

TagResource: aggiungi o modifica i tag per una chiave

L'[TagResource](#) operazione aggiunge uno o più tag a una chiave. Non puoi utilizzare questa operazione per aggiungere o modificare tag in un Account AWS diverso.

Per aggiungere un tag, specifica una nuova chiave di tag e un valore di tag. Per modificare un tag, specifica una chiave di tag esistente e un nuovo valore di tag. Ogni tag su una chiave deve avere una chiave di tag diversa. Il valore di tag può essere una stringa nulla o vuota.

Ad esempio, il comando seguente aggiunge **UseCase BIN** tag a una chiave di esempio.

```
$ 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"}]'
```

Quando questo comando ha esito positivo, non restituisce alcun output. Per visualizzare i tag su una chiave, utilizzate l'operazione [ListResourceTag](#).

Puoi inoltre usare `TagResource` per modificare il valore di un tag esistente. Per sostituire un valore di tag, specifica la stessa chiave di tag con un valore diverso. I tag non elencati in un comando di modifica non vengono modificati o rimossi.

Ad esempio, questo comando modifica il valore del tag `Project` da `Alpha` a `Noe`.

Il comando restituirà `http/200` senza contenuto. Per vedere le modifiche, usa `ListTagsForResource`

```
$ aws payment-cryptography tag-resource --resource-arn arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaif1lw2h \  
    --tags '[{"Key":"Project","Value":"Noe"}]'
```

ListResourceTag: ottieni i tag per una chiave

L'operazione [ListResourceTags](#) ottiene i tag per una chiave. Il parametro `ResourceArn` (`keyArn` o `keyAlias`) è obbligatorio. Non è possibile utilizzare questa operazione per visualizzare i tag sulle chiavi in un modo diverso. Account AWS

Ad esempio, il comando seguente ottiene i tag per una chiave di esempio.

```
$ 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: elimina i tag da una chiave

L'[UntagResource](#) operazione elimina i tag da una chiave. Per identificare i tag da eliminare, specifica le chiavi dei tag. Non è possibile utilizzare questa operazione per eliminare tag da chiavi diverse Account AWS.

Quando l'operazione `UntagResource` ha esito positivo non restituisce alcun output. Inoltre, se la chiave del tag specificata non viene trovata sulla chiave, non genera un'eccezione né restituisce una risposta. Per confermare che l'operazione ha funzionato, utilizzate l'operazione [ListResourceTags](#).

Ad esempio, questo comando elimina il **Purpose** tag e il relativo valore dalla chiave specificata.

```
$ aws payment-cryptography untag-resource \
    --resource-arn arn:aws:payment-cryptography:us-east-2:111122223333:key/
    kwapwa6qaif1lw2h --tag-keys Project
```

Controllo degli accessi ai tag

Per aggiungere, visualizzare ed eliminare i tag utilizzando l'API, i responsabili necessitano delle autorizzazioni di etichettatura nelle politiche IAM.

Puoi anche limitare queste autorizzazioni utilizzando chiavi di condizione AWS globali per i tag. In AWS Payment Cryptography, queste condizioni possono controllare l'accesso alle operazioni di tagging, come e. [TagResourceUntagResource](#)

Per esempi di policy e ulteriori informazioni, consulta [Controllo dell'accesso in base alle chiavi di tag](#) nella Guida per l'utente di IAM.

Le autorizzazioni per creare e gestire i tag funzionano come descritto di seguito.

crittografia dei pagamenti: `TagResource`

Consente ai principali di aggiungere o modificare tag. Per aggiungere tag durante la creazione di una chiave, il principale deve disporre dell'autorizzazione in una policy IAM che non è limitata a chiavi particolari.

crittografia dei pagamenti: `ListTagsForResource`

Consente ai presidi di visualizzare i tag sulle chiavi.

crittografia dei pagamenti: `UntagResource`

Consente ai principali di eliminare i tag dalle chiavi.

Autorizzazioni ad assegnare tag nelle policy

Puoi fornire l'autorizzazione ad assegnare tag in una policy delle chiavi o in una policy IAM. Ad esempio, la politica chiave di esempio seguente fornisce a determinati utenti l'autorizzazione a contrassegnare la chiave. Fornisce a tutti gli utenti che possono assumere l'esempio dei ruoli di Amministratore o Sviluppatore il permesso di visualizzare i tag.

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

```
}
```

Per concedere ai principali il permesso di etichettare più chiavi, puoi utilizzare una policy IAM. Affinché questa policy sia efficace, la policy chiave per ogni chiave deve consentire all'account di utilizzare le policy IAM per controllare l'accesso alla chiave.

Ad esempio, la seguente policy IAM consente ai principali di creare chiavi. Consente inoltre loro di creare e gestire tag su tutte le chiavi dell'account specificato. Questa combinazione consente ai responsabili di utilizzare il parametro tags dell'[CreateKey](#) operazione per aggiungere tag a una chiave durante la creazione.

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

Limitazione delle autorizzazioni ad assegnare tag

È possibile limitare le autorizzazioni di assegnazione dei tag utilizzando Condizioni della policy. Le seguenti condizioni della policy possono essere applicate alle autorizzazioni `payment-cryptography:TagResource` e `payment-cryptography:UntagResource`. Ad esempio, è possibile utilizzare la condizione `aws:RequestTag/tag-key` per consentire a un principale di aggiungere solo tag specifici o impedire a un principale di aggiungere tag con chiavi tag particolari.

- [leggi: RequestTag](#)
- [aws:ResourceTag/tag-key \(solo politiche IAM\)](#)
- [aws: TagKeys](#)

Come best practice quando usi i tag per controllare l'accesso alle chiavi, usa il tasto `aws:RequestTag/tag-key` o `aws:TagKeys` condition per determinare quali tag (o chiavi di tag) sono consentiti.

Ad esempio, la seguente istruzione della policy IAM è simile a quella precedente. Tuttavia, questa policy consente ai principali di creare tag (`TagResource`) ed eliminare i tag `UntagResource` solo per i tag con chiave di tag `Project`.

Poiché `TagResource` le `UntagResource` richieste possono includere più tag, è necessario specificare un operatore `ForAllValues` o `ForAnyValue` impostare con la `TagKeys` condizione [aws:](#). L'operatore `ForAnyValue` richiede che almeno una delle chiavi di tag nella richiesta corrisponda a una delle chiavi di tag nella policy. L'operatore `ForAllValues` richiede che tutte le chiavi di tag nella richiesta corrispondano a una delle chiavi di tag nella policy. L'`ForAllValues` operatore restituisce anche `true` se non ci sono tag nella richiesta, ma `TagResource` `UntagResource` fallisce quando non viene specificato alcun tag. Per dettagli sugli operatori del set, consulta [Utilizzare più chiavi e valori](#) nella Guida per l'utente di 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"}
  }
}
]
```

Utilizzo dei tag per controllare l'accesso alle chiavi

Puoi controllare l'accesso alla crittografia dei AWS pagamenti in base ai tag sulla chiave. Ad esempio, puoi scrivere una policy IAM che consenta ai responsabili di abilitare e disabilitare solo le chiavi con un tag particolare. Oppure puoi utilizzare una policy IAM per impedire ai principali di utilizzare le chiavi nelle operazioni crittografiche a meno che la chiave non abbia un tag particolare.

Questa funzionalità fa parte del supporto di AWS Payment Cryptography per il controllo degli accessi basato sugli attributi (ABAC). [Per informazioni sull'utilizzo dei tag per controllare l'accesso alle AWS risorse, consulta A cosa serve ABAC? AWS](#) e [Controllo dell'accesso alle AWS risorse utilizzando i tag delle risorse](#) nella Guida per l'utente IAM.

Note

AWS Payment Cryptography supporta la chiave di contesto [aws:ResourceTag/tag-key](#) global condition, che consente di controllare l'accesso alle chiavi in base ai tag sulla chiave. Poiché più chiavi possono avere lo stesso tag, questa funzionalità consente di applicare l'autorizzazione a un set selezionato di chiavi. Puoi anche cambiare facilmente le chiavi del set cambiando i relativi tag.

In AWS Payment Cryptography, la chiave di `aws:ResourceTag/tag-key` condizione è supportata solo nelle policy IAM. Non è supportata nelle politiche chiave, che si applicano solo a una chiave, o nelle operazioni che non utilizzano una chiave particolare, come le [ListAliases](#) operazioni [ListKeyso](#).

Il controllo dell'accesso con i tag offre un modo semplice, scalabile e flessibile per gestire le autorizzazioni. Tuttavia, se non è progettato e gestito correttamente, può consentire o negare l'accesso alle chiavi inavvertitamente. Se utilizzi tag per controllare l'accesso, prendi in considerazione le seguenti procedure.

- Utilizza i tag per rafforzare le best practice di [Accesso meno privilegiato](#). Concedi ai responsabili IAM solo le autorizzazioni di cui hanno bisogno solo per le chiavi che devono utilizzare o gestire. Ad esempio, usa i tag per etichettare le chiavi utilizzate per un progetto. Quindi concedi al team di progetto il permesso di utilizzare solo le chiavi con il tag del progetto.
- Fai attenzione a dare ai principali le autorizzazioni `payment-cryptography:TagResource` e `payment-cryptography:UntagResource` che consentono di aggiungere, modificare ed eliminare tag. Quando usi i tag per controllare l'accesso alle chiavi, la modifica di un tag può dare ai responsabili il permesso di usare chiavi che altrimenti non avrebbero il permesso di usare. Può anche negare l'accesso alle chiavi di cui altri dirigenti hanno bisogno per svolgere il proprio lavoro. Gli amministratori chiave che non dispongono dell'autorizzazione per modificare le politiche chiave o creare sovvenzioni possono controllare l'accesso alle chiavi se dispongono dell'autorizzazione per gestire i tag.

Quando possibile, utilizza una condizione politica, ad esempio `aws:RequestTag/tag-key` o `aws:TagKeys` per [limitare le autorizzazioni di etichettatura del principale](#) a tag o modelli di tag particolari su chiavi particolari.

- Rivedi i principi del tuo sistema Account AWS che attualmente dispongono delle autorizzazioni di etichettatura e rimozione dei tag e modificali, se necessario. Le policy IAM potrebbero consentire le autorizzazioni di etichettatura e rimozione dei tag su tutte le chiavi. Ad esempio, la policy gestita dall'amministratore consente ai responsabili di etichettare, rimuovere tag ed elencare i tag su tutte le chiavi.
- Prima di impostare una politica che dipenda da un tag, esamina i tag sulle chiavi del tuo Account AWS. Assicurati che la tua policy si applichi solo ai tag che intendi includere. Usa [CloudTrail i registri e gli](#) [CloudWatch allarmi](#) per avvisarti delle modifiche ai tag che potrebbero influire sull'accesso alle tue chiavi.
- Le condizioni delle policy basate su tag utilizzano la corrispondenza dei modelli; non sono legate a una particolare istanza di un tag. Una policy che utilizza chiavi di condizione basate su tag influisce su tutti i tag nuovi ed esistenti che corrispondono al modello. Se si elimina e si ricrea un tag che corrisponde a una condizione della policy, la condizione si applica al nuovo tag, proprio come quello precedente.

Ad esempio, considerare il seguente esempio di policy IAM. Consente ai responsabili di richiamare le operazioni [Decrypt](#) solo sulle chiavi del tuo account che si trovano nella regione Stati Uniti orientali (Virginia settentrionale) e dispongono di un tag. `"Project"="Alpha"` È possibile collegare questa policy ai ruoli nel progetto Alpha di esempio.

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

L'esempio seguente di politica IAM consente ai responsabili di utilizzare qualsiasi chiave dell'account per determinate operazioni crittografiche. Ma proibisce ai principali di utilizzare queste operazioni crittografiche su chiavi con un "Type"="Reserved" tag o senza tag. "Type"

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Sid": "IAMAllowCryptographicOperations",
      "Effect": "Allow",
      "Action": [
        "payment-cryptography:EncryptData",
        "payment-cryptography:DecryptData",
        "payment-cryptography:ReEncrypt*"
      ],
      "Resource": "arn:aws:payment-cryptography:*:111122223333:key/*"
    },
    {
      "Sid": "IAMDenyOnTag",
      "Effect": "Deny",
      "Action": [
        "payment-cryptography:EncryptData",
        "payment-cryptography:DecryptData",

```

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

Comprensione degli attributi chiave della chiave Payment Cryptography AWS

Un principio di una corretta gestione delle chiavi è che le chiavi abbiano un ambito appropriato e possano essere utilizzate solo per operazioni consentite. Pertanto, alcune chiavi possono essere create solo con determinate modalità di utilizzo. Ove possibile, ciò si allinea alle modalità d'uso disponibili definite da [TR-31](#).

Sebbene AWS Payment Cryptography ti impedisca di creare chiavi non valide, qui vengono fornite combinazioni valide per tua comodità.

Chiavi simmetriche

- TR31_B0_BASE_DERIVATION_KEY

- Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo chiave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_C0_CARD_VERIFICATION_KEY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo principali: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions
- TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo principali: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_E0_EMV_MKEY_APP_CRYPTOGAMS
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo chiave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_E1_EMV_MKEY_CONFIDENTIALITY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo chiave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_E2_EMV_MKEY_INTEGRITY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo chiave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_E4_EMV_MKEY_DYNAMIC_NUMBERS
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo chiave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_E5_EMV_MKEY_CARD_PERSONALIZATION
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo chiave: {= DeriveKey true}, {= true} NoRestrictions

- TR31_E6_EMV_MKEY_OTHER
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo chiave: {= DeriveKey true}, {= true} NoRestrictions
- TR31_K0_KEY_ENCRYPTION_KEY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo principali: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_K1_KEY_BLOCK_PROTECTION_KEY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo principali: {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
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY
 - Combinazione consentita delle principali modalità di utilizzo: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions
- TR31_M3_ISO_9797_3_MAC_KEY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY
 - Combinazione consentita delle principali modalità di utilizzo: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions
- TR31_M6_ISO_9797_5_CMAC_KEY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo principali: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions
- TR31_M7_HMAC_KEY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo principali: {Generate = true}, {Verify = true}, {Generate = true, Verify= true}, {= true} NoRestrictions
- TR31_P0_PIN_ENCRYPTION_KEY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256

- Combinazione consentita di modalità di utilizzo principali: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}, {= true} NoRestrictions
- TR31_V1_IBM3624_PIN_VERIFICATION_KEY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo principali: {Generate = true}, {Verify = true}, {Generate = true, Verify = true}, {= true} NoRestrictions
- TR31_V2_VISA_PIN_VERIFICATION_KEY
 - Algoritmi chiave consentiti: TDES_2KEY, TDES_3KEY, AES_128, AES_192, AES_256
 - Combinazione consentita di modalità di utilizzo principali: {Generate = true}, {Verify = true}, {Generate = true, Verify = true}, {= true} NoRestrictions

Chiavi asimmetriche

- TR31_D1_KEY_ASYMMETRIC_FOR_DATA_ENCRYPTION
 - Algoritmi chiave consentiti: RSA_2048, RSA_3072, RSA_4096
 - Combinazione consentita di modalità chiave di utilizzo: {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}, {Encrypt = true, Wrap = true}, {Decrypt = true, Unwrap = true}
 - NOTA: {Encrypt = true, Wrap = true} è l'unica opzione valida per importare una chiave pubblica destinata alla crittografia dei dati o al confezionamento di una chiave
- TR31_S0_ASYMMETRIC_KEY_FOR_DIGITAL_SIGNATURE
 - Algoritmi chiave consentiti: RSA_2048, RSA_3072, RSA_4096
 - Combinazione consentita delle principali modalità di utilizzo: {Sign = true}, {Verify = true}
 - NOTA: {Verify = true} è l'unica opzione valida quando si importa una chiave destinata alla firma, come il certificato radice, il certificato intermedio o i certificati di firma per TR-34.

Operazioni sui dati

Dopo aver stabilito una chiave AWS di crittografia dei pagamenti, questa può essere utilizzata per eseguire operazioni crittografiche. Diverse operazioni eseguono diversi tipi di attività, dalla crittografia all'hashing fino agli algoritmi specifici del dominio come la generazione CVV2.

I dati crittografati non possono essere decrittografati senza la chiave di decrittografia corrispondente (la chiave simmetrica o la chiave privata a seconda del tipo di crittografia). Analogamente, gli algoritmi di hashing e quelli specifici del dominio non possono essere verificati senza la chiave simmetrica o la chiave pubblica.

[Per informazioni sui tipi di chiave validi per operazioni specifiche, consulta Chiavi valide per operazioni crittografiche](#)

Note

Si consiglia di utilizzare i dati di test in un ambiente non di produzione. L'utilizzo di chiavi e dati di produzione (PAN, BDK ID, ecc.) in un ambiente non di produzione può influire sull'ambito di conformità, ad esempio per PCI DSS e PCI P2PE.

Argomenti

- [Crittografa, decrittografa e ricrittografa i dati](#)
- [Genera e verifica i dati della carta](#)
- [Generazione, traduzione e verifica dei dati PIN](#)
- [Crittogramma Verify Auth Request \(ARQC\)](#)
- [Genera e verifica MAC](#)
- [Chiavi valide per operazioni crittografiche](#)

Crittografa, decrittografa e ricrittografa i dati

I metodi di crittografia e decrittografia possono essere utilizzati per crittografare o decrittografare i dati utilizzando una varietà di tecniche simmetriche e asimmetriche tra cui TDES, AES e RSA. [Questi metodi supportano anche chiavi derivate utilizzando le tecniche DUKPT ed EMV.](#) Nei casi d'uso in cui si desidera proteggere i dati con una nuova chiave senza esporre i dati sottostanti, è possibile utilizzare anche il ReEncrypt comando.

Note

Quando si utilizzano le funzioni di crittografia/decrittografia, si presume che tutti gli input siano in HexBinary, ad esempio un valore 1 verrà immesso come 31 (esadecimale) e una t minuscola viene rappresentata come 74 (esadecimale). Tutti gli output sono anche in HexBinary.

[Per i dettagli su tutte le opzioni disponibili, consulta la Guida API per Encrypt, Decrypte Re-Encrypt.](#)

Argomenti

- [Crittografare i dati](#)
- [Decrittare i dati](#)

Crittografare i dati

[L'Encrypt DataAPI viene utilizzata per crittografare i dati utilizzando chiavi di crittografia dei dati simmetriche e asimmetriche, nonché chiavi derivate da DUKPT ed EMV.](#) Sono supportati vari algoritmi e varianti, tra cui, e. TDES RSA AES

Gli input principali sono la chiave di crittografia utilizzata per crittografare i dati, i dati in testo semplice in formato HexBinary da crittografare e gli attributi di crittografia come il vettore di inizializzazione e la modalità per i cifrari a blocchi come TDES. I dati in chiaro devono essere espressi in multipli di 8 byte per, 16 byte per TDES e della lunghezza della chiave nel caso di. AES RSA Gli input chiave simmetrici (TDES, AES, DUKPT, EMV) devono essere aggiunti nei casi in cui i dati di input non soddisfino questi requisiti. La tabella seguente mostra la lunghezza massima del testo in chiaro per ogni tipo di chiave e il tipo di padding definito per le chiavi RSA. EncryptionAttributes

Tipo di imbottitura	RSA_2048	RSA_3072	RSA_4096
OAEP SHA1	428	684	940
OAEP SHA256	380	636	892
OAEP SHA512	252	508	764
PKCS1	488	744	1000

Tipo di imbottitura	RSA_2048	RSA_3072	RSA_4096
None	488	744	1000

Gli output primari includono i dati crittografati come testo cifrato in formato HexBinary e il valore di checksum per la chiave di crittografia. [Per i dettagli su tutte le opzioni disponibili, consulta la Guida API per Encrypt.](#)

Esempi

- [Crittografa i dati utilizzando la chiave simmetrica AES](#)
- [Crittografa i dati utilizzando la chiave DUKPT](#)
- [Crittografa i dati utilizzando una chiave simmetrica derivata da EMV](#)
- [Crittografa i dati utilizzando una chiave RSA](#)

Crittografa i dati utilizzando la chiave simmetrica AES

Note

Tutti gli esempi presuppongono che la chiave pertinente esista già. Le chiavi possono essere create utilizzando l'[CreateKey](#) operazione o importate utilizzando l'[ImportKey](#) operazione.

Example

In questo esempio, crittograferemo i dati in chiaro utilizzando una chiave simmetrica che è stata creata utilizzando l'[CreateKey](#) Operazione o importata utilizzando l'Operazione. [ImportKey](#) Per questa operazione, la chiave deve essere impostata su e KeyModesOfUse impostata su. Encrypt KeyUsage TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY Per ulteriori opzioni, consulta la sezione [Chiavi per le operazioni crittografiche](#).

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

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

Crittografa i dati utilizzando la chiave DUKPT

Example

[In questo esempio, crittograferemo i dati in chiaro utilizzando una chiave DUKPT.](#) AWS

Supporti per la crittografia dei pagamenti e le chiavi DUKPT. TDES AES Per questa operazione, la chiave deve essere impostata `DeriveKey` e `KeyModesOfUse` `KeyUsage` impostata su.

`TR31_B0_BASE_DERIVATION_KEY` Per ulteriori opzioni, consulta la sezione [Chiavi per le operazioni crittografiche](#).

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

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

Crittografa i dati utilizzando una chiave simmetrica derivata da EMV

Example

In questo esempio, crittograferemo i dati di testo non crittografato utilizzando una chiave simmetrica derivata da EMV che è già stata creata. È possibile utilizzare un comando come questo per

inviare dati a una scheda EMV. Per questa operazione, la chiave deve essere KeyModesOfUse impostata su Derive e KeyUsage impostata su TR31_E1_EMV_MKEY_CONFIDENTIALITY o TR31_E6_EMV_MKEY_OTHER. Per maggiori dettagli, consulta la sezione [Chiavi per le operazioni crittografiche](#).

```
$ 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=15000000000000999,Mode=CBC}'
```

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

Crittografa i dati utilizzando una chiave RSA

Example

In questo esempio, crittograferemo i dati in chiaro utilizzando una [chiave pubblica RSA](#) che è stata importata utilizzando l'operazione. [ImportKey](#) Per questa operazione, la chiave deve essere impostata su e KeyModesOfUse impostata su. Encrypt KeyUsage TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION Per ulteriori opzioni, consulta la sezione [Chiavi per le operazioni crittografiche](#).

Per PKCS #7 o altri schemi di padding non attualmente supportati, richiedi prima di chiamare il servizio e seleziona no padding omettendo l'indicatore di padding '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":
  "12DF6A2F64CC566D124900D68E8AFEAA794CA819876E258564D525001D00AC93047A83FB13 \
E73F06329A100704FA484A15A49F06A7A2E55A241D276491AA91F6D2D8590C60CDE57A642BC64A897F4832A3930
\
0FAEC7981102CA0F7370BFBF757F271EF0BB2516007AB111060A9633D1736A9158042D30C5AE11F8C5473EC70F067
\
72590DEA1638E2B41FAE6FB1662258596072B13F8E2F62F5D9FAF92C12BB70F42F2ECDCF56AADF0E311D4118FE3591
\
FB672998CCE9D00FFFE05D2CD154E3120C5443C8CF9131C7A6A6C05F5723B8F5C07A4003A5A6173E1B425E2B5E42AD
\
7A2966734309387C9938B029AFB20828ACFC6D00CD1539234A4A8D9B94CDD4F23A",
  "KeyArn": "arn:aws:payment-cryptography:us-east-1:529027455495:key/5dza7xqd6soanjtb",
  "KeyCheckValue": "FF9DE9CE"
}
```

Decrittare i dati

[L'Decrypt DataAPI viene utilizzata per decrittografare i dati utilizzando chiavi di crittografia dei dati simmetriche e asimmetriche, nonché chiavi derivate DUKPT ed EMV.](#) Sono supportati vari algoritmi e varianti, tra cui, e. TDES RSA AES

Gli input principali sono la chiave di decrittografia utilizzata per decrittografare i dati, i dati di testo cifrato in formato HexBinary da decrittografare e gli attributi di decrittografia come il vettore di inizializzazione, la modalità come i cifrari a blocchi ecc. Gli output principali includono i dati decrittografati come testo semplice in formato HexBinary e il valore di checksum per la chiave di decrittografia. [Per i dettagli su tutte le opzioni disponibili, consulta la Guida API per Decrypt.](#)

Esempi

- [Decrittografa i dati utilizzando la chiave simmetrica AES](#)
- [Decrittografa i dati utilizzando la chiave DUKPT](#)
- [Decrittografa i dati utilizzando una chiave simmetrica derivata da EMV](#)
- [Decrittografa i dati utilizzando una chiave RSA](#)

Decrittografa i dati utilizzando la chiave simmetrica AES

Example

In questo esempio, decifreremo i dati di testo cifrato utilizzando una chiave simmetrica. Questo esempio mostra una AES chiave ma sono anche supportate. TDES_2KEY TDES_3KEY Per questa operazione, la chiave deve essere KeyModesOfUse impostata Decrypt e KeyUsage impostata su TR31_D0_SYMMETRIC_DATA_ENCRYPTION_KEY. Per ulteriori opzioni, consulta la sezione [Chiavi per le operazioni crittografiche](#).

```
$ 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"
}
```

Decrittografa i dati utilizzando la chiave DUKPT

Note

L'utilizzo dei dati di decrittografia con DUKPT per le transazioni P2PE può restituire all'applicazione i dati PAN della carta di credito e di altri titolari di carta di credito che dovranno essere presi in considerazione per determinare l'ambito PCI DSS.

Example

[In questo esempio, decifreremo i dati di testo cifrato utilizzando una chiave DUKPT che è stata creata utilizzando l'Operazione o importata utilizzando l'Operazione. CreateKeyImportKey](#) Per questa

operazione, la chiave deve essere impostata su e impostata su. `KeyModesOfUse DeriveKey`
`KeyUsage TR31_B0_BASE_DERIVATION_KEY` Per ulteriori opzioni, consulta la sezione [Chiavi per le operazioni crittografiche](#). Quando si utilizza `DUKPT`, per l'`TDES` algoritmo, la lunghezza dei dati del testo cifrato deve essere un multiplo di 16 byte. Per l'`AES` algoritmo, la lunghezza dei dati del testo cifrato deve essere un multiplo di 32 byte.

```
$ aws payment-cryptography-data decrypt-data --key-identifier
arn:aws:payment-cryptography:us-east-2:111122223333:key/tqv5yij6wtxx64pi
--cipher-text 33612AB9D6929C3A828EB6030082B2BD --decryption-attributes
'Dukpt={KeySerialNumber=FFFF9876543210E00001}'
```

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

Decrittografa i dati utilizzando una chiave simmetrica derivata da EMV

Example

In questo esempio, decifreremo i dati di testo cifrato utilizzando una chiave simmetrica derivata da EMV che è stata creata utilizzando l'operazione o importata utilizzando l'operazione.

[CreateKeyImportKey](#) Per questa operazione, la chiave deve essere impostata su e impostata su o. `KeyModesOfUse Derive KeyUsage TR31_E1_EMV_MKEY_CONFIDENTIALITY`
`TR31_E6_EMV_MKEY_OTHER` Per maggiori dettagli, consulta la sezione [Chiavi per le operazioni crittografiche](#).

```
$ 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"
}
```

Decrittografa i dati utilizzando una chiave RSA

Example

In questo esempio, decifreremo i dati di testo cifrato utilizzando una [coppia di chiavi](#) RSA che è stata creata utilizzando l'operazione. [CreateKey](#) Per questa operazione, la chiave deve essere impostata su enable e KeyModesOfUse impostata su. Decrypt KeyUsage TR31_D1_ASYMMETRIC_KEY_FOR_DATA_ENCRYPTION Per ulteriori opzioni, consulta la sezione [Chiavi per le operazioni crittografiche](#).

Per PKCS #7 o altri schemi di padding non attualmente supportati, seleziona no padding omettendo l'indicatore di padding 'Asymmetric= {}' e rimuovi il padding dopo aver chiamato il servizio.

```
$ aws payment-cryptography-data decrypt-data \
  --key-identifier arn:aws:payment-cryptography:us-
east-2:111122223333:key/5dza7xqd6soanjtb --cipher-text
8F4C1CAFE7A5DEF9A40BEDE7F2A264635C... \
  --decryption-attributes 'Asymmetric={PaddingType=0AEP_SHA256}'
```

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

Genera e verifica i dati della carta

Genera e verifica i dati delle carte incorpora i dati derivati dai dati delle carte, ad esempio CVV, CVV2, CVC e DCVV.

Argomenti

- [Genera i dati delle carte](#)
- [Verifica i dati della carta](#)

Genera i dati delle carte

L'Generate Card DataAPI viene utilizzata per generare i dati delle carte utilizzando algoritmi come CVV, CVV2 o Dynamic CVV2. [Per vedere quali chiavi possono essere utilizzate per questo comando, consulta la sezione Chiavi valide per le operazioni crittografiche.](#)

Molti valori crittografici come CVV, CVV2, iCVV, CAVV V8 utilizzano lo stesso algoritmo crittografico ma variano i valori di input. [Ad esempio CardVerification, Value1 ha gli input di, numero di carta e data di scadenza.](#) ServiceCode Sebbene [CardVerificationValue2](#) abbia solo due di questi ingressi, ciò è dovuto al fatto che per CVV2/CVC2 è fissato a 000. ServiceCode Analogamente, per iCVV è fissato a 999. ServiceCode Alcuni algoritmi possono riutilizzare i campi esistenti, come CAVV V8, nel qual caso sarà necessario consultare il manuale del provider per i valori di input corretti.

Note

La data di scadenza deve essere inserita nello stesso formato (ad esempio MMYT o YYMM) affinché la generazione e la convalida producano risultati corretti.

Genera CVV2

Example

In questo esempio, genereremo un CVV2 per un determinato PAN con input e data di scadenza della carta. [PAN](#) [Ciò presuppone che sia stata generata una chiave di verifica della carta.](#)

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

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

```
"KeyCheckValue": "CADD1",
"ValidationData": "801"
}
```

Genera iCVV

Example

In questo esempio, genereremo un [iCVV](#) per un determinato PAN con gli input di [PAN](#), un codice di servizio 999 e la data di scadenza della carta. [Ciò presuppone che sia stata generata una chiave di verifica della carta.](#)

Per tutti i parametri disponibili, consulta [CardVerificationValue1 nella guida](#) di riferimento dell'API.

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

```
{
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
tqv5yij6wtxx64pi",
  "KeyCheckValue": "CADD1",
  "ValidationData": "801"
}
```

Verifica i dati della carta

Verify Card Data viene utilizzato per verificare i dati che sono stati creati utilizzando algoritmi di pagamento che si basano su principi di crittografia come. DISCOVER_DYNAMIC_CARD_VERIFICATION_CODE

I valori di input vengono in genere forniti come parte di una transazione in entrata a un emittente o a un partner della piattaforma di supporto. [Per verificare un crittogramma ARQC \(utilizzato per le schede con chip EMV\), consulta Verify ARQC.](#)

Per ulteriori informazioni, consulta la guida alle API. [VerifyCardValidationData](#)

Se il valore è verificato, l'api restituirà http/200. Se il valore non è verificato, restituirà http/400.

Verifica CVV2

Example

In questo esempio, convalideremo un CVV/CVV2 per un determinato PAN. Il CVV2 viene in genere fornito dal titolare della carta o dall'utente durante la fase di convalida della transazione. Per convalidare i dati immessi, verranno forniti i seguenti valori in fase di esecuzione: [Key to Use for validation \(CVK\)](#), [data di scadenza della carta e CVV2](#) inseriti. [PAN](#) Il formato di scadenza della carta deve corrispondere a quello utilizzato nella generazione del valore iniziale.

Per tutti i parametri disponibili, consulta [CardVerificationValue2](#) nella guida di riferimento dell'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": "CADD1"
}
```

Verifica iCVV

Example

In questo esempio, verificheremo un [iCVV](#) per un determinato PAN inserendo i seguenti campi: [Key to Use for validation \(CVK\)](#), un codice di servizio 999[PAN](#), la data di scadenza della carta e l'iCVV fornito dalla transazione per la convalida.

iCVV non è un valore inserito dall'utente (come CVV2) ma è incorporato in una scheda EMV. Si dovrebbe valutare se debba sempre essere convalidato quando fornito.

Per tutti i parametri disponibili, consulta [CardVerificationValue1 nella guida](#) di riferimento dell'API.

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

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

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
tqv5yij6wtxx64pi",  
  "KeyCheckValue": "CADD1",  
  "ValidationData": "801"  
}
```

Generazione, traduzione e verifica dei dati PIN

Le funzioni relative ai dati PIN consentono di generare pin casuali, valori di verifica dei pin (PVV) e convalidare i pin crittografati in entrata rispetto a PVV o PIN Offset.

La traduzione dei pin consente di tradurre un pin da una chiave funzionante all'altra senza esporre il pin in testo non crittografato, come specificato dal requisito 1 del PCI PIN.

Note

Poiché la generazione e la convalida del PIN sono in genere funzioni dell'emittente e la traduzione del PIN è una tipica funzione di acquisizione, si consiglia di prendere in considerazione l'accesso con privilegi minimi e di impostare politiche appropriate per il caso d'uso del sistema.

Argomenti

- [Traduci i dati del PIN](#)
- [Genera dati PIN](#)
- [Verifica i dati del PIN](#)

Traduci i dati del PIN

Le funzioni Translate PIN data vengono utilizzate per tradurre i dati PIN crittografati da un set di chiavi a un altro senza che i dati crittografati escano dall'HSM. Viene utilizzato per la crittografia

P2PE, in cui le chiavi di lavoro devono cambiare, ma il sistema di elaborazione non deve o non è autorizzato a decrittografare i dati. Gli input principali sono i dati crittografati, la chiave di crittografia utilizzata per crittografare i dati, i parametri utilizzati per generare i valori di input. L'altro set di input è costituito dai parametri di output richiesti, come la chiave da utilizzare per crittografare l'output e i parametri utilizzati per creare quell'output. Gli output principali sono un set di dati appena crittografato e i parametri utilizzati per generarlo.

Note

I tipi di chiavi AES supportano solo blocchi a 4 [pin in](#) formato ISO.

Argomenti

- [PIN da PEK a DUKPT](#)
- [PIN da DUKPT a AWK](#)

PIN da PEK a DUKPT

Example

[In questo esempio, tradurremo un PIN dalla crittografia PEK TDES utilizzando un blocco PIN ISO 0 in un blocco PIN AES ISO 4 utilizzando l'algoritmo DUKPT.](#) In genere questa operazione può essere eseguita al contrario, ovvero un terminale di pagamento crittografa un pin in ISO 4 e poi può essere ritradotto in TDES per l'elaborazione a valle.

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

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

```

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

```

PIN da DUKPT a AWK

Example

[In questo esempio, tradurremo un PIN da un PIN crittografato AES DUKPT a un pin crittografato con AWK.](#) Dal punto di vista funzionale è l'inverso dell'esempio precedente.

```

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

```

```

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

```

Genera dati PIN

Le funzioni di generazione dei dati PIN vengono utilizzate per generare valori relativi al PIN, ad esempio gli offset [PVV](#) e pin block utilizzati per convalidare l'immissione dei pin da parte degli utenti durante la transazione o il momento dell'autorizzazione. Questa API può anche generare un nuovo pin casuale utilizzando vari algoritmi.

Genera Visa PVV per un pin

Example

In questo esempio, genereremo un nuovo pin (casuale) in cui gli output saranno crittografati PIN block (. PinData PinBlock) e un PVV (PinData.offset). Gli input chiave sono [PAN](#), the, the e. [Pin Verification Key Pin Encryption Key](#) PIN block format

Questo comando richiede che la chiave sia di tipo `TR31_V2_VISA_PIN_VERIFICATION_KEY`.

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

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

Genera l'offset dei pin IBM3624 per un pin

IBM 3624 PIN Offset, a volte chiamato anche metodo IBM. Questo metodo genera un PIN naturale/intermedio utilizzando i dati di convalida (in genere il PAN) e una chiave PIN (PVK). I pin naturali sono in effetti un valore derivato e, essendo deterministici, sono molto efficienti da gestire per l'emittente perché non è necessario archiviare i dati relativi ai pin a livello del titolare della carta. L'aspetto più evidente è che questo schema non tiene conto dei pin casuali o selezionabili dal titolare della carta. Per consentire questi tipi di pin, è stato aggiunto allo schema un algoritmo di offset. L'offset rappresenta la differenza tra il pin selezionato dall'utente (o casuale) e la chiave naturale. Il valore di offset viene memorizzato dall'emittente della carta o dal processore della carta. Al momento della transazione, il servizio AWS Payment Cryptography ricalcola internamente il pin naturale e

applica l'offset per trovare il pin. Quindi lo confronta con il valore fornito dall'autorizzazione della transazione.

Esistono diverse opzioni per IBM3624:

- `Ibm3624NaturalPin` metterà il pin naturale e un blocco pin crittografato
- `Ibm3624PinFromOffset` genererà un blocco pin crittografato dato un offset
- `Ibm3624RandomPin` genererà un pin casuale e quindi l'offset corrispondente e il blocco pin crittografato.
- `Ibm3624PinOffset` genera l'offset del pin in base a un pin selezionato dall'utente.

Internamente alla crittografia dei AWS pagamenti, vengono eseguiti i seguenti passaggi:

- Riempi il riquadro fornito a 16 caratteri. Se vengono forniti <16, compatta sul lato destro usando il carattere di padding fornito.
- Crittografa i dati di convalida utilizzando la chiave di generazione del PIN.
- Decimalizza i dati crittografati utilizzando la tabella di decimalizzazione. Questo mappa le cifre esadecimali in cifre decimali, ad esempio «A» può essere mappato a 9 e 1 può essere mappato a 1.
- Ottieni le prime 4 cifre da una rappresentazione esadecimale dell'output. Questa è la spilla naturale.
- Se è stato generato un pin selezionato dall'utente o casuale, il modulo sottrae il pin naturale con il pin del cliente. Il risultato è l'offset del pin.

Esempi

- [Genera l'offset dei pin IBM3624 per un pin](#)

Genera l'offset dei pin IBM3624 per un pin

In questo esempio, genereremo un nuovo pin (casuale) in cui le uscite saranno crittografate (. PIN block PinData PinBlock) e un valore di IBM3624 offset (pinData.offset). Gli input sono i dati di convalida (in genere il pan) `PAN`, il carattere di riempimento, `il`, `il` e il [Pin Verification Key Pin Encryption Key](#) PIN block format

Questo comando richiede che la chiave di generazione del pin sia di tipo TR31_V1_IBM3624_PIN_VERIFICATION_KEY e che la chiave di crittografia sia di tipo TR31_P0_PIN_ENCRYPTION_KEY

Example

L'esempio seguente mostra la generazione di un pin casuale, quindi l'emissione del blocco pin crittografato e del valore di offset IBM3624 utilizzando Ibm3624 RandomPin

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

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

Verifica i dati del PIN

Le funzioni di verifica dei dati del PIN vengono utilizzate per verificare la correttezza di un pin. Ciò comporta in genere il confronto del valore del pin precedentemente memorizzato con quello inserito dal titolare della carta in un POI. Queste funzioni confrontano due valori senza esporre il valore sottostante di nessuna delle due fonti.

Convalida il PIN crittografato utilizzando il metodo PVV

Example

In questo esempio, convalideremo un PIN per un determinato PAN. Il PIN viene in genere fornito dal titolare della carta o dall'utente durante il momento della transazione per la convalida e viene confrontato con il valore registrato (l'input del titolare della carta viene fornito come valore crittografato dal terminale o da altro provider a monte). Per convalidare questo input, in fase di esecuzione verranno forniti anche i seguenti valori: la chiave utilizzata per crittografare il pin di input (spesso si tratta di un IWK) [PAN](#) e il valore con cui eseguire la verifica (a o). PVV PIN offset

Se AWS Payment Cryptography è in grado di convalidare il pin, viene restituito un http/200. Se il pin non è convalidato, restituirà 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",
}
```

Convalida un PIN rispetto all'offset dei pin IBM3624 precedentemente memorizzato

In questo esempio, convalideremo il PIN fornito dal titolare della carta in base all'offset del pin memorizzato in un file presso l'emittente/processore della carta. Gli input sono simili all'???aggiunta del pin crittografato fornito dal terminale di pagamento (o da un altro provider a monte come Card Network). Se il pin corrisponde, l'api restituirà http 200. dove gli output saranno crittografati (. PIN block PinData PinBlock) e un valore di IBM3624 offset (pinData.offset).

Questo comando richiede che la chiave di generazione del pin sia di tipo TR31_V1_IBM3624_PIN_VERIFICATION_KEY e che la chiave di crittografia sia di tipo TR31_P0_PIN_ENCRYPTION_KEY

Example

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

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

Crittogramma Verify Auth Request (ARQC)

[L'API del crittogramma di verifica della richiesta di autenticazione viene utilizzata per verificare l'ARQC.](#) La generazione dell'ARQC non rientra nell'ambito della crittografia dei AWS pagamenti e viene generalmente eseguita su una Chip Card EMV (o un equivalente digitale come un portafoglio mobile) durante il periodo di autorizzazione della transazione. Un ARQC è unico per ogni transazione e ha lo scopo di mostrare crittograficamente sia la validità della carta sia di garantire che i dati della transazione corrispondano esattamente alla transazione corrente (prevista).

AWS La crittografia dei pagamenti offre una varietà di opzioni per la convalida dell'ARQC e la generazione di valori ARQC opzionali, inclusi quelli definiti in [EMV 4.4 Book 2](#) e altri schemi utilizzati da Visa e Mastercard. [Per un elenco completo di tutte le opzioni disponibili, consulta la sezione della Guida API. VerifyCardValidationData](#)

I crittogrammi ARQC richiedono in genere i seguenti input (sebbene ciò possa variare in base all'implementazione):

- [PAN](#) - Specificato nel campo PrimaryAccountNumber
- [Numero di sequenza PAN \(PSN\)](#) - specificato nel campo PanSequenceNumber
- Metodo di derivazione delle chiavi come Common Session Key (CSK) - Specificato nel SessionKeyDerivationAttributes
- Modalità di derivazione della chiave principale (ad esempio EMV Option A): specificata nella MajorKeyDerivationMode
- Dati sulle transazioni: una stringa di vari dati relativi a transazioni, terminali e carte, come Importo e Data, specificati nel campo TransactionData
- [Chiave principale dell'emittente](#): la chiave master utilizzata per derivare la chiave crittografica (AC) utilizzata per proteggere le singole transazioni e specificata nel campo KeyIdentifier

Argomenti

- [Creazione di dati sulle transazioni](#)
- [Riempimento dei dati delle transazioni](#)
- [Esempi](#)

Creazione di dati sulle transazioni

Il contenuto esatto (e l'ordine) del campo di dati della transazione varia in base all'implementazione e allo schema di rete, ma i campi minimi consigliati (e la sequenza di concatenazione) sono definiti nel [Libro 2 di EMV 4.4, Sezione 8.1.1](#) - Selezione dei dati. Se i primi tre campi sono importo (17,00), altro importo (0,00) e paese di acquisto, i dati della transazione inizierebbero come segue:

- 000000001700 - importo - 12 posizioni implicavano un decimale a due cifre
- 000000000000 - altro importo - 12 posizioni implicavano un decimale a due cifre
- 0124 - prefisso internazionale a quattro cifre
- Dati di transazione (parziali) in uscita - 00000000170000000000000000124

Riempimento dei dati delle transazioni

I dati delle transazioni devono essere aggiunti prima dell'invio al servizio. La maggior parte degli schemi utilizza il padding ISO 9797 Metodo 2, in cui una stringa esadecimale viene aggiunta dall'esadecimale 80 seguito da 00 fino a quando il campo non è un multiplo della dimensione del blocco di crittografia; 8 byte o 16 caratteri per TDES e 16 byte o 32 caratteri per AES. L'alternativa (metodo 1) non è così comune, ma utilizza solo 00 come caratteri di riempimento.

ISO 9797 Metodo 1: Imbottitura

Senza imbottitura:

00000000000000000000000008400080008000084016051700000000093800000B03011203 (74 caratteri o 37 byte)

Imbottito: 00000000000000000000000008400080008000084016051700000000093800000B03011203 000000 (80 caratteri o 40 byte)

Imbottitura ISO 9797 Metodo 2

Senza imbottitura:

00000000000000000000000008400080008000084016051700000000093800000B1F220103000000 (80 caratteri o 40 byte)

Imbottito: 00000000000000000000000008400080008000084016051700000000093800000B1F220103000000 8000000000000000 (88 caratteri o 44 byte)

Esempi

Visa CVN10

Example

In questo esempio, convalideremo un ARQC generato utilizzando Visa CVN10.

Se AWS Payment Cryptography è in grado di convalidare l'ARQC, viene restituito un http/200. Se l'arqc non è convalidato, restituirà una risposta http/400.

```
$ aws payment-cryptography-data verify-auth-request-cryptogram --auth-request-cryptogram D791093C8A921769 \
```

```
--key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6n162t5ushfk \
--major-key-derivation-mode EMV_OPTION_A \
--transaction-data
000000001700000000000000000008400080008000084016051700000000093800000B03011203000000 \
--session-key-derivation-attributes='{"Visa":{"PanSequenceNumber":"01" \
,"PrimaryAccountNumber":"9137631040001422"}}'
```

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

Visa CVN18 e Visa CVN22

Example

In questo esempio, convalideremo un ARQC generato utilizzando Visa CVN18 o CVN22. Le operazioni crittografiche sono le stesse tra CVN18 e CVN22, ma i dati contenuti nei dati delle transazioni variano. Rispetto a CVN10, viene generato un crittogramma completamente diverso anche con gli stessi input.

Se AWS Payment Cryptography è in grado di convalidare l'ARQC, viene restituito un http/200. Se l'arqc non è convalidato, restituirà un http/400.

```
$ aws payment-cryptography-data verify-auth-request-cryptogram \
--auth-request-cryptogram 61EDCC708B4C97B4
--key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
pw3s6n162t5ushfk \
--major-key-derivation-mode EMV_OPTION_A
--transaction-data
000000001700000000000000000008400080008000084016051700000000093800000B1F220103000000000000
\
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"  
}
```

Genera e verifica MAC

I codici di autenticazione dei messaggi (MAC) vengono in genere utilizzati per autenticare l'integrità di un messaggio (indipendentemente dal fatto che sia stato modificato). Gli hash crittografici come HMAC (Hash-Based Message Authentication Code), CBC-MAC e CMAC (Cipher-Based Message Authentication Code) forniscono inoltre un'ulteriore garanzia del mittente del MAC utilizzando la crittografia. HMAC si basa su funzioni hash mentre CMAC si basa su cifrari a blocchi.

Tutti gli algoritmi MAC di questo servizio combinano una funzione hash crittografica e una chiave segreta condivisa. Accettano un messaggio e una chiave segreta, ad esempio il materiale chiave contenuto in una chiave, e restituiscono un tag o mac univoco. Se anche solo un carattere del messaggio cambia o se la chiave segreta cambia, il tag risultante è completamente diverso. Richiedendo una chiave segreta, i MAC crittografici garantiscono anche l'autenticità; è impossibile generare un mac identico senza la chiave segreta. I MAC crittografici sono talvolta chiamati firme simmetriche, perché funzionano come le firme digitali, ma utilizzano un'unica chiave sia per la firma che per la verifica.

AWSPayment Cryptography supporta diversi tipi di MAC:

ALGORITMO ISO9797 1

Denotato con o ISO9797_ALGORITHM1 KeyUsage

ALGORITMO ISO9797 3 (RETAIL MAC)

Denotato da ISO9797_ALGORITHM3 KeyUsage

ALGORITMO ISO9797 5 (CMAC)

Denotato con o TR31_M6_ISO_9797_5_CMAC_KEY KeyUsage

HMAC

Denotato con KeyUsage TR31_M7_HMAC_KEY, inclusi HMAC_SHA224, HMAC_SHA256, HMAC_SHA384 e HMAC_SHA512

Argomenti

- [Genera MAC](#)

- [Verifica MAC](#)

Genera MAC

L'API Generate MAC viene utilizzata per autenticare i dati relativi alle carte, ad esempio tracciare i dati provenienti dalla banda magnetica di una scheda, utilizzando valori di dati noti per generare un MAC (Message Authentication Code) per la convalida dei dati tra le parti che inviano e ricevono. I dati utilizzati per generare un MAC includono i dati dei messaggi, la chiave di crittografia MAC segreta e l'algoritmo MAC per generare un valore MAC univoco per la trasmissione. La parte ricevente del MAC utilizzerà gli stessi dati dei messaggi MAC, la stessa chiave di crittografia MAC e lo stesso algoritmo per riprodurre un altro valore MAC per il confronto e l'autenticazione dei dati. Anche se un carattere del messaggio cambia o la chiave MAC utilizzata per la verifica non è identica, il valore MAC risultante è diverso. L'API supporta le chiavi di crittografia DUPKT MAC, HMAC ed EMV MAC per questa operazione.

Il valore di input per message-data deve essere un dato HexBinary.

In questo esempio, genereremo un HMAC (Hash-Based Message Authentication Code) per l'autenticazione dei dati delle carte utilizzando l'algoritmo HMAC HMAC_SHA256 e la chiave di crittografia HMAC. La chiave deve essere KeyUsage impostata su e su. TR31_M7_HMAC_KEY KeyModesOfUse Generate La chiave MAC può essere creata con AWS Payment Cryptography chiamando [CreateKey](#) o importata [ImportKey](#) chiamando.

Example

```
$ aws payment-cryptography-data generate-mac \  
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/  
qno151ghrzunce6 \  
  --message-data  
  "3b313038383439303031303733393431353d32343038323236303030373030303f33" \  
  --generation-attributes Algorithm=HMAC_SHA256
```

```
{  
  "KeyArn": "arn:aws:payment-cryptography:us-east-2:111122223333:key/  
qno151ghrzunce6,  
  "KeyCheckValue": "2976E7",  
  "Mac": "ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C"  
}
```

Verifica MAC

L'API Verify MAC viene utilizzata per verificare il MAC (Message Authentication Code) per l'autenticazione dei dati relativi alla carta. Deve utilizzare la stessa chiave di crittografia utilizzata durante la generazione del MAC per riprodurre il valore MAC per l'autenticazione. La chiave di crittografia MAC può essere creata con AWS Payment Cryptography chiamando [CreateKey](#) importata chiamando [ImportKey](#). L'API supporta le chiavi di crittografia DUPKT MAC, HMAC ed EMV MAC per questa operazione.

Se il valore è verificato, `MacDataVerificationSuccessful` verrà restituito il parametro di risposta `Http/200`, altrimenti `Http/400` con un messaggio che lo indica. `Mac verification failed`

In questo esempio, verificheremo un HMAC (Hash-Based Message Authentication Code) per l'autenticazione dei dati delle carte utilizzando l'algoritmo HMAC HMAC_SHA256 e la chiave di crittografia HMAC. La chiave deve essere KeyUsage impostata su `e su. TR31_M7_HMAC_KEY` `KeyModesOfUse Verify`

Example

```
$ aws payment-cryptography-data verify-mac \
  --key-identifier arn:aws:payment-cryptography:us-east-2:111122223333:key/
  qnobl5lghrzunce6 \
  --message-data
  "3b343038383439303031303733393431353d32343038323236303030373030303f33" \
  --verification-attributes='Algorithm=HMAC_SHA256' \
  --mac ED87F26E961C6D0DDB78DA5038AA2BDDEA0DCE03E5B5E96BDDD494F4A7AA470C
```

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

Chiavi valide per operazioni crittografiche

Alcune chiavi possono essere utilizzate solo per determinate operazioni. Inoltre, alcune operazioni possono limitare le modalità di utilizzo dei tasti. Consulta la tabella seguente per le combinazioni consentite.

Note

Alcune combinazioni, sebbene consentite, possono creare situazioni inutilizzabili, come la generazione di codici CVV, che (generate) però non possono essere verificate. (verify)

Argomenti

- [GenerateCardDati](#)
- [VerifyCardDati](#)
- [GeneratePinData \(per schemi VISA/ABA\)](#)
- [GeneratePinData \(perIBM3624\)](#)
- [VerifyPinData \(per schemi VISA/ABA\)](#)
- [VerifyPinData \(perIBM3624\)](#)
- [Decrittografia dei dati](#)
- [Encrypt Data \(Crittografa dati\)](#)
- [Traduci PIN Data](#)
- [Genera/verifica MAC](#)
- [VerifyAuthRequestCryptogram](#)
- [Chiave Import/Export](#)
- [Tipi di chiavi non utilizzati](#)

GenerateCardDati

Endpoint API	Operazione o algoritmo crittografico	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
GenerateCardDati	<ul style="list-style-type: none"> • AMEX_CARD_SECURITY_CODE_VERIFICATION_1 	TR31_C0_CARD_CHIAVE_VERIFY_DELETE_CARD_	<ul style="list-style-type: none"> • CHIAVE_TDES_2 • TDES_3KEY 	{Generate = true}, {Generate = true, Verify = true}

Endpoint API	Operazione o algoritmo crittografico	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazioni consentite delle principali modalità di utilizzo
	<ul style="list-style-type: none"> VERSIONE_CODE_DEL_SICUREZZA_AAMEX_CARD_2 			
GenerateCardData	<ul style="list-style-type: none"> CARD_VERIFICATION_VALUE_1 VALORE_VERIFICA_DELLA CARTA 	TR31_C0_CARD_VERIFICATION_KEY	<ul style="list-style-type: none"> CHIAVE TDES_2 	{Generate = true}, {Generate = true, Verify = true}
GenerateCardData	<ul style="list-style-type: none"> CARDHOLDER_AUTHENTICATION_VERIFICATION_VALUE 	TR31_E6_EMV_MKEY_ALTRO	<ul style="list-style-type: none"> CHIAVE TDES_2 	{= vero} DeriveKey
GenerateCardData	<ul style="list-style-type: none"> DYNAMIC_CARD_VERIFICATION_CODE 	TR31_E4_EMV_MKEY_NUMERIAMICI	<ul style="list-style-type: none"> CHIAVE TDES_2 	{= vero} DeriveKey
GenerateCardData	<ul style="list-style-type: none"> DYNAMIC_CARD_VERIFICATION_VALUE 	TR31_E6_EMV_MKEY_ALTRO	<ul style="list-style-type: none"> CHIAVE TDES_2 	{= vero} DeriveKey

VerifyCardDati

Operazione o algoritmo crittografico	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
<ul style="list-style-type: none"> AMEX_CARD_SECURITY_CODE_VERSION_1 VERSIONE_CODE_DELLA_SICUREZZA_AMEX_CARD_2 	TR31_C0_CARD_VERIFICATION_KEY	<ul style="list-style-type: none"> CHIAVE TDES_2 TDES_3KEY 	{Generate = true}, {Generate = true, Verify = true}
<ul style="list-style-type: none"> CARD_VERIFICATION_VALUE_1 VALORE_VERIFICA_2 DELLA CARTA 	TR31_C0_CARD_VERIFICATION_KEY	<ul style="list-style-type: none"> CHIAVE TDES_2 	{Generate = true}, {Generate = true, Verify = true}
<ul style="list-style-type: none"> CARDHOLDER_AUTHENTICATION_VERIFICATION_VALUE 	TR31_E6_MV_MKEY_ALTRO	<ul style="list-style-type: none"> CHIAVE TDES_2 	{= vero} DeriveKey
<ul style="list-style-type: none"> CODICE_DINAMICO_CARD_VERIFICATION_CODE 	TR31_E4_MV_MKEY_NUMERI_DINAMICI	<ul style="list-style-type: none"> CHIAVE TDES_2 	{= vero} DeriveKey
<ul style="list-style-type: none"> VALORE_DIVERIFICA 	TR31_E6_MV_MKEY_ALTRO	<ul style="list-style-type: none"> CHIAVE TDES_2 	{= vero} DeriveKey

Operazione o algoritmo crittografico	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
ZIONE_DINAMICO DELLA CARTA			

GeneratePinData (per schemi VISA/ABA)

VISA_PIN or VISA_PIN_VERIFICATION_VALUE

Tipo di chiavi	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
Chiave di crittografia PIN	TR31_P0_P IN_ENCRYPT TION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • TDES_3KEY 	<ul style="list-style-type: none"> • {Crittografa = vero, Wrap = vero} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • {= vero} NoRestrictions
Chiave di generazione del PIN	TR31_V2_V ISA_PIN_VERIFICATI ON_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_3 	<ul style="list-style-type: none"> • {Genera = vero} • {Genera = vero, verifica = vero}

GeneratePinData (per **IBM3624**)

IBM3624_PIN_OFFSET, IBM3624_NATURAL_PIN, IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET)

Tipo di chiavi	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
Chiave di crittografia PIN	TR31_P0_P IN_ENCRYPT TION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • TDES_3KEY 	<p>Per IBM3624_N ATURAL_P N, IBM3624_R ANDOM_P IN, IBM3624_P IN_FROM_OFFSET</p> <ul style="list-style-type: none"> • {Encrypt = true, Wrap = true} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • {= vero} NoRestrictions <p>Per IBM3624_P IN_OFFSET</p> <ul style="list-style-type: none"> • {Encrypt = true, Unwrap = true} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • {= vero} NoRestrictions
Chiave di generazione del PIN	TR31_V1_I BM3624_PI	<ul style="list-style-type: none"> • CHIAVE TDES_3 	<ul style="list-style-type: none"> • {Genera = vero} • {Genera = vero, verifica = vero}

Tipo di chiavi	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
	N_VERIFICATION_KEY		

VerifyPinData (per schemi VISA/ABA)

VISA_PIN

Tipo di chiavi	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
Chiave di crittografia PIN	TR31_P0_PIN_ENCRYPTION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • TDES_3KEY 	<ul style="list-style-type: none"> • {Decrypt = true, Unwrap = true} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • {= vero} NoRestrictions
Chiave di generazione del PIN	TR31_V2_VISA_PIN_VERIFICATION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_3 	<ul style="list-style-type: none"> • {Verifica = vero} • {Genera = vero, verifica = vero}

VerifyPinData (per **IBM3624**)

IBM3624_PIN_OFFSET, IBM3624_NATURAL_PIN, IBM3624_RANDOM_PIN, IBM3624_PIN_FROM_OFFSET)

Tipo di chiavi	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
Chiave di crittografia PIN	TR31_P0_P IN_ENCRYPT TION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • TDES_3KEY 	Per IBM3624_N ATURAL_PI N, IBM3624_R ANDOM_PIN , IBM3624_P IN_FROM_OFFSET <ul style="list-style-type: none"> • {Decrypt = true, Unwrap = true} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • {= vero} NoRestrictions
Chiave di verifica del PIN	TR31_V1_I BM3624_PI N_VERIFIC ATION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_3 	<ul style="list-style-type: none"> • {Verifica = vero} • {Genera = vero, verifica = vero}

Decrittografia dei dati

Tipo di chiavi	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
DISCARICA	TR31_B0_B ASE_DERIV ATION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • AES_128 	<ul style="list-style-type: none"> • {= vero} DeriveKey

Tipo di chiavi	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
		<ul style="list-style-type: none"> AES_192 AES_256 	<ul style="list-style-type: none"> { NoRestrictions = vero }
EMV	TR31_E1_E MV_MKEY_C ONFIDENTIALITY TR31_E6_E MV_MKEY_ALTRO	<ul style="list-style-type: none"> CHIAVE TDES_2 	<ul style="list-style-type: none"> {= vero} DeriveKey
RSA	TR31_D1_K EY_ASYMME TRIC_FOR_ DATA_ENCRYPTION	<ul style="list-style-type: none"> RSA_2048 RSA_3072 RSA_4096 	<ul style="list-style-type: none"> {Decrypt = true, unwrap=True} {encrypt=True, wrap=True, Decrypt = vero, unwrap=vero}
Chiavi simmetriche	TR31_D0_S YMMETRIC_ DATA_ENCR YPTION_KEY	<ul style="list-style-type: none"> CHIAVE TDES_2 TDES_3KEY AES_128 AES_192 AES_256 	<ul style="list-style-type: none"> {Decrypt = true, unwrap=true} {encrypt=True, wrap=True, Decrypt = vero, unwrap=vero} NoRestrictions {= vero}

Encrypt Data (Crittografia dati)

Tipo di chiavi	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
DISCARICA	TR31_B0_B ASE_DERIV ATION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • AES_128 • AES_192 • AES_256 	<ul style="list-style-type: none"> • {= vero} DeriveKey • { NoRestrictions = vero}
EMV	TR31_E1_E MV_MKEY_C ONFIDENTIALITY TR31_E6_E MV_MKEY_ALTRO	<ul style="list-style-type: none"> • CHIAVE TDES_2 	<ul style="list-style-type: none"> • {= vero} DeriveKey
RSA	TR31_D1_K EY_ASYMME TRIC_FOR_ DATA_ENCRYPTION	<ul style="list-style-type: none"> • RSA_2048 • RSA_3072 • RSA_4096 	<ul style="list-style-type: none"> • {Crittografia = vero, wrap=vero} • {encrypt=True, wrap=True, Decrypt = true, unwrap=True}
Chiavi simmetriche	TR31_D0_S YMMETRIC_ DATA_ENCR YPTION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • TDES_3KEY • AES_128 • AES_192 • AES_256 	<ul style="list-style-type: none"> • {Crittografia = vero, wrap=vero} • {encrypt=True, wrap=True, Decrypt = true, unwrap=True} • NoRestrictions {= vero}

Traduci PIN Data

Direzione	Tipo di chiavi	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
Fonte di dati in entrata	DUMPT	TR31_B0_B ASE_DERIV ATION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • AES_128 • AES_192 • AES_256 	<ul style="list-style-type: none"> • {= vero} DeriveKey • { NoRestrictions = vero}
Fonte di dati in entrata	Non DUPPT (PEK, AWK, IWK, ecc.)	TR31_P0_P IN_ENCRYP TION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • TDES_3KEY • AES_128 • AES_192 • AES_256 	<ul style="list-style-type: none"> • {Decrypt = true, Unwrap = true} • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} • {= vero} NoRestrictions
Target dati in uscita	DUPPT	TR31_B0_B ASE_DERIV ATION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • AES_128 • AES_192 • AES_256 	<ul style="list-style-type: none"> • {= vero} DeriveKey • { NoRestrictions = vero}
Target dati in uscita	Non DUPPT (PEK, IWK, AWK, ecc.)	TR31_P0_P IN_ENCRYP TION_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • TDES_3KEY • AES_128 	<ul style="list-style-type: none"> • {Crittografia = vero, Wrap = vero}

Direzione	Tipo di chiavi	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
			<ul style="list-style-type: none"> AES_192 AES_256 	<ul style="list-style-type: none"> {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true} {= vero} NoRestrictions

Genera/verifica MAC

Le chiavi MAC vengono utilizzate per creare hash crittografici di un messaggio/corpo di dati. Non è consigliabile creare una chiave con modalità di utilizzo limitate in quanto non sarà possibile eseguire l'operazione di abbinamento. Tuttavia, è possibile importare/esportare una chiave con una sola operazione se l'altro sistema è destinato a eseguire l'altra metà della coppia di operazioni.

Utilizzo delle chiavi consentito	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
Chiave MAC	TR31_M1_I SO_9797_1 _MAC_KEY	<ul style="list-style-type: none"> CHIAVE TDES_2 TDES_3KEY 	<ul style="list-style-type: none"> {Genera = vero} {Genera = vero, verifica = vero} {Verifica = vero} {Genera = vero}
Chiave MAC (MAC per la vendita al dettaglio)	TR31_M1_I SO_9797_3 _MAC_KEY	<ul style="list-style-type: none"> CHIAVE TDES_2 TDES_3KEY 	<ul style="list-style-type: none"> {Genera = vero} {Genera = vero, verifica = vero}

Utilizzo delle chiavi consentito	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
			<ul style="list-style-type: none"> • {Verifica = vero} • {Genera = vero}
Chiave MAC (CMAC)	TR31_M6_I SO_9797_5 _CMAC_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • TDES_3KEY • AES_128 • AES_192 • AES_256 	<ul style="list-style-type: none"> • {Genera = vero} • {Genera = vero, verifica = vero} • {Verifica = vero} • {Genera = vero}
Chiave MAC (HMAC)	TR31_M7_H MAC_KEY	<ul style="list-style-type: none"> • CHIAVE TDES_2 • TDES_3KEY • AES_128 • AES_192 • AES_256 	<ul style="list-style-type: none"> • {Genera = vero} • {Genera = vero, verifica = vero} • {Verifica = vero} • {Genera = vero}

VerifyAuthRequestCryptogram

Utilizzo delle chiavi consentito	Opzione EMV	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
<ul style="list-style-type: none"> • OPZIONE A • OPZIONE B 	TR31_E0_E MV_MKEY_A PP_CRYPTOGRAMS	<ul style="list-style-type: none"> • CHIAVE TDES_2 	<ul style="list-style-type: none"> • {= vero} DeriveKey

Chiave Import/Export

Tipo di operazione	Utilizzo delle chiavi consentito	Algoritmo a chiave consentito	Combinazione consentita delle principali modalità di utilizzo
Chiave avvolgente TR-31	TR31_K1_KEY_BLOCK_PROTECTION_KEY TR31_K0_CHIAVE_DIGITALE_CRITTOGRAFIA	<ul style="list-style-type: none"> • CHIAVE TDES_2 • TDES_3KEY • AES_128 	<ul style="list-style-type: none"> • {Encrypt = true, Wrap = true} (solo esportazione) • {Decrypt = true, Unwrap = true} (solo importazione) • {Encrypt = true, Decrypt = true, Wrap = true, Unwrap = true}
Importazione di CA affidabili	TR31_S0_KEY_ASYMMETRIC_FOR_DIGITAL_SIGNATURE	<ul style="list-style-type: none"> • RSA_2048 • RSA_3072 • RSA_4096 	<ul style="list-style-type: none"> • {Verifica = vero}
Importazione di un certificato a chiave pubblica per la crittografia asimmetrica	TR31_D1_KEY_ASYMMETRIC_FOR_DATA_ENCRYPTION	<ul style="list-style-type: none"> • RSA_2048 • RSA_3072 • RSA_4096 	<ul style="list-style-type: none"> • {encrypt=Vero, wrap=Vero}

Tipi di chiavi non utilizzati

I seguenti tipi di chiave non sono attualmente utilizzati da AWS Payment Cryptography

- TR31_P1_PIN_GENERATION_KEY
- TR31_K3_CHIAVE_ASIMMETRICA PER CONTRATTO_CHIAVE

Sicurezza nella crittografia dei pagamenti AWS

La sicurezza del cloud AWS è la massima priorità. In qualità di AWS cliente, puoi beneficiare di un data center e di un'architettura di rete progettati per soddisfare i requisiti delle organizzazioni più sensibili alla sicurezza.

La sicurezza è una responsabilità condivisa tra AWS te e te. Il [modello di responsabilità condivisa](#) descrive questo modello come sicurezza del cloud e sicurezza nel cloud:

- **Sicurezza del cloud:** AWS è responsabile della protezione dell'infrastruttura che gestisce AWS i servizi nel AWS cloud. AWS ti fornisce anche servizi che puoi utilizzare in modo sicuro. I revisori esterni testano e verificano regolarmente l'efficacia della nostra sicurezza nell'ambito dei [AWS Programmi di AWS conformità dei Programmi di conformità](#) dei di . Per ulteriori informazioni sui programmi di conformità che si applicano alla crittografia dei AWS pagamenti, consulta [AWS Services in Scope by Compliance Program](#) .
- **Sicurezza nel cloud:** la tua responsabilità è determinata dal AWS servizio che utilizzi. Sei anche responsabile di altri fattori, tra cui la riservatezza dei dati, i requisiti della tua azienda e le leggi e normative vigenti.

Questo argomento ti aiuta a capire come applicare il modello di responsabilità condivisa quando usi la crittografia dei AWS pagamenti. Ti mostra come configurare la crittografia dei AWS pagamenti per soddisfare i tuoi obiettivi di sicurezza e conformità. Imparerai anche come utilizzare altri AWS servizi che ti aiutano a monitorare e proteggere le tue risorse AWS di crittografia dei pagamenti.

Argomenti

- [Protezione dei dati nella crittografia dei AWS pagamenti](#)
- [Resilienza nella crittografia dei pagamenti AWS](#)
- [Sicurezza dell'infrastruttura in AWS Payment Cryptography](#)
- [Connessione alla crittografia dei AWS pagamenti tramite un endpoint VPC](#)
- [Le migliori pratiche di sicurezza per la crittografia dei pagamenti AWS](#)

Protezione dei dati nella crittografia dei AWS pagamenti

Il modello di [responsabilità AWS condivisa modello](#) si applica alla protezione dei dati nella crittografia dei AWS pagamenti. Come descritto in questo modello, AWS è responsabile della protezione

dell'infrastruttura globale che gestisce tutti i Cloud AWS. L'utente è responsabile del controllo dei contenuti ospitati su questa infrastruttura. L'utente è inoltre responsabile della configurazione della protezione e delle attività di gestione per i Servizi AWS utilizzati. Per ulteriori informazioni sulla privacy dei dati, vedi le [Domande frequenti sulla privacy dei dati](#). Per informazioni sulla protezione dei dati in Europa, consulta il post del blog relativo al [Modello di responsabilità condivisa AWS e GDPR](#) nel Blog sulla sicurezza AWS .

Ai fini della protezione dei dati, consigliamo di proteggere Account AWS le credenziali e configurare i singoli utenti con AWS IAM Identity Center or AWS Identity and Access Management (IAM). In tal modo, a ogni utente verranno assegnate solo le autorizzazioni necessarie per svolgere i suoi compiti. Ti suggeriamo, inoltre, di proteggere i dati nei seguenti modi:

- Utilizza l'autenticazione a più fattori (MFA) con ogni account.
- Usa SSL/TLS per comunicare con le risorse. AWS È richiesto TLS 1.2 ed è consigliato TLS 1.3.
- Configura l'API e la registrazione delle attività degli utenti con. AWS CloudTrail
- Utilizza soluzioni di AWS crittografia, insieme a tutti i controlli di sicurezza predefiniti all'interno Servizi AWS.
- Utilizza i servizi di sicurezza gestiti avanzati, come Amazon Macie, che aiutano a individuare e proteggere i dati sensibili archiviati in Amazon S3.
- Se hai bisogno di moduli crittografici convalidati FIPS 140-2 per l'accesso AWS tramite un'interfaccia a riga di comando o un'API, utilizza un endpoint FIPS. Per ulteriori informazioni sugli endpoint FIPS disponibili, consulta il [Federal Information Processing Standard \(FIPS\) 140-2](#).

Ti consigliamo vivamente di non inserire mai informazioni riservate o sensibili, ad esempio gli indirizzi e-mail dei clienti, nei tag o nei campi di testo in formato libero, ad esempio nel campo Nome. Ciò include quando lavori con AWS Payment Cryptography o altro Servizi AWS utilizzando la console, l'API o gli SDK. AWS CLI AWS I dati inseriti nei tag o nei campi di testo in formato libero utilizzati per i nomi possono essere utilizzati per la fatturazione o i log di diagnostica. Quando fornisci un URL a un server esterno, ti suggeriamo vivamente di non includere informazioni sulle credenziali nell'URL per convalidare la tua richiesta al server.

AWS Payment Cryptography archivia e protegge le chiavi di crittografia dei pagamenti per renderle altamente disponibili e allo stesso tempo fornirti un controllo degli accessi solido e flessibile.

Argomenti

- [Protezione del materiale della chiave](#)

- [Crittografia dei dati](#)
- [Crittografia a riposo](#)
- [Crittografia in transito](#)
- [Riservatezza del traffico Internet](#)

Protezione del materiale della chiave

Per impostazione predefinita, AWS Payment Cryptography protegge il materiale delle chiavi crittografiche per le chiavi di pagamento gestite dal servizio. Inoltre, AWS Payment Cryptography offre opzioni per importare materiale chiave creato al di fuori del servizio. Per dettagli tecnici sulle chiavi di pagamento e sul materiale chiave, consulta [AWS Payment Cryptography Cryptographic Details](#).

Crittografia dei dati

I dati in AWS Payment Cryptography sono costituiti da chiavi di crittografia AWS Payment, dal materiale chiave di crittografia che rappresentano e dai relativi attributi di utilizzo. Il materiale chiave è disponibile in testo semplice solo all'interno dei moduli di sicurezza hardware (HSM) di AWS Payment Cryptography e solo quando è in uso. Altrimenti, il materiale e gli attributi chiave vengono crittografati e archiviati in uno storage persistente durevole.

Il materiale chiave che AWS Payment Cryptography genera o carica per le chiavi di pagamento non esce mai dai confini degli HSM di AWS Payment Cryptography non crittografati. Può essere esportato crittografato dalle operazioni dell'API AWS Payment Cryptography.

Crittografia a riposo

AWS Payment Cryptography genera materiale chiave per le chiavi di pagamento negli HSM elencati in PCI PTS HSM. Quando non in uso, il materiale della chiave viene crittografato da una chiave HSM e scritto in uno storage persistente e durevole. Il materiale chiave per le chiavi di crittografia dei pagamenti e le chiavi di crittografia che proteggono il materiale chiave non lasciano mai gli HSM in forma di testo semplice.

La crittografia e la gestione del materiale chiave per le chiavi di crittografia dei pagamenti sono gestite interamente dal servizio.

Per ulteriori dettagli, consulta i dettagli crittografici di [AWS Key Management Service](#).

Crittografia in transito

Il materiale chiave che AWS Payment Cryptography genera o carica per le chiavi di pagamento non viene mai esportato o trasmesso nelle operazioni dell'API AWS Payment Cryptography in chiaro. AWS Payment Cryptography utilizza identificatori di chiave per rappresentare le chiavi nelle operazioni API.

Tuttavia, alcune operazioni dell'API AWS Payment Cryptography esportano chiavi crittografate da una chiave di scambio di chiavi precedentemente condivisa o asimmetrica. Inoltre, i clienti possono utilizzare le operazioni API per importare materiale con chiavi crittografate per le chiavi di pagamento.

Tutte le chiamate API AWS Payment Cryptography devono essere firmate e trasmesse utilizzando Transport Layer Security (TLS). AWS Payment Cryptography richiede versioni TLS e suite di crittografia definite da PCI come «crittografia avanzata». Tutti gli endpoint di servizio supportano TLS 1.0—1.3 e TLS ibrido post-quantistico.

Per ulteriori dettagli, consulta i dettagli crittografici di AWS Key Management Service.

Riservatezza del traffico Internet

AWS Payment Cryptography supporta una console di gestione AWS e una serie di operazioni API che consentono di creare e gestire chiavi di pagamento e utilizzarle in operazioni crittografiche.

AWS Payment Cryptography supporta due opzioni di connettività di rete dalla tua rete privata ad AWS.

- Una connessione VPN IPsec su Internet.
- AWS Direct Connect, che collega la rete interna a una posizione AWS Direct Connect tramite un cavo Ethernet standard in fibra ottica.

Tutte le chiamate all'API Payment Cryptography devono essere firmate e trasmesse utilizzando Transport Layer Security (TLS). Le chiamate richiedono anche una moderna suite di cifratura che supporta la perfect forward secrecy. Il traffico verso i moduli di sicurezza hardware (HSM) che memorizzano il materiale chiave per le chiavi di pagamento è consentito solo da host API AWS Payment Cryptography noti sulla rete interna AWS.

Per connetterti direttamente alla crittografia di AWS Payment dal tuo cloud privato virtuale (VPC) senza inviare traffico su Internet pubblico, utilizza gli endpoint VPC, con tecnologia AWS PrivateLink. Per ulteriori informazioni, consulta Connessione alla crittografia di AWS Payment tramite un endpoint VPC.

AWS Payment Cryptography supporta anche un'opzione ibrida di scambio di chiavi post-quantistiche per il protocollo di crittografia di rete Transport Layer Security (TLS). Puoi utilizzare questa opzione con TLS quando ti connetti agli endpoint dell'API AWS Payment Cryptography.

Resilienza nella crittografia dei pagamenti AWS

AWS l'infrastruttura globale è costruita attorno a AWS regioni e zone di disponibilità. Le regioni forniscono più zone di disponibilità fisicamente separate e isolate, connesse tramite reti altamente ridondanti, a bassa latenza e throughput elevato. Con le zone di disponibilità, è possibile progettare e gestire applicazioni e database che eseguono il failover automatico tra zone di disponibilità senza interruzioni. Le zone di disponibilità sono più disponibili, tolleranti ai guasti e scalabili rispetto alle infrastrutture a data center singolo o multiplo tradizionali.

Per ulteriori informazioni su AWS regioni e zone di disponibilità, consulta [Infrastruttura AWS globale](#).

Isolamento regionale

AWS Payment Cryptography è un servizio regionale disponibile in più regioni.

Il design isolato a livello regionale di AWS Payment Cryptography garantisce che un problema di disponibilità in una regione AWS non possa influire sul funzionamento della crittografia di AWS Payment in nessun'altra regione. AWS Payment Cryptography è progettata per garantire zero tempi di inattività pianificati, con tutti gli aggiornamenti software e le operazioni di scalabilità eseguiti senza interruzioni e impercettibili.

L'accordo sul livello di servizio (SLA) di AWS Payment Cryptography include un impegno di servizio del 99,99% per tutte le API di crittografia dei pagamenti. Per adempiere a questo impegno, AWS Payment Cryptography garantisce che tutti i dati e le informazioni di autorizzazione necessari per eseguire una richiesta API siano disponibili su tutti gli host regionali che ricevono la richiesta.

L'infrastruttura di crittografia AWS Payment viene replicata in almeno tre zone di disponibilità (AZ) in ogni regione. Per garantire che i guasti di più host non influiscano sulle prestazioni della crittografia di AWS Payment, AWS Payment Cryptography è progettata per servire il traffico dei clienti da qualsiasi AZ di una regione.

Le modifiche apportate alle proprietà o alle autorizzazioni di una chiave di pagamento vengono replicate su tutti gli host della regione per garantire che la richiesta successiva possa essere elaborata correttamente da qualsiasi host della regione. Le richieste di operazioni crittografiche che

utilizzano la chiave di pagamento vengono inoltrate a una flotta di moduli di sicurezza hardware (HSM) di AWS Payment Cryptography, ognuno dei quali può eseguire l'operazione con la chiave di pagamento.

Design multi-tenant

Il design multi-tenant di AWS Payment Cryptography consente di soddisfare lo SLA di disponibilità e di sostenere tassi di richiesta elevati, proteggendo al contempo la riservatezza di chiavi e dati.

Vengono implementati diversi meccanismi di rafforzamento dell'integrità per garantire che la chiave di pagamento specificata per l'operazione crittografica sia sempre quella utilizzata.

Il materiale chiave in testo semplice per le chiavi di crittografia dei pagamenti è ampiamente protetto. Il materiale chiave viene crittografato nell'HSM non appena viene creato e il materiale chiave crittografato viene immediatamente spostato in un archivio sicuro. La chiave crittografata viene recuperata e decrittografata all'interno del modulo HSM solo nel momento in cui viene utilizzata. La chiave in testo normale rimane nella memoria HSM solo per il tempo necessario al completamento dell'operazione di crittografia. Il materiale della chiave in testo normale non lascia mai i moduli HSM e non viene mai scritto su storage persistente.

Per ulteriori informazioni sui meccanismi utilizzati da AWS Payment Cryptography per proteggere le tue chiavi, consulta [Dettagli crittografici di AWS Payment Cryptography](#).

Sicurezza dell'infrastruttura in AWS Payment Cryptography

In quanto servizio gestito, AWS Payment Cryptography è protetto dalle procedure di sicurezza della rete AWS globale descritte nel white paper [Amazon Web Services: Overview of Security Processes](#).

Utilizzi chiamate API AWS pubblicate per accedere AWS Payment Cryptography attraverso la rete. I client devono supportare Transport Layer Security (TLS) 1.2 o versioni successive. I client devono, inoltre, supportare le suite di cifratura con PFS (Perfect Forward Secrecy), ad esempio Ephemeral Diffie-Hellman (DHE) o Elliptic Curve Ephemeral Diffie-Hellman (ECDHE). La maggior parte dei sistemi moderni come Java 7 e versioni successive, supporta tali modalità.

Inoltre, le richieste devono essere firmate tramite un ID chiave di accesso e una chiave di accesso segreta associata a un principal IAM. In alternativa, è possibile utilizzare [AWS Security Token Service](#) (AWS STS) per generare le credenziali di sicurezza temporanee per firmare le richieste.

Isolamento degli host fisici

La sicurezza dell'infrastruttura fisica utilizzata da AWS Payment Cryptography è soggetta ai controlli descritti nella sezione Sicurezza fisica e ambientale di Amazon Web Services: panoramica dei processi di sicurezza. Puoi trovare altri dettagli nei report di conformità e nei risultati degli audit di terze parti elencati nella sezione precedente.

La crittografia di AWS Payment è supportata da moduli di sicurezza hardware (HSM) dedicati elencati in commercial-off-the-shelf PCI PTS HSM. Il materiale chiave per le chiavi di crittografia di AWS Payment viene archiviato solo nella memoria volatile degli HSM e solo mentre la chiave di crittografia dei pagamenti è in uso. Gli HSM si trovano in rack con accesso controllato all'interno dei data center di Amazon che applicano il doppio controllo per qualsiasi accesso fisico. Per informazioni dettagliate sul funzionamento degli HSM di AWS Payment Cryptography, consulta [AWS Payment Cryptography Cryptographic Details](#).

Connessione alla crittografia dei AWS pagamenti tramite un endpoint VPC

Puoi connetterti direttamente alla crittografia dei AWS pagamenti tramite un endpoint di interfaccia privato nel tuo cloud privato virtuale (VPC). Quando utilizzi un endpoint VPC di interfaccia, la comunicazione tra il tuo VPC e la crittografia dei AWS pagamenti viene condotta interamente all'interno della rete. AWS

AWS Payment Cryptography supporta gli endpoint Amazon Virtual Private Cloud (Amazon VPC) con tecnologia [AWS PrivateLink](#). Ogni endpoint VPC è rappresentato da una o più [interfacce di rete elastiche \(ENI\)](#) con indirizzi IP privati nelle sottoreti del VPC.

L'interfaccia VPC endpoint collega il tuo VPC direttamente alla crittografia dei AWS pagamenti senza un gateway Internet, un dispositivo NAT, una connessione VPN o una connessione. AWS Direct Connect Le istanze del tuo VPC non necessitano di indirizzi IP pubblici per comunicare AWS con Payment Cryptography.

Regioni

[AWS Payment Cryptography supporta gli endpoint VPC e le policy degli endpoint VPC Regioni AWS in tutti AWS i paesi in cui è supportata la crittografia dei pagamenti.](#)

Argomenti

- [Considerazioni sugli endpoint AWS VPC per la crittografia dei pagamenti](#)
- [Creazione di un endpoint VPC per la crittografia dei pagamenti AWS](#)
- [Connessione a un endpoint VPC per la crittografia dei AWS pagamenti](#)
- [Controllo dell'accesso all'endpoint VPC](#)
- [Utilizzo di un endpoint VPC in un'istruzione di policy](#)
- [Registrazione dell'endpoint VPC](#)

Considerazioni sugli endpoint AWS VPC per la crittografia dei pagamenti

Prima di configurare un endpoint VPC di interfaccia per la crittografia dei AWS pagamenti, consulta l'argomento [Proprietà e limitazioni dell'endpoint dell'interfaccia](#) nella Guida AWS PrivateLink

AWS Il supporto per la crittografia dei pagamenti per un endpoint VPC include quanto segue.

- [È possibile utilizzare l'endpoint VPC per richiamare tutte le operazioni AWS Payment Cryptography Controlplane e AWS le operazioni Payment Cryptography Dataplane da un VPC.](#)
- È possibile creare un endpoint VPC di interfaccia che si connette a un endpoint della regione di crittografia dei AWS pagamenti.
- AWS La crittografia dei pagamenti consiste in un piano di controllo e un piano dati. È possibile scegliere di configurare uno o entrambi i servizi secondari, ma ciascuno è configurato separatamente.
- Puoi utilizzare AWS CloudTrail i log per verificare l'utilizzo delle chiavi di crittografia dei AWS pagamenti tramite l'endpoint VPC. Per informazioni dettagliate, vedi [Registrazione dell'endpoint VPC](#).

Creazione di un endpoint VPC per la crittografia dei pagamenti AWS

Puoi creare un endpoint VPC per la crittografia dei AWS pagamenti utilizzando la console Amazon VPC o l'API Amazon VPC. Per ulteriori informazioni, consulta la sezione [Creazione di un endpoint di interfaccia](#) nella Guida per l'utente di AWS PrivateLink .

- Per creare un endpoint VPC per la crittografia dei AWS pagamenti, utilizza i seguenti nomi di servizio:

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

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

Ad esempio, nella regione degli Stati Uniti occidentali (Oregon) (us-west-2), i nomi dei servizi sarebbero:

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

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

Per semplificare l'utilizzo dell'endpoint VPC, puoi abilitare un [nome DNS privato](#) per l'endpoint VPC. Se selezioni l'opzione Abilita nome DNS, il nome host DNS standard AWS di crittografia dei pagamenti viene risolto sul tuo endpoint VPC. Ad esempio, `https://controlplane.payment-cryptography.us-west-2.amazonaws.com` si risolverebbe in un endpoint VPC connesso al nome del servizio `com.amazonaws.us-west-2.payment-cryptography.controlplane`.

Questa opzione rende più semplice utilizzare l'endpoint VPC. Per impostazione predefinita, AWS gli SDK AWS CLI utilizzano il nome host DNS standard di crittografia dei AWS pagamenti, quindi non è necessario specificare l'URL dell'endpoint VPC nelle applicazioni e nei comandi.

Per ulteriori informazioni, consulta la sezione [Accesso a un servizio tramite un endpoint di interfaccia](#) nella Guida di AWS PrivateLink .

Connessione a un endpoint VPC per la crittografia dei AWS pagamenti

Puoi connetterti alla crittografia dei AWS pagamenti tramite l'endpoint VPC utilizzando AWS un SDK, o. AWS CLI AWS Tools for PowerShell Per specificare l'endpoint VPC, utilizzare il nome DNS.

Ad esempio, il comando [list-keys](#) utilizza il parametro `endpoint-url` per specificare l'endpoint VPC. Per utilizzare un comando come questo, sostituisci l'ID dell'endpoint VPC con uno presente nel tuo account.

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

Se hai attivato nomi host privati al momento della creazione dell'endpoint VPC, non è necessario specificare l'URL dell'endpoint VPC nella configurazione dell'applicazione o nei comandi della CLI. Il nome host DNS standard AWS di Payment Cryptography viene risolto sul tuo endpoint VPC. Gli AWS CLI e SDK utilizzano questo nome host per impostazione predefinita, quindi puoi iniziare a

utilizzare l'endpoint VPC per connetterti a un endpoint regionale di crittografia dei AWS pagamenti senza modificare nulla negli script e nelle applicazioni.

Per utilizzare nomi host privati, gli attributi `enableDnsHostnames` e `enableDnsSupport` del VPC devono essere impostati su `true`. [Per impostare questi attributi, utilizzate l'operazione `AttributeModifyVpc`](#) Per informazioni dettagliate, consulta la sezione [Visualizzazione e aggiornamento degli attributi DNS per il VPC](#) nella Guida per l'utente di Amazon VPC.

Controllo dell'accesso all'endpoint VPC

Per controllare l'accesso al tuo endpoint VPC per la crittografia dei AWS pagamenti, allega una policy sugli endpoint VPC all'endpoint VPC. La policy degli endpoint determina se i mandanti possono utilizzare l'endpoint VPC per richiamare operazioni di crittografia dei AWS pagamenti con risorse specifiche di crittografia dei pagamenti. AWS

Puoi creare una policy di endpoint VPC quando crei l'endpoint e puoi modificare la policy di endpoint VPC in qualsiasi momento. [Utilizza la console di gestione VPC o le operazioni `EndpointCreateVpcEndpoint`, `EndpointModifyVpc`](#) Puoi anche creare e modificare una policy per gli endpoint VPC [utilizzando](#) un modello. AWS CloudFormation Per informazioni sull'utilizzo della console di gestione VPC, consulta la sezione [Creazione di un endpoint di interfaccia](#) e [Modifica di un endpoint di interfaccia](#) nella Guida di AWS PrivateLink .

Argomenti

- [Informazioni sulle policy di endpoint VPC](#)
- [Policy di endpoint VPC predefinita](#)
- [Creazione di una policy degli endpoint VPC](#)
- [Visualizzazione di una policy di endpoint VPC](#)

Informazioni sulle policy di endpoint VPC

Affinché una richiesta AWS di crittografia dei pagamenti che utilizza un endpoint VPC abbia esito positivo, il principale richiede le autorizzazioni da due fonti:

- Una [politica basata sull'identità](#) deve fornire all'utente principale l'autorizzazione a richiamare l'operazione sulla risorsa (chiavi o alias di crittografia dei pagamenti)AWS .
- Una policy di endpoint VPC deve concedere l'autorizzazione al principale per utilizzare l'endpoint per effettuare la richiesta.

Ad esempio, una politica chiave potrebbe fornire l'autorizzazione principale per chiamare [Decrypt su una particolare chiave di crittografia dei pagamenti](#). AWS Tuttavia, la politica degli endpoint VPC potrebbe non consentire a tale principale di Decrypt richiamare le chiavi di crittografia dei AWS pagamenti utilizzando l'endpoint.

Oppure, una policy sugli endpoint VPC potrebbe consentire a un principale di utilizzare l'endpoint per richiamare l'[StopKeyutilizzo](#) su determinate AWS chiavi di crittografia dei pagamenti. Ma se il preside non dispone delle autorizzazioni previste da una policy IAM, la richiesta fallisce.

Policy di endpoint VPC predefinita

Ogni endpoint VPC dispone di una policy di endpoint VPC, ma non è necessario specificare la policy. Se non specifichi una policy, la policy di endpoint predefinita consente tutte le operazioni effettuate da tutte i principali su tutte le risorse dell'endpoint.

Tuttavia, per le risorse AWS Payment Cryptography, il mandante deve anche avere l'autorizzazione a richiamare l'operazione da una [policy IAM](#). Pertanto, in pratica, la policy predefinita indica che se un principale dispone dell'autorizzazione per chiamare un'operazione su una risorsa, può anche chiamarla utilizzando l'endpoint.

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

Per permettere ai principali di utilizzare l'endpoint VPC solo per un sottoinsieme di operazioni consentite, [crea o modifica la policy di endpoint VPC](#).

Creazione di una policy degli endpoint VPC

Una policy di endpoint VPC determina se un principale dispone dell'autorizzazione per utilizzare l'endpoint VPC per eseguire operazioni su una risorsa. Per quanto riguarda le risorse AWS Payment Cryptography, il committente deve inoltre disporre dell'autorizzazione a eseguire le operazioni in base a una [policy IAM](#).

Ogni istruzione della policy di endpoint VPC richiede i seguenti elementi:

- Il principale che può eseguire operazioni.
- Le azioni che possono essere eseguite
- Le risorse sui cui si possono eseguire le azioni

L'istruzione della policy non specifica l'endpoint VPC. Si applica invece a qualsiasi endpoint VPC a cui è collegata la policy. Per ulteriori informazioni, consulta [Controllo degli accessi ai servizi con endpoint VPC](#) nella Guida per l'utente di Amazon VPC.

Di seguito è riportato un esempio di policy degli endpoint VPC per AWS la crittografia dei pagamenti. Se collegata a un endpoint VPC, questa policy consente di utilizzare l'endpoint VPC `ExampleUser` per richiamare le operazioni specificate sulle chiavi di crittografia dei pagamenti specificate. AWS Prima di utilizzare una politica come questa, sostituisci l'[identificatore principale e chiave](#) di esempio con valori validi del tuo account.

```
{
  "Statement": [
    {
      "Sid": "AllowDecryptAndView",
      "Principal": {"AWS": "arn:aws:iam::111122223333:user/ExampleUser"},
      "Effect": "Allow",
      "Action": [
        "payment-cryptography:Decrypt",
        "payment-cryptography:GetKey",
        "payment-cryptography:ListAliases",
        "payment-cryptography:ListKeys",
        "payment-cryptography:GetAlias"
      ],
      "Resource": "arn:aws:payment-cryptography:us-east-2:111122223333:key/
        kwapwa6qaiFlLw2h"
    }
  ]
}
```

AWS CloudTrail registra tutte le operazioni che utilizzano l'endpoint VPC. Tuttavia, CloudTrail i log non includono le operazioni richieste dai responsabili in altri account o le operazioni relative alle chiavi di crittografia dei AWS pagamenti in altri account.

Pertanto, potresti voler creare una policy sugli endpoint VPC che impedisca ai responsabili degli account esterni di utilizzare l'endpoint VPC per richiamare qualsiasi operazione di crittografia dei AWS pagamenti su qualsiasi chiave dell'account locale.

L'esempio seguente utilizza la chiave [aws: PrincipalAccount](#) global condition per negare l'accesso a tutti i principali per tutte le operazioni su tutte le chiavi di crittografia dei AWS pagamenti a meno che il principale non si trovi nell'account locale. Prima di utilizzare una policy come questa, sostituisci l'ID account dell'esempio con uno valido.

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

Visualizzazione di una policy di endpoint VPC

[Per visualizzare la policy degli endpoint VPC per un endpoint, utilizza la console di gestione VPC o l'operazione Endpoints. DescribeVpc](#)

Il AWS CLI comando seguente ottiene la policy per l'endpoint con l'ID endpoint VPC specificato.

Prima di eseguire questo comando, sostituisci l'ID endpoint dell'esempio con un ID valido del tuo account.

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

Utilizzo di un endpoint VPC in un'istruzione di policy

Puoi controllare l'accesso alle risorse e alle operazioni di crittografia dei AWS pagamenti quando la richiesta proviene da VPC o utilizza un endpoint VPC. [A tale scopo, utilizza una policy IAM](#)

- Usa la chiave di condizione `aws:sourceVpce` per concedere o limitare l'accesso in base all'endpoint VPC.
- Usa la chiave di condizione `aws:sourceVpc` per concedere o limitare l'accesso in base al VPC che ospita l'endpoint privato.

Note

La chiave di `aws:sourceIP` condizione non è efficace quando la richiesta proviene da un [endpoint Amazon VPC](#). Per limitare le richieste a un endpoint VPC, utilizza il comando `aws:sourceVpce` o le chiavi di condizione `aws:sourceVpc`. Per ulteriori informazioni, consulta la sezione [Gestione delle identità e degli accessi per endpoint VPC e servizi endpoint VPC](#) nella Guida di AWS PrivateLink .

Puoi utilizzare queste chiavi di condizione globali per controllare l'accesso alle chiavi AWS di crittografia dei pagamenti, agli alias e a operazioni del genere [CreateKey](#) che non dipendono da alcuna risorsa particolare.

Ad esempio, la seguente politica di chiave di esempio consente a un utente di eseguire particolari operazioni crittografiche con una chiave di crittografia di AWS pagamento solo quando la richiesta utilizza l'endpoint VPC specificato, bloccando l'accesso sia da Internet che dalle connessioni (se configurato). Quando un utente effettua una richiesta a AWS Payment Cryptography, l'ID dell'endpoint VPC nella richiesta viene confrontato con il valore `aws:sourceVpce` della chiave di condizione nella policy. Se non corrisponde, la richiesta viene rifiutata.

Per utilizzare una politica come questa, sostituisci l'ID segnaposto e gli Account AWS ID endpoint VPC con valori validi per il tuo account.

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

```

Puoi anche utilizzare la chiave di `aws:sourceVpce` condizione per limitare l'accesso alle tue chiavi di crittografia dei AWS pagamenti in base al VPC in cui risiede l'endpoint VPC.

La seguente politica chiave di esempio consente i comandi che gestiscono le chiavi di crittografia dei AWS pagamenti solo quando provengono da `vpc-12345678`. Inoltre, consente i comandi che utilizzano le chiavi AWS di crittografia dei pagamenti per operazioni crittografiche solo quando provengono da `vpc-2b2b2b2b`. Puoi usare una policy come questa se un'applicazione è in esecuzione in un VPC, ma devi utilizzare un secondo VPC separato per le funzioni di gestione.

Per utilizzare una politica come questa, sostituisci l'ID segnaposto e gli Account AWS ID endpoint VPC con valori validi per il tuo account.

```

{
  "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": "*"
}
]
}

```

Registrazione dell'endpoint VPC

AWS CloudTrail registra tutte le operazioni che utilizzano l'endpoint VPC. Quando una richiesta di crittografia dei AWS pagamenti utilizza un endpoint VPC, l'ID dell'endpoint VPC viene visualizzato nella voce di registro che registra [AWS CloudTrail la](#) richiesta. Puoi utilizzare l'ID endpoint per verificare l'uso del tuo endpoint VPC AWS Payment Cryptography.

Per proteggere il tuo VPC, le richieste rifiutate da una [policy sugli endpoint VPC](#), ma che altrimenti sarebbero state consentite, non vengono registrate in. [AWS CloudTrail](#)

Ad esempio, questa voce di registro di esempio registra una [GenerateMac](#) richiesta che ha utilizzato l'endpoint VPC. Il campo `vpcEndpointId` viene visualizzato alla fine della voce di log.

```
{
  "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": "-oo28vrvr.controlplane.payment-cryptography.us-
east-1.vpce.amazonaws.com"
}
}
```

Le migliori pratiche di sicurezza per la crittografia dei pagamenti AWS

AWS Payment Cryptography supporta molte funzionalità di sicurezza integrate o che è possibile implementare opzionalmente per migliorare la protezione delle chiavi di crittografia e garantire che vengano utilizzate per lo scopo previsto, tra cui [le politiche IAM](#), un ampio set di chiavi di condizione delle policy per perfezionare le policy chiave e le policy IAM e l'applicazione integrata delle regole PCI PIN relative ai blocchi chiave.

Important

Le linee guida generali fornite non rappresentano una soluzione di sicurezza completa. Poiché non tutte le best practice sono appropriate per tutte le situazioni, non sono prescrittive.

- Utilizzo delle chiavi e modalità d'uso: la crittografia dei AWS pagamenti segue e applica le restrizioni sull'uso delle chiavi e sulla modalità di utilizzo, come descritto nella specifica ANSI X9 TR 31-2018 Interoperable Secure Key Exchange Key Block e conforme al requisito di sicurezza PCI PIN 18-3. Ciò limita la possibilità di utilizzare una singola chiave per più scopi e associa

crittograficamente i metadati della chiave (come le operazioni consentite) al materiale chiave stesso. AWS La crittografia dei pagamenti applica automaticamente queste restrizioni, ad esempio una chiave di crittografia a chiave (TR31_K0_KEY_ENCRYPTION_KEY) non può essere utilizzata anche per la decrittografia dei dati. Per ulteriori dettagli, consulta [Comprensione degli attributi chiave della chiave Payment Cryptography AWS](#).

- Limita la condivisione di materiale a chiave simmetrica: condividi solo materiale a chiave simmetrica (come chiavi di crittografia PIN o chiavi di crittografia a chiave) solo con un'altra entità. Se è necessario trasmettere materiale sensibile a più entità o partner, crea chiavi aggiuntive. AWS La crittografia dei pagamenti non espone mai in chiaro materiale a chiave simmetrica o materiale a chiave privata asimmetrica.
- Utilizza alias o tag per associare le chiavi a determinati casi d'uso o partner: gli alias possono essere utilizzati per indicare facilmente il caso d'uso associato a una chiave come Alias/bin_12345_CVK per indicare una chiave di verifica della carta associata a BIN 12345. Per offrire maggiore flessibilità, prendi in considerazione la creazione di tag come bin=12345, use_case=acquiring, country=us, partner=foo. Gli alias e i tag possono essere utilizzati anche per limitare l'accesso, ad esempio per imporre i controlli di accesso tra l'emissione e l'acquisizione dei casi d'uso.
- Pratica l'accesso con privilegi minimi: IAM può essere usato per limitare l'accesso alla produzione ai sistemi anziché ai singoli utenti, ad esempio vietando ai singoli utenti di creare chiavi o eseguire operazioni crittografiche. IAM può essere utilizzato anche per limitare l'accesso a comandi e chiavi che potrebbero non essere applicabili al caso d'uso, ad esempio per limitare la capacità di generare o convalidare i pin per un acquirente. Un altro modo per utilizzare l'accesso con privilegi minimi consiste nel limitare le operazioni sensibili (come l'importazione di chiavi) a specifici account di servizio. Per esempi, consulta [AWS Esempi di politiche basate sull'identità della crittografia dei pagamenti](#).

Consulta anche

- [Gestione delle identità e degli accessi per la crittografia dei AWS pagamenti](#)
- [Best practice per la sicurezza in IAM](#) nella Guida per l'utente di IAM

Convalida della conformità per AWS Crittografia dei pagamenti

I revisori di terze parti valutano la sicurezza e la conformità di AWS La crittografia dei pagamenti come parte di una procedura multipla AWS programmi di conformità. Questi includono SOC, PCI e altri.

AWS La crittografia dei pagamenti è stata valutata in base a diversi standard PCI oltre al PCI DSS. Questi includono la crittografia PCI PIN Security (PCI PIN) e la crittografia PCI Point-to-Point (P2PE). Si prega di vedere AWS Artifact per gli attestati e le guide di conformità disponibili.

Per un elenco dei servizi AWS coperti da programmi di conformità specifici, consulta [Servizi AWS coperti dal programma di compliance](#). Per informazioni generali, consulta [Programmi per la conformità di AWS](#).

È possibile scaricare i report di audit di terze parti utilizzando AWS Artifact. Per ulteriori informazioni, consulta [Download di report in AWS Artifact](#).

La tua responsabilità in materia di conformità durante l'utilizzo AWS La crittografia dei pagamenti è determinata dalla sensibilità dei tuoi dati, dagli obiettivi di conformità della tua azienda e dalle leggi e dai regolamenti applicabili. AWS fornisce le seguenti risorse per contribuire alla conformità:

- [Guide di avvio rapido su sicurezza e conformità](#)—Queste guide all'implementazione illustrano le considerazioni architettoniche e forniscono i passaggi per l'implementazione di ambienti di base incentrati sulla sicurezza e la conformità su AWS.
- [AWS Risorse per la conformità](#)—Questa raccolta di cartelle di lavoro e guide potrebbe essere applicabile al tuo settore e alla tua posizione.
- [Valutazione delle risorse con regole](#) nel AWS Config Guida per gli sviluppatori—AWS Config; valuta la conformità delle configurazioni delle risorse alle pratiche interne, alle linee guida del settore e alle normative.
- [AWS Security Hub](#)—Questo AWS il servizio fornisce una visione completa dello stato di sicurezza all'interno AWS che ti aiuta a verificare la tua conformità agli standard e alle migliori pratiche del settore della sicurezza.

Gestione delle identità e degli accessi per la crittografia dei AWS pagamenti

AWS Identity and Access Management (IAM) è uno strumento Servizio AWS che aiuta un amministratore a controllare in modo sicuro l'accesso alle risorse. AWS Gli amministratori IAM controllano chi può essere autenticato (effettuato l'accesso) e autorizzato (disporre delle autorizzazioni) a utilizzare AWS le risorse Payment Cryptography. IAM è uno strumento Servizio AWS che puoi utilizzare senza costi aggiuntivi.

Argomenti

- [Destinatari](#)
- [Autenticazione con identità](#)
- [Gestione dell'accesso con policy](#)
- [Come funziona la crittografia dei AWS pagamenti con IAM](#)
- [AWS Esempi di politiche basate sull'identità della crittografia dei pagamenti](#)
- [Risoluzione dei problemi relativi alla crittografia dei AWS pagamenti, all'identità e all'accesso](#)

Destinatari

Il modo in cui utilizzi AWS Identity and Access Management (IAM) varia a seconda del lavoro svolto nella crittografia dei AWS pagamenti.

Utente del servizio: se utilizzi il servizio AWS Payment Cryptography per svolgere il tuo lavoro, l'amministratore ti fornisce le credenziali e le autorizzazioni necessarie. Poiché utilizzi più funzionalità AWS di crittografia dei pagamenti per svolgere il tuo lavoro, potresti aver bisogno di autorizzazioni aggiuntive. La comprensione della gestione dell'accesso ti consente di richiedere le autorizzazioni corrette all'amministratore. Se non riesci ad accedere a una funzionalità di AWS Payment Cryptography, consulta [Risoluzione dei problemi relativi alla crittografia dei AWS pagamenti, all'identità e all'accesso](#)

Amministratore del servizio: se sei responsabile delle risorse di crittografia dei AWS pagamenti presso la tua azienda, probabilmente hai pieno accesso a AWS Payment Cryptography. È tuo compito determinare a quali funzionalità e risorse AWS di Payment Cryptography devono accedere gli utenti del servizio. Devi inviare le richieste all'amministratore IAM per cambiare le autorizzazioni

degli utenti del servizio. Esamina le informazioni contenute in questa pagina per comprendere i concetti di base relativi a IAM. Per saperne di più su come la tua azienda può utilizzare IAM con AWS Payment Cryptography, consulta [Come funziona la crittografia dei AWS pagamenti con IAM](#)

Amministratore IAM: se sei un amministratore IAM, potresti voler conoscere i dettagli su come scrivere politiche per gestire l'accesso alla crittografia dei AWS pagamenti. Per visualizzare esempi AWS di policy basate sull'identità di Payment Cryptography che puoi utilizzare in IAM, consulta [AWS Esempi di politiche basate sull'identità della crittografia dei pagamenti](#)

Autenticazione con identità

L'autenticazione è il modo in cui accedi AWS utilizzando le tue credenziali di identità. Devi essere autenticato (aver effettuato l' Utente root dell'account AWS accesso AWS) come utente IAM o assumendo un ruolo IAM.

Puoi accedere AWS come identità federata utilizzando le credenziali fornite tramite una fonte di identità. AWS IAM Identity Center Gli utenti (IAM Identity Center), l'autenticazione Single Sign-On della tua azienda e le tue credenziali di Google o Facebook sono esempi di identità federate. Se accedi come identità federata, l'amministratore ha configurato in precedenza la federazione delle identità utilizzando i ruoli IAM. Quando accedi AWS utilizzando la federazione, assumi indirettamente un ruolo.

A seconda del tipo di utente, puoi accedere al AWS Management Console o al portale di AWS accesso. Per ulteriori informazioni sull'accesso a AWS, vedi [Come accedere al tuo Account AWS nella Guida per l'Accedi ad AWS utente](#).

Se accedi a AWS livello di codice, AWS fornisce un kit di sviluppo software (SDK) e un'interfaccia a riga di comando (CLI) per firmare crittograficamente le tue richieste utilizzando le tue credenziali. Se non utilizzi AWS strumenti, devi firmare tu stesso le richieste. Per ulteriori informazioni sull'utilizzo del metodo consigliato per firmare autonomamente le richieste, consulta [Signing AWS API request](#) nella IAM User Guide.

A prescindere dal metodo di autenticazione utilizzato, potrebbe essere necessario specificare ulteriori informazioni sulla sicurezza. Ad esempio, ti AWS consiglia di utilizzare l'autenticazione a più fattori (MFA) per aumentare la sicurezza del tuo account. Per ulteriori informazioni, consulta [Autenticazione a più fattori](#) nella Guida per l'utente di AWS IAM Identity Center e [Utilizzo dell'autenticazione a più fattori \(MFA\) in AWS](#) nella Guida per l'utente IAM.

Account AWS utente root

Quando si crea un account Account AWS, si inizia con un'identità di accesso che ha accesso completo a tutte Servizi AWS le risorse dell'account. Questa identità è denominata utente Account AWS root ed è accessibile effettuando l'accesso con l'indirizzo e-mail e la password utilizzati per creare l'account. Si consiglia vivamente di non utilizzare l'utente root per le attività quotidiane. Conserva le credenziali dell'utente root e utilizzale per eseguire le operazioni che solo l'utente root può eseguire. Per un elenco completo delle attività che richiedono l'accesso come utente root, consulta la sezione [Attività che richiedono le credenziali dell'utente root](#) nella Guida per l'utente IAM.

Utenti e gruppi IAM

Un [utente IAM](#) è un'identità interna Account AWS che dispone di autorizzazioni specifiche per una singola persona o applicazione. Ove possibile, consigliamo di fare affidamento a credenziali temporanee invece di creare utenti IAM con credenziali a lungo termine come le password e le chiavi di accesso. Tuttavia, se si hanno casi d'uso specifici che richiedono credenziali a lungo termine con utenti IAM, si consiglia di ruotare le chiavi di accesso. Per ulteriori informazioni, consulta la pagina [Rotazione periodica delle chiavi di accesso per casi d'uso che richiedono credenziali a lungo termine](#) nella Guida per l'utente IAM.

Un [gruppo IAM](#) è un'identità che specifica un insieme di utenti IAM. Non è possibile eseguire l'accesso come gruppo. È possibile utilizzare gruppi per specificare le autorizzazioni per più utenti alla volta. I gruppi semplificano la gestione delle autorizzazioni per set di utenti di grandi dimensioni. Ad esempio, è possibile avere un gruppo denominato IAMAdmins e concedere a tale gruppo le autorizzazioni per amministrare le risorse IAM.

Gli utenti sono diversi dai ruoli. Un utente è associato in modo univoco a una persona o un'applicazione, mentre un ruolo è destinato a essere assunto da chiunque ne abbia bisogno. Gli utenti dispongono di credenziali a lungo termine permanenti, mentre i ruoli forniscono credenziali temporanee. Per ulteriori informazioni, consulta [Quando creare un utente IAM \(invece di un ruolo\)](#) nella Guida per l'utente IAM.

Ruoli IAM

Un [ruolo IAM](#) è un'identità interna all'utente Account AWS che dispone di autorizzazioni specifiche. È simile a un utente IAM, ma non è associato a una persona specifica. Puoi assumere temporaneamente un ruolo IAM in AWS Management Console [cambiando ruolo](#). Puoi assumere un ruolo chiamando un'operazione AWS CLI o AWS API o utilizzando un URL personalizzato. Per

ulteriori informazioni sui metodi per l'utilizzo dei ruoli, consulta [Utilizzo di ruoli IAM](#) nella Guida per l'utente IAM.

I ruoli IAM con credenziali temporanee sono utili nelle seguenti situazioni:

- **Accesso utente federato:** per assegnare le autorizzazioni a una identità federata, è possibile creare un ruolo e definire le autorizzazioni per il ruolo. Quando un'identità federata viene autenticata, l'identità viene associata al ruolo e ottiene le autorizzazioni da esso definite. Per ulteriori informazioni sulla federazione dei ruoli, consulta [Creazione di un ruolo per un provider di identità di terza parte](#) nella Guida per l'utente IAM. Se utilizzi IAM Identity Center, configura un set di autorizzazioni. IAM Identity Center mette in correlazione il set di autorizzazioni con un ruolo in IAM per controllare a cosa possono accedere le identità dopo l'autenticazione. Per informazioni sui set di autorizzazioni, consulta [Set di autorizzazioni](#) nella Guida per l'utente di AWS IAM Identity Center .
- **Autorizzazioni utente IAM temporanee:** un utente IAM o un ruolo può assumere un ruolo IAM per ottenere temporaneamente autorizzazioni diverse per un'attività specifica.
- **Accesso multi-account:** è possibile utilizzare un ruolo IAM per permettere a un utente (un principale affidabile) con un account diverso di accedere alle risorse nell'account. I ruoli sono lo strumento principale per concedere l'accesso multi-account. Tuttavia, con alcuni Servizi AWS, è possibile allegare una policy direttamente a una risorsa (anziché utilizzare un ruolo come proxy). Per informazioni sulle differenze tra ruoli e policy basate su risorse per l'accesso multi-account, consulta [Differenza tra i ruoli IAM e le policy basate su risorse](#) nella Guida per l'utente IAM.
- **Accesso a più servizi:** alcuni Servizi AWS utilizzano le funzionalità di altri Servizi AWS. Ad esempio, quando effettui una chiamata in un servizio, è comune che tale servizio esegua applicazioni in Amazon EC2 o archivi oggetti in Amazon S3. Un servizio può eseguire questa operazione utilizzando le autorizzazioni dell'entità chiamante, utilizzando un ruolo di servizio o utilizzando un ruolo collegato al servizio.
- **Sessioni di accesso diretto (FAS):** quando utilizzi un utente o un ruolo IAM per eseguire azioni AWS, sei considerato un preside. Quando si utilizzano alcuni servizi, è possibile eseguire un'operazione che attiva un'altra operazione in un servizio diverso. FAS utilizza le autorizzazioni del principale che chiama un Servizio AWS, combinate con la richiesta Servizio AWS per effettuare richieste ai servizi downstream. Le richieste FAS vengono effettuate solo quando un servizio riceve una richiesta che richiede interazioni con altri Servizi AWS o risorse per essere completata. In questo caso è necessario disporre delle autorizzazioni per eseguire entrambe le azioni. Per i dettagli delle policy relative alle richieste FAS, consulta la pagina [Forward access sessions](#).

- **Ruolo di servizio:** un ruolo di servizio è un [ruolo IAM](#) che un servizio assume per eseguire azioni per tuo conto. Un amministratore IAM può creare, modificare ed eliminare un ruolo di servizio dall'interno di IAM. Per ulteriori informazioni, consulta la sezione [Creazione di un ruolo per delegare le autorizzazioni a un Servizio AWS](#) nella Guida per l'utente IAM.
- **Ruolo collegato al servizio:** un ruolo collegato al servizio è un tipo di ruolo di servizio collegato a un Servizio AWS. Il servizio può assumere il ruolo per eseguire un'azione per tuo conto. I ruoli collegati al servizio vengono visualizzati nel tuo account Account AWS e sono di proprietà del servizio. Un amministratore IAM può visualizzare le autorizzazioni per i ruoli collegati ai servizi, ma non modificarle.
- **Applicazioni in esecuzione su Amazon EC2:** puoi utilizzare un ruolo IAM per gestire le credenziali temporanee per le applicazioni in esecuzione su un'istanza EC2 e che AWS CLI effettuano richieste API. AWS. Ciò è preferibile all'archiviazione delle chiavi di accesso nell'istanza EC2. Per assegnare un ruolo AWS a un'istanza EC2 e renderlo disponibile per tutte le sue applicazioni, crei un profilo di istanza collegato all'istanza. Un profilo dell'istanza contiene il ruolo e consente ai programmi in esecuzione sull'istanza EC2 di ottenere le credenziali temporanee. Per ulteriori informazioni, consulta [Utilizzo di un ruolo IAM per concedere autorizzazioni ad applicazioni in esecuzione su istanze di Amazon EC2](#) nella Guida per l'utente IAM.

Per informazioni sull'utilizzo dei ruoli IAM, consulta [Quando creare un ruolo IAM \(invece di un utente\)](#) nella Guida per l'utente IAM.

Gestione dell'accesso con policy

Puoi controllare l'accesso AWS creando policy e collegandole a AWS identità o risorse. Una policy è un oggetto AWS che, se associato a un'identità o a una risorsa, ne definisce le autorizzazioni. AWS valuta queste politiche quando un principale (utente, utente root o sessione di ruolo) effettua una richiesta. Le autorizzazioni nelle policy determinano l'approvazione o il rifiuto della richiesta. La maggior parte delle politiche viene archiviata AWS come documenti JSON. Per ulteriori informazioni sulla struttura e sui contenuti dei documenti delle policy JSON, consulta [Panoramica delle policy JSON](#) nella Guida per l'utente IAM.

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. In altre parole, quale principale può eseguire azioni su quali risorse e in quali condizioni.

Per impostazione predefinita, utenti e ruoli non dispongono di autorizzazioni. Per concedere agli utenti l'autorizzazione a eseguire operazioni sulle risorse di cui hanno bisogno, un amministratore

IAM può creare policy IAM. L'amministratore può quindi aggiungere le policy IAM ai ruoli e gli utenti possono assumere i ruoli.

Le policy IAM definiscono le autorizzazioni relative a un'operazione, a prescindere dal metodo utilizzato per eseguirla. Ad esempio, supponiamo di disporre di una policy che consente l'operazione `iam:GetRole`. Un utente con tale policy può ottenere informazioni sul ruolo dall' AWS Management Console AWS CLI, dall' AWS API.

Policy basate su identità

Le policy basate su identità sono documenti di policy di autorizzazione JSON che è possibile allegare a un'identità (utente, gruppo di utenti o ruolo IAM). Tali policy definiscono le azioni che utenti e ruoli possono eseguire, su quali risorse e in quali condizioni. Per informazioni su come creare una policy basata su identità, consulta [Creazione di policy IAM](#) nella Guida per l'utente IAM.

Le policy basate su identità possono essere ulteriormente classificate come policy inline o policy gestite. Le policy inline sono integrate direttamente in un singolo utente, gruppo o ruolo. Le politiche gestite sono politiche autonome che puoi allegare a più utenti, gruppi e ruoli nel tuo Account AWS. Le politiche gestite includono politiche AWS gestite e politiche gestite dai clienti. Per informazioni su come scegliere tra una policy gestita o una policy inline, consulta [Scelta fra policy gestite e policy inline](#) nella Guida per l'utente IAM.

Policy basate su risorse

Le policy basate su risorse sono documenti di policy JSON che è possibile collegare a una risorsa. Gli esempi più comuni di policy basate su risorse sono le policy di attendibilità dei ruoli IAM e le policy dei bucket Amazon S3. Nei servizi che supportano policy basate sulle risorse, gli amministratori dei servizi possono utilizzarli per controllare l'accesso a una risorsa specifica. Quando è collegata a una risorsa, una policy definisce le azioni che un principale può eseguire su tale risorsa e a quali condizioni. È necessario [specificare un principale](#) in una policy basata sulle risorse. I principali possono includere account, utenti, ruoli, utenti federati o Servizi AWS.

Le policy basate sulle risorse sono policy inline che si trovano in tale servizio. Non puoi utilizzare le policy AWS gestite di IAM in una policy basata sulle risorse.

Liste di controllo degli accessi (ACL)

Le liste di controllo degli accessi (ACL) controllano quali principali (membri, utenti o ruoli dell'account) hanno le autorizzazioni per accedere a una risorsa. Le ACL sono simili alle policy basate su risorse, sebbene non utilizzino il formato del documento di policy JSON.

Amazon S3 e Amazon VPC sono esempi di servizi che supportano gli ACL. AWS WAF Per maggiori informazioni sulle ACL, consulta [Panoramica delle liste di controllo degli accessi \(ACL\)](#) nella Guida per gli sviluppatori di Amazon Simple Storage Service.

Altri tipi di policy

AWS supporta tipi di policy aggiuntivi e meno comuni. Questi tipi di policy possono impostare il numero massimo di autorizzazioni concesse dai tipi di policy più comuni.

- **Limiti delle autorizzazioni:** un limite delle autorizzazioni è una funzionalità avanzata nella quale si imposta il numero massimo di autorizzazioni che una policy basata su identità può concedere a un'entità IAM (utente o ruolo IAM). È possibile impostare un limite delle autorizzazioni per un'entità. Le autorizzazioni risultanti sono l'intersezione delle policy basate su identità dell'entità e i relativi limiti delle autorizzazioni. Le policy basate su risorse che specificano l'utente o il ruolo nel campo `Principal` sono condizionate dal limite delle autorizzazioni. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni sui limiti delle autorizzazioni, consulta [Limiti delle autorizzazioni per le entità IAM](#) nella Guida per l'utente IAM.
- **Politiche di controllo dei servizi (SCP):** le SCP sono politiche JSON che specificano le autorizzazioni massime per un'organizzazione o un'unità organizzativa (OU) in AWS Organizations. AWS Organizations è un servizio per il raggruppamento e la gestione centralizzata di più Account AWS di proprietà dell'azienda. Se abiliti tutte le funzionalità in un'organizzazione, puoi applicare le policy di controllo dei servizi (SCP) a uno o tutti i tuoi account. L'SCP limita le autorizzazioni per le entità negli account dei membri, inclusa ciascuna. Utente root dell'account AWS Per ulteriori informazioni su organizzazioni e policy SCP, consulta la pagina sulle [Policy di controllo dei servizi](#) nella Guida per l'utente di AWS Organizations .
- **Policy di sessione:** le policy di sessione sono policy avanzate che vengono trasmesse come parametro quando si crea in modo programmatico una sessione temporanea per un ruolo o un utente federato. Le autorizzazioni della sessione risultante sono l'intersezione delle policy basate su identità del ruolo o dell'utente e le policy di sessione. Le autorizzazioni possono anche provenire da una policy basata su risorse. Un rifiuto esplicito in una qualsiasi di queste policy sostituisce l'autorizzazione. Per ulteriori informazioni, consulta [Policy di sessione](#) nella Guida per l'utente IAM.

Più tipi di policy

Quando più tipi di policy si applicano a una richiesta, le autorizzazioni risultanti sono più complicate da comprendere. Per scoprire come si AWS determina se consentire una richiesta quando sono coinvolti più tipi di policy, consulta [Logica di valutazione delle policy](#) nella IAM User Guide.

Come funziona la crittografia dei AWS pagamenti con IAM

Prima di utilizzare IAM per gestire l'accesso alla crittografia dei AWS pagamenti, è necessario comprendere quali funzionalità IAM sono disponibili per l'uso con AWS Payment Cryptography. Per avere una visione di alto livello di come AWS Payment Cryptography e altri AWS servizi funzionano con IAM, consulta [AWS Services That Work with IAM nella IAM](#) User Guide.

Argomenti

- [AWS Politiche basate sull'identità della crittografia dei pagamenti](#)
- [Autorizzazione basata sui tag di crittografia dei pagamenti AWS](#)

AWS Politiche basate sull'identità della crittografia dei pagamenti

Con le policy basate su identità di IAM, è possibile specificare quali azioni e risorse sono consentite o rifiutate, nonché le condizioni in base alle quali le azioni sono consentite o rifiutate. AWS La crittografia dei pagamenti supporta azioni, risorse e chiavi di condizione specifiche. Per informazioni su tutti gli elementi utilizzati in una policy JSON, consulta [Documentazione di riferimento degli elementi delle policy JSON IAM](#) nella Guida per l'utente IAM.

Azioni

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire azioni su quali risorse, e in quali condizioni.

L'elemento `Action` di una policy JSON descrive le azioni che è possibile utilizzare per consentire o negare l'accesso a un criterio. Le azioni politiche in genere hanno lo stesso nome dell'operazione AWS API associata. Ci sono alcune eccezioni, ad esempio le azioni di sola autorizzazione che non hanno un'operazione API corrispondente. Esistono anche alcune operazioni che richiedono più operazioni in una policy. Queste operazioni aggiuntive sono denominate operazioni dipendenti.

Includi le operazioni in una policy per concedere le autorizzazioni a eseguire l'operazione associata.

Le azioni politiche in AWS Payment Cryptography utilizzano il seguente prefisso prima dell'azione: `payment-cryptography:` Ad esempio, per concedere a qualcuno il permesso di eseguire un'operazione dell'`VerifyCardDataAPI` AWS Payment Cryptography, includi `payment-cryptography:VerifyCardData` azione nella sua politica. Le istruzioni della policy devono includere un elemento `Action` o `NotAction`. AWS Payment Cryptography definisce una propria serie di azioni che descrivono le attività che è possibile eseguire con questo servizio.

Per specificare più azioni in una sola istruzione, separa ciascuna di esse con una virgola come mostrato di seguito:

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

È possibile specificare più azioni tramite caratteri jolly (*). Ad esempio, per specificare tutte le azioni che iniziano con la parola `List` (come `ListKeys` e `ListAliases`), includi l'azione seguente:

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

Per visualizzare un elenco di azioni di crittografia dei AWS pagamenti, consulta [Azioni definite dalla crittografia dei AWS pagamenti](#) nella Guida per l'utente IAM.

Risorse

Gli amministratori possono utilizzare le policy AWS JSON per specificare chi ha accesso a cosa. Cioè, quale principale può eseguire operazioni su quali risorse, e in quali condizioni.

L'elemento JSON `Resource` della policy specifica l'oggetto o gli oggetti ai quali si applica l'operazione. Le istruzioni devono includere un elemento `Resource` o un elemento `NotResource`. Come best practice, specifica una risorsa utilizzando il suo [nome della risorsa Amazon \(ARN\)](#). Puoi eseguire questa operazione per azioni che supportano un tipo di risorsa specifico, note come autorizzazioni a livello di risorsa.

Per le azioni che non supportano le autorizzazioni a livello di risorsa, ad esempio le operazioni di elenco, utilizza un carattere jolly (*) per indicare che l'istruzione si applica a tutte le risorse.

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

La risorsa chiave di crittografia dei pagamenti ha il seguente ARN:

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

Per ulteriori informazioni sul formato degli ARN, consulta [Amazon Resource Names \(ARNs\) e AWS Service Namespaces](#).

Ad esempio, per specificare l'istanza `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaif11w2h` nell'istruzione, utilizza il seguente ARN:

```
"Resource": "arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaif11w2h"
```

Per specificare tutte le chiavi che appartengono a un account specifico, usa il carattere jolly (*):

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

Alcune azioni AWS di crittografia dei pagamenti, come quelle per la creazione di chiavi, non possono essere eseguite su una risorsa specifica. In questi casi, è necessario utilizzare il carattere jolly (*).

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

Per specificare più risorse in una singola istruzione, utilizzate una virgola come illustrato di seguito:

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

Esempi

Per visualizzare esempi di politiche basate sull'identità della crittografia dei AWS pagamenti, consulta [AWS Esempi di politiche basate sull'identità della crittografia dei pagamenti](#)

Autorizzazione basata sui tag di crittografia dei pagamenti AWS

AWS Esempi di politiche basate sull'identità della crittografia dei pagamenti

Per impostazione predefinita, gli utenti e i ruoli IAM non sono autorizzati a creare o modificare AWS risorse di crittografia dei pagamenti. Inoltre, non possono eseguire attività utilizzando l' AWS API AWS Management Console AWS CLI, o. Un amministratore IAM deve creare policy IAM che concedono a utenti e ruoli l'autorizzazione per eseguire operazioni API specifiche sulle risorse specificate di cui hanno bisogno. L'amministratore deve quindi allegare queste policy a utenti o IAM che richiedono tali autorizzazioni.

Per informazioni su come creare una policy basata su identità IAM utilizzando questi documenti di policy JSON di esempio, consulta [Creazione di policy nella scheda JSON](#) nella Guida per l'utente IAM.

Argomenti

- [Best practice delle policy](#)
- [Utilizzo della console Payment Cryptography AWS](#)
- [Consentire agli utenti di visualizzare le loro autorizzazioni](#)
- [Capacità di accedere a tutti gli aspetti della crittografia dei pagamenti AWS](#)
- [Possibilità di chiamare le API utilizzando chiavi specificate](#)
- [Capacità di negare specificamente una risorsa](#)

Best practice delle policy

Le politiche basate sull'identità determinano se qualcuno può creare, accedere o eliminare le risorse di crittografia dei AWS pagamenti nel tuo account. Queste azioni possono comportare costi aggiuntivi per l' Account AWS. Quando crei o modifichi policy basate su identità, segui queste linee guida e raccomandazioni:

- Inizia con le policy AWS gestite e passa alle autorizzazioni con privilegi minimi: per iniziare a concedere autorizzazioni a utenti e carichi di lavoro, utilizza le politiche gestite che concedono le autorizzazioni per molti casi d'uso comuni. AWS Sono disponibili nel tuo Account AWS Ti consigliamo di ridurre ulteriormente le autorizzazioni definendo politiche gestite dai AWS clienti specifiche per i tuoi casi d'uso. Per ulteriori informazioni, consulta [Policy gestite da AWS](#) o [Policy gestite da AWS per le funzioni dei processi](#) nella Guida per l'utente IAM.
- Applica le autorizzazioni con privilegio minimo: quando imposti le autorizzazioni con le policy IAM, concedi solo le autorizzazioni richieste per eseguire un'attività. Puoi farlo definendo le azioni che possono essere intraprese su risorse specifiche in condizioni specifiche, note anche come autorizzazioni con privilegi minimi. Per ulteriori informazioni sull'utilizzo di IAM per applicare le autorizzazioni, consulta [Policy e autorizzazioni in IAM](#) nella Guida per l'utente IAM.
- Condizioni d'uso nelle policy IAM per limitare ulteriormente l'accesso: per limitare l'accesso a operazioni e risorse puoi aggiungere una condizione alle tue policy. Ad esempio, è possibile scrivere una condizione di policy per specificare che tutte le richieste devono essere inviate utilizzando SSL. Puoi anche utilizzare le condizioni per concedere l'accesso alle azioni del servizio se vengono utilizzate tramite uno specifico Servizio AWS, ad esempio AWS CloudFormation. Per

ulteriori informazioni, consulta la sezione [Elementi delle policy JSON di IAM: condizione](#) nella Guida per l'utente IAM.

- Utilizzo di IAM Access Analyzer per convalidare le policy IAM e garantire autorizzazioni sicure e funzionali: IAM Access Analyzer convalida le policy nuove ed esistenti in modo che aderiscano alla sintassi della policy IAM (JSON) e alle best practice di IAM. IAM Access Analyzer offre oltre 100 controlli delle policy e consigli utili per creare policy sicure e funzionali. Per ulteriori informazioni, consulta [Convalida delle policy per IAM Access Analyzer](#) nella Guida per l'utente IAM.
- Richiedi l'autenticazione a più fattori (MFA): se hai uno scenario che richiede utenti IAM o un utente root nel Account AWS tuo, attiva l'MFA per una maggiore sicurezza. Per richiedere la MFA quando vengono chiamate le operazioni API, aggiungi le condizioni MFA alle policy. Per ulteriori informazioni, consulta [Configurazione dell'accesso alle API protetto con MFA](#) nella Guida per l'utente IAM.

Per maggiori informazioni sulle best practice in IAM, consulta [Best practice di sicurezza in IAM](#) nella Guida per l'utente di IAM.

Utilizzo della console Payment Cryptography AWS

Per accedere alla console AWS Payment Cryptography, devi disporre di un set minimo di autorizzazioni. Queste autorizzazioni devono consentirti di elencare e visualizzare i dettagli sulle risorse di crittografia dei AWS pagamenti presenti nel tuo account. AWS Se crei una policy basata su identità più restrittiva rispetto alle autorizzazioni minime richieste, la console non funzionerà nel modo previsto per le entità (utenti e ruoli IAM) associate a tale policy.

Per garantire che tali entità possano ancora utilizzare la console AWS di crittografia dei pagamenti, allega anche la seguente politica AWS gestita alle entità. Per ulteriori informazioni, consulta [Aggiunta di autorizzazioni a un utente](#) nella Guida per l'utente IAM.

Non è necessario consentire autorizzazioni minime per la console per gli utenti che effettuano chiamate solo verso AWS CLI o l' AWS API. Al contrario, puoi accedere solo alle operazioni che soddisfano l'operazione API che stai cercando di eseguire.

Consentire agli utenti di visualizzare le loro autorizzazioni

Questo esempio mostra in che modo è possibile creare una policy che consente agli utenti IAM di visualizzare le policy inline e gestite che sono collegate alla relativa identità utente. Questa politica include le autorizzazioni per completare questa azione sulla console o utilizzando l'API o a livello di codice. AWS CLI 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": "*"
    }
  ]
}
```

Capacità di accedere a tutti gli aspetti della crittografia dei pagamenti AWS

Warning

Questo esempio fornisce autorizzazioni ampie e non è consigliato. Considerate invece i modelli di accesso meno privilegiati.

In questo esempio, vuoi concedere a un utente IAM del tuo AWS account l'accesso a tutte le tue chiavi di crittografia dei AWS pagamenti e la possibilità di chiamare tutte le API di crittografia dei AWS pagamenti, comprese entrambe le operazioni. ControlPlane DataPlane

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

Possibilità di chiamare le API utilizzando chiavi specificate

In questo esempio, vuoi concedere a un utente IAM del tuo AWS account l'accesso a una delle tue chiavi di crittografia dei AWS pagamenti, quindi utilizzare questa risorsa in due API `arn:aws:payment-cryptography:us-east-2:111122223333:key/kwapwa6qaiif1lw2h` e `GenerateCardData` `VerifyCardData`. Al contrario, l'utente IAM non avrà accesso all'uso di questa chiave per altre operazioni come `DeleteKey` `ExportKey`.

Le risorse possono essere chiavi, con prefisso `key` o `alias`, con prefisso. `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"
  ]
}
]
}

```

Capacità di negare specificamente una risorsa

Warning

Valuta attentamente le implicazioni della concessione dell'accesso con caratteri jolly. Considerate invece un modello con privilegi minimi.

In questo esempio, desideri consentire a un utente IAM del tuo AWS account di accedere a una qualsiasi delle tue chiavi di crittografia dei AWS pagamenti, ma desideri negare le autorizzazioni a una chiave specifica. L'utente avrà accesso a `VerifyCardData` e `GenerateCardData` con tutte le chiavi ad eccezione di quella specificata nella dichiarazione di negazione.

```

{
  "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/  
    kwapwa6qaiif1lw2h"  
  ]  
}  
]  
}
```

Risoluzione dei problemi relativi alla crittografia dei AWS pagamenti, all'identità e all'accesso

Gli argomenti verranno aggiunti a questa sezione man mano che verranno identificati i problemi relativi all'IAM specifici della crittografia dei AWS pagamenti. Per contenuti generali sulla risoluzione dei problemi sugli argomenti IAM, consulta la [sezione sulla risoluzione dei problemi della Guida](#) per l'utente IAM.

Monitoraggio della crittografia AWS dei pagamenti

Il monitoraggio è importante per garantire affidabilità, disponibilità e prestazioni di AWS Payment Cryptography e delle altre soluzioni AWS. AWS fornisce i seguenti strumenti di monitoraggio per tenere sotto controllo la crittografia dei AWS pagamenti, segnalare un problema e intervenire automaticamente quando necessario:

- Amazon CloudWatch monitora le AWS risorse e le applicazioni che esegui su AWS in tempo reale. Puoi raccogliere i parametri e tenerne traccia, creare pannelli di controllo personalizzati e impostare allarmi per inviare una notifica o intraprendere azioni quando un parametro specificato raggiunge una determinata soglia. Ad esempio, puoi impostare perché CloudWatch tenga traccia dell'uso della CPU o di altri parametri delle tue istanze Amazon EC2 e avviare automaticamente nuove istanze quando necessario. Per ulteriori informazioni, consulta la Guida per l'utente di [Amazon CloudWatch User Guide](#).
- Amazon CloudWatch Logs consente di monitorare, archiviare e accedere ai file di log dalle istanze Amazon EC2 e da altre CloudTrail origini. CloudWatch log sono in grado di monitorare le informazioni nei file di log e notificare quando vengono raggiunte determinate soglie. Puoi inoltre archiviare i dati del log in storage estremamente durevole. Per ulteriori informazioni, consulta la [Guida per l'utente di Amazon CloudWatch Logs](#).
- Amazon EventBridge può essere utilizzato per automatizzare i AWS servizi e rispondere automaticamente a eventi di sistema, come i problemi relativi alla disponibilità delle applicazioni o le modifiche delle risorse. Gli eventi dei servizi AWS vengono recapitati a EventBridge quasi in tempo reale. Puoi compilare regole semplici che indichino quali eventi sono considerati di interesse per te e quali operazioni automatizzate intraprendere quando un evento corrisponde a una regola. Per ulteriori informazioni, consulta la Guida per [EventBridge l'utente di Amazon User Guide](#).
- AWS CloudTrail acquisisce le chiamate API e gli eventi correlati effettuati da o per conto del tuo account AWS e fornisce i file di log a un bucket Amazon S3 specificato. Puoi identificare quali utenti e account hanno richiamato AWS, l'indirizzo IP di origine da cui sono state effettuate le chiamate e quando sono avvenute. Per ulteriori informazioni, consultare la [Guida per l'utente AWS CloudTrail](#).

Note

AWS CloudTraili registri sono supportati per le operazioni del piano di controllo, ad esempio, CreateKey ma non per le operazioni sul piano dati, come la generazione dei dati delle schede

Registrazione delle AWS chiamate API di crittografia dei pagamenti utilizzando AWS CloudTrail

AWSLa crittografia dei pagamenti è integrata conAWS CloudTrail, un servizio che offre un record delle operazioni eseguite da un utente, da ruolo o da un AWS servizio nella AWS Crittografia dei pagamenti. CloudTrailacquisil tutte le chiamate AWS API come eventi. Le chiamate acquisite includono le chiamate dalla console AWS di crittografia dei pagamenti e le chiamate di codice alle operazioni API di Crittografia dei AWS pagamenti. Se viene creato un trail, è possibile abilitare la distribuzione continua di CloudTrail eventi in un bucket Amazon S3, inclusi gli eventi per la crittografia dei AWS pagamenti. Se non configuri un trail, è comunque possibile visualizzare gli eventi più recenti nella console di CloudTrail in Event history (Cronologia eventi). Le informazioni raccolte daCloudTrail, consentono di determinare la richiesta effettuata adAWS, l'indirizzo IP da cui è partita la richiesta, l'autore della richiesta, il momento in cui è stata eseguita e altri dettagli.

Per ulteriori informazioni su CloudTrail, consulta la [Guida per l'utente di AWS CloudTrail](#).

Note

L'integrazione con Cloudtrail è attualmente supportata solo per le operazioni sui piani di controllo.

AWSInformazioni sulla crittografia dei pagamenti in CloudTrail

CloudTrail è abilitato sull'account AWS al momento della sua creazione. Quando si verifica un'attività in Crittografia dei AWS pagamenti, questa viene registrata in un CloudTrail evento insieme ad altri eventi di AWS servizio nella Cronologia degli eventi. È possibile visualizzare, cercare e scaricare gli eventi recenti nell'account AWS. Per ulteriori informazioni, consulta [Visualizzazione di eventi mediante la cronologia eventi di CloudTrail](#).

Per una registrazione continua degli eventi nell'AWSaccount, inclusi gli eventi relativi alla crittografia dei pagamenti, crea un trail. Un trail consente di CloudTrail distribuire i file di log in un bucket Amazon S3. Per impostazione predefinita, quando si crea un trail nella console, il trail sarà valido in tutte le regioni AWS. Il trail registra gli eventi di tutte le Regioni nella partizione AWS e distribuisce i file di log nel bucket Amazon S3 specificato. Inoltre, puoi configurare altri servizi AWS per analizzare con maggiore dettaglio e usare i dati raccolti nei log CloudTrail. Per ulteriori informazioni, consulta gli argomenti seguenti:

- [Panoramica della creazione di un percorso](#)
- [Servizi e integrazioni CloudTrail supportati](#)
- [Configurazione delle notifiche Amazon SNS per CloudTrail](#)
- [Ricezione di file di log CloudTrail da più regioni](#)
- [Ricezione di file di log CloudTrail da più account](#)

CloudTrailregistra le operazioni AWS di crittografia dei pagamenti, come,,, [CreateKeyImportKeyDeleteKeyListKeys](#), e tutte le altre operazioni del [TagResource](#)piano di controllo.

Ogni evento o voce di log contiene informazioni sull'utente che ha generato la richiesta. Le informazioni di identità consentono di determinare quanto segue:

- Se la richiesta è stata effettuata con credenziali utente root o AWS Identity and Access Management (IAM).
- Se la richiesta è stata effettuata con le credenziali di sicurezza temporanee per un ruolo o un utente federato.
- Se la richiesta è stata effettuata da un altro servizio AWS.

Per ulteriori informazioni, consulta [Elemento CloudTrail userIdentity](#).

Informazioni sulle voci dei AWS file di registro di Payment Cryptography

Un percorso è una configurazione che consente la distribuzione di eventi come i file di log in un bucket Amazon S3 specificato. I file di log di CloudTrail contengono una o più voci di log. Un evento rappresenta una singola richiesta da un'fonte e include informazioni sul operazione richiesta, data e ora dell'operazione, parametri richiesti e così via. I file di log di CloudTrail non sono una traccia stack ordinata delle chiamate pubbliche dell'API, quindi non vengono visualizzati in un ordine specifico.

L'esempio seguente mostra una voce di CloudTrail registro che illustra l'operazione Crittografia CreateKey dei AWS pagamenti.

```
{
  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: {

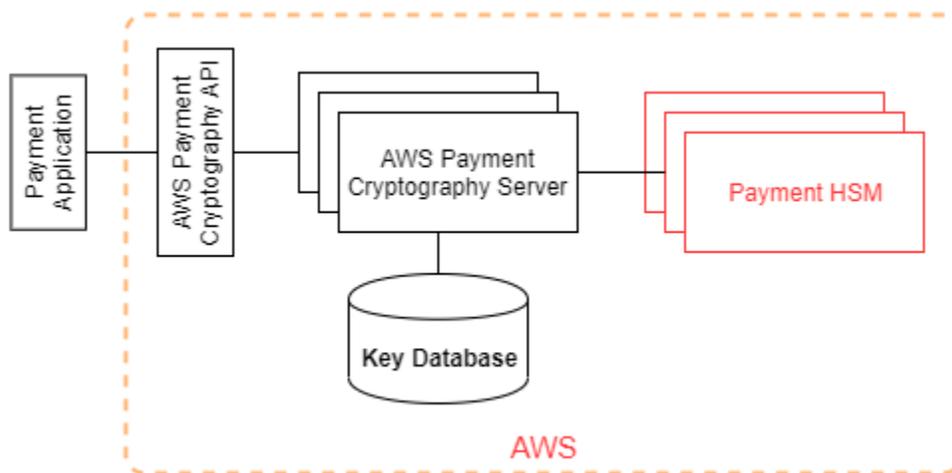
```

```
        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={  
}  
}  
}
```

Dettagli crittografici

AWS Payment Cryptography fornisce un'interfaccia web per generare e gestire chiavi crittografiche per le transazioni di pagamento. AWS Payment Cryptography offre servizi standard di gestione delle chiavi e crittografia delle transazioni di pagamento e strumenti che è possibile utilizzare per la gestione e il controllo centralizzati. Questa documentazione fornisce una descrizione dettagliata delle operazioni di crittografia AWS di crittografia di protezione hardware.

[AWS Payment Cryptography](#) contiene più interfacce (inclusa un'API RESTful, tramite AWS CLI, AWS SDK e altro AWS Management Console) per richiedere le operazioni crittografiche di una flotta distribuita di moduli di sicurezza hardware convalidati da PCI PTS HSM.



AWS Payment Cryptography è un servizio su più livelli costituito da host rivolti Web rivolti Web rivolti al Web rivolti AWS al Web. Il raggruppamento di host a più livelli forma lo stack di crittografia AWS di protezione hardware. Tutte le richieste a AWS Payment Cryptography devono essere effettuate tramite il protocollo TLS (Transport Layer Security) di protezione hardware. AWS Gli host di servizi consentono il protocollo TLS solo con una suite di crittografia che garantisce una [perfetta](#) segretezza di inoltro. Il servizio autentica e autorizza le tue richieste utilizzando gli stessi meccanismi di credenziali e policy di IAM disponibili per tutte le altre operazioni API. AWS

AWS I server di crittografia dei pagamenti si connettono all'[HSM](#) sottostante tramite una rete privata non virtuale. Le connessioni tra i componenti del servizio e [HSM](#) sono protette con TLS reciproco (mTLS) per l'autenticazione e la crittografia.

Obiettivi di progettazione di

AWS La crittografia di pagamento è progettata per soddisfare i seguenti requisiti:

- **Affidabile:** l'utilizzo delle chiavi è protetto alle policy di controllo accessi definite e gestite dall'utente. Non esiste alcun meccanismo per esportare le chiavi di crittografia AWS di pagamento in chiaro. La riservatezza delle chiavi di crittografia è fondamentale. Per eseguire azioni amministrative sui moduli di protezione hardware sono necessari più dipendenti Amazon con accesso specifico ai controlli di accesso basati su quorum. Nessun dipendente Amazon ha accesso alle chiavi principali (o master) o ai backup HSM. Le chiavi principali non possono essere sincronizzate con gli HSM che non fanno parte di un'area di crittografia dei AWS pagamenti. Tutte le altre chiavi sono protette da chiavi principali HSM. Pertanto, le chiavi AWS di crittografia dei pagamenti del cliente non sono utilizzabili al di fuori del servizio di crittografia dei AWS pagamenti che opera all'interno dell'account del cliente.
- **Bassa latenza e velocità effettiva elevata:** AWS Payment Cryptography fornisce operazioni crittografiche a livello di latenza e throughput adatte alla gestione delle chiavi crittografiche di pagamento e all'elaborazione delle transazioni di pagamento.
- **Durabilità:** la durabilità delle chiavi di crittografia è progettata per eguagliare quella dei servizi di durabilità più elevati in AWS. Una singola chiave crittografica può essere condivisa con un terminale di pagamento, una carta con chip EMV o un altro dispositivo crittografico sicuro (SCD) in uso da molti anni.
- **regioni indipendenti:** AWS offre regioni indipendenti per i clienti che devono limitare l'accesso ai dati in regioni diverse o devono soddisfare i requisiti di durabilità dei dati. L'utilizzo delle chiavi può essere isolato all'interno di una regione AWS.
- **Fonte sicura di numeri casuali:** Poiché la crittografia forte dipende dalla generazione di numeri casuali davvero imprevedibili, la crittografia di numeri casuali davvero imprevedibili, AWS fornisce una fonte convalidata e a qualità elevata di numeri casuali. Tutta la generazione di chiavi per la crittografia dei AWS pagamenti utilizza HSM certificato PCI PTS HSM, che opera in modalità PCI.
- **Audit:** la crittografia dei AWS pagamenti registra l'uso e la gestione delle chiavi crittografiche nei registri e nei CloudTrail registri dei servizi disponibili tramite Amazon. CloudWatch È possibile utilizzare CloudTrail i log per ispezionare l'utilizzo delle chiavi di crittografia, nonché l'uso delle chiavi da parte degli account con cui le chiavi sono condivise. AWS La crittografia dei pagamenti viene verificata da valutatori terzi rispetto agli standard PCI, al marchio della carta e agli standard di sicurezza dei pagamenti regionali applicabili. Gli attestati e le guide sulla responsabilità condivisa sono disponibili su AWS Artifact.

- **Elastico:** la crittografia dei AWS pagamenti si espande e aumenta in base alla tua richiesta. Invece di prevedere e riservare la capacità HSM, AWS Payment Cryptography fornisce la crittografia dei pagamenti su richiesta. AWS Payment Cryptography si assume la responsabilità di mantenere la sicurezza e la conformità di HSM per fornire una capacità sufficiente a soddisfare i picchi di domanda dei clienti.

Fondazioni

Gli argomenti di questo capitolo descrivono le primitive crittografiche della crittografia dei AWS pagamenti e dove vengono utilizzate. Inoltre introducono gli elementi di base del servizio.

Argomenti

- [Primitive di crittografia](#)
- [Entropia e generazione di numeri casuali](#)
- [Operazioni chiave simmetriche](#)
- [Operazioni con chiavi asimmetriche](#)
- [Archiviazione delle chiavi](#)
- [Importazione di chiavi tramite chiavi simmetriche](#)
- [Importazione di chiavi tramite chiavi asimmetriche](#)
- [Esportazione di chiavi](#)
- [Protocollo DUKPT \(Derived Unique Key Per Transaction\)](#)
- [Gerarchia delle chiavi](#)

Primitive di crittografia

AWS La crittografia dei pagamenti utilizza algoritmi crittografici standard parametrizzabili in modo che le applicazioni possano implementare gli algoritmi necessari per il loro caso d'uso. L'insieme di algoritmi crittografici è definito dagli standard PCI, ANSI X9, EMVCo e ISO. Tutta la crittografia viene eseguita da moduli HSM PCI PTS HSM elencati negli standard in esecuzione in modalità PCI.

Entropia e generazione di numeri casuali

AWS La generazione di chiavi di crittografia dei pagamenti viene eseguita sugli HSM di crittografia dei pagamenti. AWS Gli HSM implementano un generatore di numeri casuali che soddisfa i requisiti PCI PTS HSM per tutti i tipi e i parametri di chiave supportati.

Operazioni chiave simmetriche

Sono supportati gli algoritmi a chiave simmetrica e i punti di forza chiave definiti in ANSI X9 TR 31, ANSI X9.24 e PCI PIN Annex C:

- Funzioni hash: algoritmi della famiglia SHA2 e SHA3 con dimensioni di output superiori a 2551. Fatta eccezione per la retrocompatibilità con i terminali POI PTS v3 pre-PCI.
- Crittografia e decrittografia: AES con dimensione della chiave maggiore o uguale a 128 bit o TDEA con dimensione delle chiavi maggiore o uguale a 112 bit (2 o 3 chiavi).
- Codici di autenticazione dei messaggi (MAC) CMAC o GMAC con AES, nonché HMAC con una funzione hash approvata e una dimensione della chiave maggiore o uguale a 128.

AWS Payment Cryptography utilizza AES 256 per le chiavi principali HSM, le chiavi di protezione dei dati e le chiavi di sessione TLS.

Operazioni con chiavi asimmetriche

Sono supportati gli algoritmi a chiave asimmetrica e i punti di forza chiave definiti in ANSI X9 TR 31, ANSI X9.24 e PCI PIN Annex C:

- Schemi di definizione delle chiavi approvati, come descritto in NIST SP800-56A (accordo chiave basato su ECC/FCC2), NIST SP800-56B (accordo chiave basato su IFC) e NIST SP800-38F (crittografia/avvolgimento delle chiavi basata su AES).

AWS Gli host di crittografia dei pagamenti consentono solo connessioni al servizio tramite [TLS con una suite di crittografia che](#) offre una perfetta segretezza di inoltro.

Archiviazione delle chiavi

AWS Le chiavi di crittografia dei pagamenti sono protette dalle chiavi principali HSM AES 256 e archiviate in blocchi di chiavi ANSI X9 TR 31 in un database crittografato. Il database viene replicato in un database in memoria sui server Payment Cryptography. AWS

Secondo l'allegato C della normativa sulla sicurezza dei PCI PIN, le chiavi AES 256 sono altrettanto potenti o più potenti di:

- TDEA a 3 tasti

- RSA a 15360 bit
- ECC a 512 bit
- DSA, DH e MQV 15360/512

Importazione di chiavi tramite chiavi simmetriche

AWS La crittografia dei pagamenti supporta l'importazione di crittogrammi e blocchi di chiavi con chiavi simmetriche o pubbliche con una chiave di crittografia a chiave simmetrica (KEK) che è altrettanto potente o più potente della chiave protetta per l'importazione.

Importazione di chiavi tramite chiavi asimmetriche

AWS La crittografia dei pagamenti supporta l'importazione di crittogrammi e blocchi di chiavi con chiavi simmetriche o pubbliche protette da una chiave di crittografia a chiave privata (KEK) che è altrettanto potente o più potente della chiave protetta per l'importazione. L'autenticità e l'integrità della chiave pubblica fornita per la decrittografia devono essere garantite da un certificato rilasciato da un'autorità di fiducia del cliente.

Le KEK pubbliche fornite da AWS Payment Cryptography hanno l'autenticazione e la protezione dell'integrità di un'autorità di certificazione (CA) con conformità attestata a PCI PIN Security e PCI P2PE Annex A.

Esportazione di chiavi

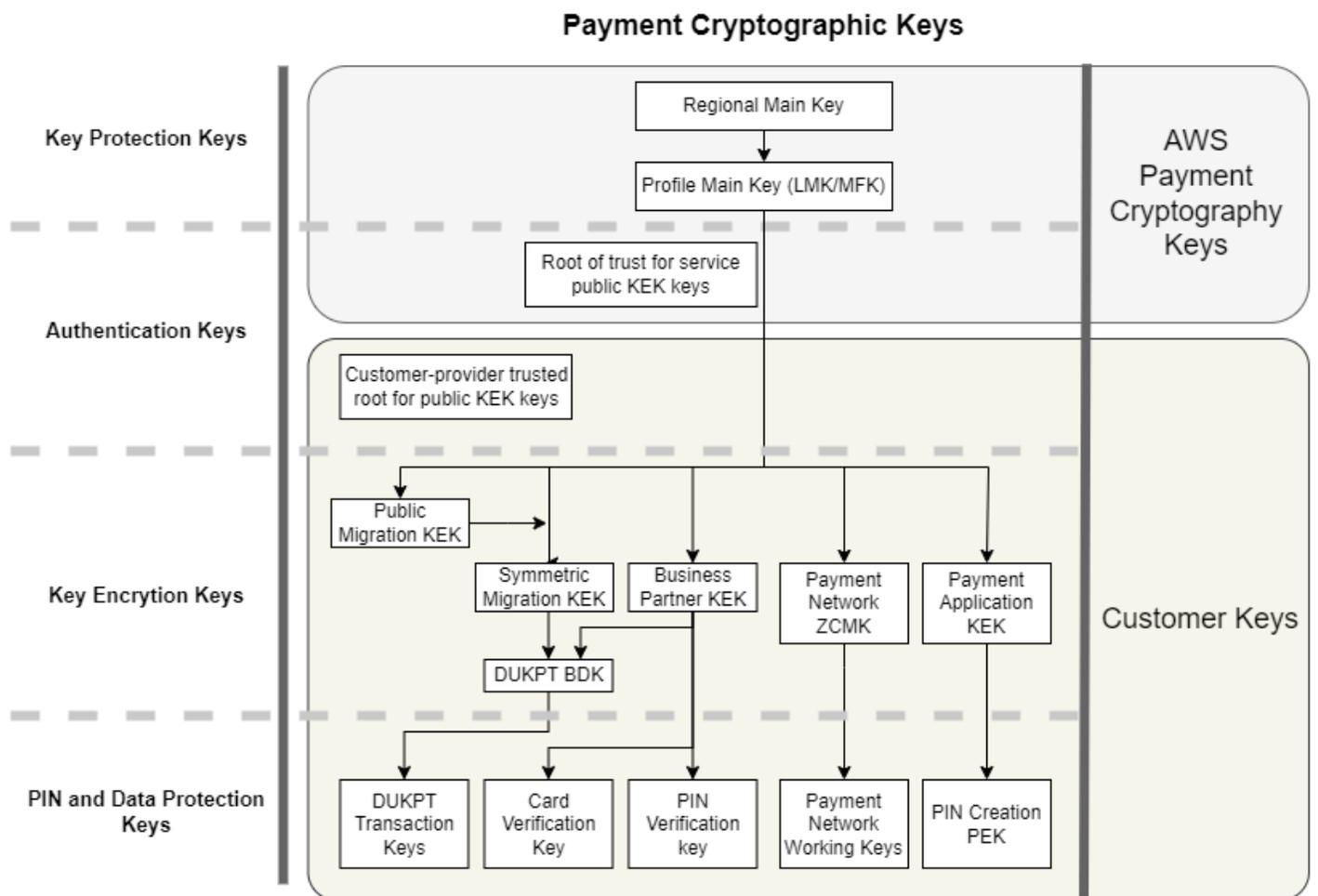
Le chiavi possono essere esportate e protette con chiavi appropriate KeyUsage e che siano altrettanto forti o più forti della chiave da esportare.

Protocollo DUKPT (Derived Unique Key Per Transaction)

AWS La crittografia dei pagamenti supporta le chiavi di derivazione di base (BDK) TDEA e AES come descritto da ANSI X9.24-3.

Gerarchia delle chiavi

La gerarchia delle chiavi di AWS Payment Cryptography garantisce che le chiavi siano sempre protette da chiavi altrettanto potenti o più potenti delle chiavi che proteggono.



AWS Le chiavi di crittografia dei pagamenti vengono utilizzate per la protezione delle chiavi all'interno del servizio:

Chiave	Descrizione
Chiave principale regionale	Protegge le immagini o i profili HSM virtuali utilizzati per l'elaborazione crittografica. Questa chiave esiste solo nei backup HSM e sicuri.
Chiave principale del profilo	Chiave di protezione delle chiavi del cliente di alto livello, tradizionalmente chiamata Local Master Key (LMK) o Master File Key (MFK) per le chiavi del cliente. Questa chiave esiste solo nei backup HSM e sicuri. I profili definiscono configurazioni HSM distinte, come richiesto

Chiave	Descrizione
	dagli standard di sicurezza per i casi d'uso dei pagamenti.
Radice di fiducia per le chiavi di crittografia a chiave pubblica (KEK) di AWS Payment Cryptography	La chiave pubblica principale e il certificato affidabili per l'autenticazione e la convalida delle chiavi pubbliche forniti da AWS Payment Cryptography per l'importazione e l'esportazione di chiavi utilizzando chiavi asimmetriche.

Le chiavi del cliente sono raggruppate in base alle chiavi utilizzate per proteggere altre chiavi e chiavi che proteggono i dati relativi ai pagamenti. Questi sono esempi di chiavi cliente di entrambi i tipi:

Chiave	Descrizione
Root affidabile fornito dal cliente per le chiavi KEK pubbliche	Chiave pubblica e certificato forniti dall'utente come base di fiducia per l'autenticazione e la convalida delle chiavi pubbliche fornite per l'importazione e l'esportazione delle chiavi utilizzando chiavi asimmetriche.
Chiavi di crittografia a chiave (KEK)	Le KEK vengono utilizzate esclusivamente per crittografare altre chiavi per lo scambio tra archivi di chiavi esterni e AWS Payment Cryptography, partner commerciali, reti di pagamento o diverse applicazioni all'interno dell'organizzazione.
Chiave di derivazione di base Derived Unique Key Per Transaction (DUKPT) (BDK)	I BDK vengono utilizzati per creare chiavi univoche per ogni terminale di pagamento e tradurre le transazioni da più terminali in un'unica chiave di funzionamento della banca o dell'acquirente. La best practice, richiesta dalla crittografia PCI Point-to-Point Encryption (P2PE), prevede l'utilizzo di BDK diversi per diversi modelli di terminali, servizi di iniezione o

Chiave	Descrizione
	inizializzazione di chiavi o altra segmentazione per limitare l'impatto della compromissione di un BDK.
Chiave principale per il controllo della zona di rete di pagamento (ZCMK)	Le ZCMK, note anche come chiavi di zona o chiavi master di zona, vengono fornite dalle reti di pagamento per stabilire le chiavi di lavoro iniziali.
chiavi di transazione DUKPT	I terminali di pagamento configurati per DUKPT derivano una chiave unica per il terminale e la transazione. L'HSM che riceve la transazione può determinare la chiave dall'identificatore del terminale e dal numero di sequenza della transazione.
Chiavi per la preparazione dei dati delle carte	Le chiavi master dell'emittente EMV, le chiavi e i valori di verifica delle carte EMV e le chiavi di protezione dei file di dati per la personalizzazione delle carte vengono utilizzate per creare dati per singole carte utilizzabili da un fornitore di servizi di personalizzazione delle carte. Queste chiavi e i dati di convalida crittografica vengono utilizzati anche dalle banche emittenti, o dagli emittenti, per autenticare i dati delle carte nell'ambito dell'autorizzazione delle transazioni.

Chiave	Descrizione
Chiavi per la preparazione dei dati delle carte	Le chiavi master dell'emittente EMV, le chiavi e i valori di verifica delle carte EMV e le chiavi di protezione dei file di dati per la personalizzazione delle carte vengono utilizzate per creare dati per singole carte utilizzabili da un fornitore di servizi di personalizzazione delle carte. Queste chiavi e i dati di convalida crittografica vengono utilizzati anche dalle banche emittenti, o dagli emittenti, per autenticare i dati delle carte nell'ambito dell'autorizzazione delle transazioni.
Chiavi funzionanti della rete di pagamento	Spesso denominate chiave di lavoro dell'emittente o chiave di lavoro dell'acquirente, sono le chiavi che crittografano le transazioni inviate o ricevute dalle reti di pagamento. Queste chiavi vengono ruotate frequentemente dalla rete, spesso ogni giorno o ogni ora. Si tratta di chiavi di crittografia PIN (PEK) per transazioni PIN/debito.
Chiavi di crittografia PIN (Personal Identification Number) (PEK)	Le applicazioni che creano o decrittografano blocchi PIN utilizzano il formato PEK per impedire l'archiviazione o la trasmissione di PIN in testo non crittografato.

Operazioni interne

Questo argomento descrive i requisiti interni implementati dal servizio per proteggere le chiavi dei clienti e le operazioni crittografiche per un servizio di crittografia dei pagamenti e gestione delle chiavi distribuito a livello globale e scalabile.

Specifiche e ciclo di vita HSM

AWS Payment Cryptography utilizza una flotta di HSM disponibili in commercio. Gli HSM sono convalidati secondo lo standard FIPS 140-2 di livello 3 e utilizzano anche versioni firmware e la politica di sicurezza elencate nell'elenco dei dispositivi PCI PTS [approvato dal PCI Security Standards Council come conforme allo standard PCI HSM v3](#). Lo standard PCI PTS HSM include requisiti aggiuntivi per la produzione, la spedizione, l'implementazione, la gestione e la distruzione dell'hardware HSM, importanti per la sicurezza e la conformità dei pagamenti ma non soddisfatti dal FIPS 140.

Tutti gli HSM sono gestiti in modalità PCI e configurati con la politica di sicurezza PCI PTS HSM. Sono abilitate solo le funzioni necessarie per supportare i casi d'uso della crittografia dei AWS pagamenti. AWS Payment Cryptography non prevede la stampa, la visualizzazione o la restituzione di PIN in testo non crittografato.

Sicurezza fisica dei dispositivi HSM

Solo gli HSM con chiavi del dispositivo firmate da un'autorità di certificazione (CA) di crittografia dei AWS pagamenti (CA) del produttore prima della consegna possono essere utilizzati dal servizio. La AWS Payment Cryptography è una CA secondaria della CA del produttore che è alla base della fiducia per i certificati dei produttori e dei dispositivi HSM. La CA del produttore implementa ANSI TR 34 e ha attestato la conformità all'allegato A sulla sicurezza del PCI PIN e all'allegato A. Il produttore verifica che tutti gli HSM con chiavi del dispositivo firmate da AWS Payment Cryptography CA vengano spediti al destinatario designato di AWS.

Come richiesto da PCI PIN Security, il produttore fornisce un elenco di numeri seriali tramite un canale di comunicazione diverso da quello di spedizione HSM. Questi numeri di serie vengono controllati in ogni fase del processo di installazione di HSM nei data center AWS. Infine, gli operatori AWS di Payment Cryptography convalidano l'elenco degli HSM installati confrontandolo con l'elenco del produttore prima di aggiungere il numero di serie all'elenco degli HSM autorizzati a ricevere le chiavi di crittografia dei pagamenti. AWS

Gli HSM sono archiviati in modo sicuro o sotto doppio controllo in qualsiasi momento, il che include:

- Spedizione dal produttore a un impianto di assemblaggio su rack AWS.
- Durante l'assemblaggio del rack.
- Spedizione dall'impianto di assemblaggio del rack a un data center.

- Ricezione e installazione in una sala di elaborazione sicura del data center. I rack HSM prevedono un doppio controllo con serrature con accesso tramite scheda, sensori allarmati sulle porte e telecamere.
- Durante le operazioni.
- Durante lo smantellamento e la distruzione.

Per ogni HSM viene mantenuto e monitorato un sistema completo chain-of-custody, con responsabilità individuale.

Inizializzazione HSM

Un HSM viene inizializzato come parte della flotta AWS Payment Cryptography solo dopo che la sua identità e integrità sono state convalidate mediante numeri di serie, chiavi del dispositivo installate dal produttore e checksum del firmware. Dopo la convalida dell'autenticità e dell'integrità di un HSM, questo viene configurato, inclusa l'attivazione della modalità PCI. Quindi vengono stabilite le chiavi principali della regione AWS Payment Cryptography e le chiavi principali del profilo e l'HSM è disponibile per il servizio.

Assistenza e riparazione HSM

HSM dispone di componenti riparabili che non richiedono la violazione del limite crittografico del dispositivo. Questi componenti includono ventole di raffreddamento, alimentatori e batterie. Se un HSM o un altro dispositivo all'interno del rack HSM necessita di assistenza, il doppio controllo viene mantenuto per tutto il periodo di apertura del rack.

Disattivazione HSM

La disattivazione avviene a causa end-of-life o a un guasto di un HSM. Gli HSM vengono logicamente azzerati prima di essere rimossi dal rack, se funzionanti, e poi distrutti all'interno delle sale di elaborazione sicure dei data center AWS. Non vengono mai restituiti al produttore per la riparazione, utilizzati per altri scopi o altrimenti rimossi da una sala di elaborazione sicura prima della distruzione.

Aggiornamento del firmware HSM

Gli aggiornamenti del firmware HSM vengono applicati quando necessario per mantenere l'allineamento con le versioni elencate di PCI PTS HSM e FIPS 140-2 (o FIPS 140-3), se un aggiornamento è legato alla sicurezza o se si stabilisce che i clienti possono beneficiare delle

funzionalità di una nuova versione. AWS Gli HSM di crittografia dei pagamenti eseguono il firmware, corrispondente alle versioni PCI PTS elencate in HSM. off-the-shelf L'integrità delle nuove versioni del firmware viene convalidata con le versioni del firmware certificate PCI o FIPS, quindi testate per verificarne la funzionalità prima dell'implementazione su tutti gli HSM.

Accesso dell'operatore

Gli operatori possono accedere a HSM senza console per la risoluzione dei problemi nei rari casi in cui le informazioni raccolte da HSM durante le normali operazioni non siano sufficienti per identificare un problema o pianificare una modifica. Vengono eseguiti i seguenti passaggi:

- Le attività di risoluzione dei problemi vengono sviluppate e approvate e viene pianificata la sessione non basata sulla console.
- Un HSM viene rimosso dal servizio di elaborazione clienti.
- Le chiavi principali vengono eliminate, sotto doppio controllo.
- All'operatore è consentito l'accesso all'HSM senza console per eseguire attività di risoluzione dei problemi approvate, sotto doppio controllo.
 - Dopo il termine della sessione non console, viene eseguito il processo di provisioning iniziale sull'HSM, restituendo il firmware e la configurazione standard, quindi sincronizzando la chiave principale, prima di restituire l'HSM ai clienti.
 - I record della sessione vengono registrati nel monitoraggio delle modifiche.
 - Le informazioni ottenute dalla sessione vengono utilizzate per pianificare le modifiche future.

Tutti i record di accesso non relativi alla console vengono esaminati per verificarne la conformità dei processi e le potenziali modifiche al monitoraggio HSM, al processo di non-console-access gestione o alla formazione degli operatori.

Gestione delle chiavi

Tutti gli HSM di una regione sono sincronizzati con una chiave principale regionale. Una chiave principale regionale protegge almeno una chiave principale del profilo. Una Profile Main Key protegge le chiavi del cliente.

Tutte le chiavi principali sono generate da un HSM e distribuite mediante distribuzione simmetrica delle chiavi utilizzando tecniche asimmetriche, in linea con ANSI X9 TR 34 e PCI PIN Annex A.

Argomenti

- [Generazione](#)
- [Sincronizzazione delle chiavi principali della regione](#)
- [Rotazione dei tasti principali della regione](#)
- [Sincronizzazione delle chiavi principali del profilo](#)
- [Rotazione della chiave principale del profilo](#)
- [Protezione](#)
- [Durabilità](#)
- [Sicurezza delle comunicazioni](#)
- [Gestione delle chiavi dei clienti](#)
- [Registrazione di log e monitoraggio](#)

Generazione

Le chiavi principali AES a 256 bit vengono generate su uno degli HSM predisposti per il parco HSM del servizio, utilizzando il generatore di numeri casuali PCI PTS HSM.

Sincronizzazione delle chiavi principali della regione

Le chiavi principali della regione HSM sono sincronizzate dal servizio su tutta la flotta regionale con meccanismi definiti da ANSI X9 TR-34, che includono:

- Autenticazione reciproca tramite chiavi e certificati KDH (Key Distribution Host) e Key Receiver Device (KRD) per garantire l'autenticazione e l'integrità delle chiavi pubbliche.
- I certificati sono firmati da un'autorità di certificazione (CA) che soddisfa i requisiti del PIN PCI allegato A2, ad eccezione degli algoritmi asimmetrici e dei punti di forza chiave appropriati per proteggere le chiavi AES a 256 bit.
- Identificazione e protezione delle chiavi per le chiavi simmetriche distribuite in conformità con ANSI X9 TR-34 e PCI PIN Annex A1, ad eccezione degli algoritmi asimmetrici e dei punti di forza chiave appropriati per proteggere le chiavi AES a 256 bit.

Le chiavi principali della regione vengono stabilite per gli HSM che sono stati autenticati e forniti per una regione da:

- Una chiave principale viene generata su un HSM della regione. Tale HSM è designato come host di distribuzione delle chiavi.

- Tutti gli HSM forniti nella regione generano un token di autenticazione KRD, che contiene la chiave pubblica dell'HSM e informazioni di autenticazione non rigiocabili.
- I token KRD vengono aggiunti all'elenco di autorizzazioni KDH dopo che il KDH ha convalidato l'identità e l'autorizzazione dell'HSM a ricevere le chiavi.
- Il KDH produce un token di chiave principale autenticabile per ogni HSM. I token contengono informazioni di autenticazione KDH e una chiave principale crittografata che può essere caricata solo su un HSM per cui sono stati creati.
- A ogni HSM viene inviato il token chiave principale creato appositamente. Dopo aver convalidato le informazioni di autenticazione proprie dell'HSM e le informazioni di autenticazione KDH, la chiave principale viene decrittografata dalla chiave privata KRD e caricata nella chiave principale.

Nel caso in cui un singolo HSM debba essere risincronizzato con una regione:

- Viene riconvalidato e dotato di firmware e configurazione.
- Se è nuovo nella regione:
 - L'HSM genera un token di autenticazione KRD.
 - Il KDH aggiunge il token all'elenco degli elementi consentiti.
 - Il KDH genera un token chiave principale per l'HSM.
 - L'HSM carica la chiave principale.
 - L'HSM viene messo a disposizione del servizio.

Ciò garantisce che:

- Solo gli HSM convalidati per l'elaborazione della crittografia dei AWS pagamenti all'interno di una regione possono ricevere la chiave principale di quella regione.
- Solo una chiave master di un AWS Payment Cryptography HSM può essere distribuita a un HSM del parco macchine.

Rotazione dei tasti principali della regione

Le chiavi principali della regione vengono ruotate alla scadenza del periodo crittografico, nell'improbabile eventualità che si sospetti una compromissione della chiave o dopo modifiche al servizio che si ritiene abbiano un impatto sulla sicurezza della chiave.

Una nuova chiave principale della regione viene generata e distribuita come durante il provisioning iniziale. Le chiavi principali del profilo salvate devono essere tradotte nella nuova chiave principale della regione.

La rotazione delle chiavi principali della regione non influisce sull'elaborazione dei clienti.

Sincronizzazione delle chiavi principali del profilo

Le chiavi principali del profilo sono protette dalle chiavi principali della regione. Ciò limita un profilo a una regione specifica.

Le chiavi principali del profilo vengono fornite di conseguenza:

- Una chiave principale del profilo viene generata su un HSM con la chiave principale della regione sincronizzata.
- La chiave principale del profilo viene archiviata e crittografata con la configurazione del profilo e altri contesti.
- Il profilo viene utilizzato per le funzioni crittografiche del cliente da qualsiasi HSM della regione con la chiave principale della regione.

Rotazione della chiave principale del profilo

Le chiavi principali del profilo vengono ruotate alla scadenza del periodo crittografico, dopo una sospetta compromissione della chiave o dopo modifiche al servizio che si ritiene abbiano un impatto sulla sicurezza della chiave.

Fasi di rotazione:

- Una nuova chiave principale del profilo viene generata e distribuita come chiave principale in sospenso, come nel caso del provisioning iniziale.
- Un processo in background traduce il materiale chiave del cliente dalla chiave principale del profilo stabilito alla chiave principale in sospenso.
- Quando tutte le chiavi del cliente sono state crittografate con la chiave in sospenso, la chiave in sospenso viene promossa alla chiave principale del profilo.
- Un processo in background elimina il materiale chiave del cliente protetto dalla chiave scaduta.

La rotazione delle chiavi principali del profilo non influisce sull'elaborazione dei clienti.

Protezione

Le chiavi dipendono solo dalla gerarchia delle chiavi per la protezione. La protezione delle chiavi principali è fondamentale per prevenire la perdita o la compromissione di tutte le chiavi del cliente.

Le chiavi principali della regione sono ripristinabili dal backup solo su sistemi HSM autenticati e forniti per il servizio. Queste chiavi possono essere archiviate solo come token di chiave principale crittografati e reciprocamente autenticabili da un KDH specifico per un HSM specifico.

Le chiavi master del profilo vengono archiviate con la configurazione del profilo e le informazioni di contesto crittografate per regione.

Le chiavi del cliente sono archiviate in blocchi chiave, protetti da una chiave master del profilo.

Tutte le chiavi esistono esclusivamente all'interno di un HSM o sono archiviate protette da un'altra chiave con una forza crittografica uguale o superiore.

Durabilità

Le chiavi del cliente per la crittografia delle transazioni e le funzioni aziendali devono essere disponibili anche in situazioni estreme che in genere causerebbero interruzioni. AWS La crittografia dei pagamenti utilizza un modello di ridondanza a più livelli tra zone e regioni di disponibilità. AWS Il cliente che richiede una disponibilità e una durabilità maggiori per le operazioni crittografiche di pagamento rispetto a quelle fornite dal servizio deve implementare architetture multiregionali.

L'autenticazione HSM e i token della chiave principale vengono salvati e possono essere utilizzati per ripristinare una chiave principale o sincronizzarsi con una nuova chiave principale, nel caso in cui sia necessario reimpostare un HSM. I token vengono archiviati e utilizzati solo sotto doppio controllo quando necessario.

Sicurezza delle comunicazioni

Esterno

AWS Gli endpoint dell'API Payment Cryptography soddisfano gli standard AWS di sicurezza tra cui TLS versione 1.2 o superiore e Signature Version 4 per l'autenticazione e l'integrità delle richieste.

Le connessioni TLS in entrata vengono terminate sui sistemi di bilanciamento del carico di rete e inoltrate ai gestori API tramite connessioni TLS interne.

Interno

Le comunicazioni interne tra i componenti del servizio e tra i componenti del servizio e altri servizi AWS sono protette da TLS utilizzando una crittografia avanzata.

Gli HSM si trovano su una rete privata e non virtuale raggiungibile solo dai componenti del servizio. Tutte le connessioni tra HSM e i componenti del servizio sono protette con TLS reciproco (mTLS), pari o superiore a TLS 1.2. I certificati interni per TLS e MTL sono gestiti da Amazon Certificate Manager utilizzando un'autorità di certificazione privata AWS. I VPC interni e la rete HSM vengono monitorati per attività e modifiche di configurazione impreviste.

Gestione delle chiavi dei clienti

At AWS, la fiducia dei clienti è la nostra massima priorità. Mantieni il pieno controllo delle chiavi che carichi o crei nel servizio con il tuo account AWS e la responsabilità della configurazione dell'accesso alle chiavi.

AWS Payment Cryptography ha la piena responsabilità della conformità fisica HSM e della gestione delle chiavi per le chiavi gestite dal servizio. Ciò richiede la proprietà e la gestione delle chiavi principali HSM e l'archiviazione delle chiavi protette del cliente all'interno del database delle chiavi di AWS Payment Cryptography.

Separazione dello spazio delle chiavi del cliente

AWS Payment Cryptography applica politiche chiave per tutti gli usi delle chiavi, inclusa la limitazione dei principi all'account proprietario della chiave, a meno che una chiave non venga esplicitamente condivisa con un altro account.

Backup e ripristino

Il backup delle chiavi e delle informazioni chiave di una regione viene eseguito in archivi crittografati da AWS. Gli archivi richiedono un doppio controllo AWS per il ripristino.

Blocchi chiave

Tutte le chiavi sono memorizzate in blocchi chiave in formato ANSI X9 TR-31.

Le chiavi possono essere importate nel servizio da crittogrammi o altri formati di blocchi di chiavi supportati da ImportKey. Allo stesso modo, le chiavi possono essere esportate, se esportabili, in altri formati di blocchi di chiavi o crittogrammi supportati dai profili di esportazione delle chiavi.

Uso delle chiavi

L'uso delle chiavi è limitato a quello configurato KeyUsage dal servizio. Il servizio fallirà qualsiasi richiesta con utilizzo della chiave, modalità di utilizzo o algoritmo inappropriati per l'operazione di crittografia richiesta.

Relazioni di scambio chiave

PCI PIN Security e PCI P2PE richiedono che le organizzazioni che condividono chiavi che crittografano i PIN, inclusa la KEK utilizzata per condividere tali chiavi, non condividano tali chiavi con altre organizzazioni. È consigliabile condividere le chiavi simmetriche solo tra due parti, anche all'interno della stessa organizzazione. Ciò riduce al minimo l'impatto dei sospetti compromessi chiave che impongono la sostituzione delle chiavi interessate.

Anche i casi aziendali che richiedono la condivisione delle chiavi tra più di 2 parti dovrebbero mantenere il numero di parti al minimo.

AWS Payment Cryptography fornisce tag chiave che possono essere utilizzati per tracciare e far rispettare l'utilizzo delle chiavi entro tali requisiti.

Ad esempio, KEK e BDK per diversi impianti di iniezione di chiavi possono essere identificati impostando un «KIF» = «POSStation» per tutte le chiavi condivise con quel fornitore di servizi. Un altro esempio potrebbe essere quello di etichettare le chiavi condivise con le reti di pagamento con «Network» = «». PayCard L'etichettatura consente di creare controlli di accesso e creare report di audit per applicare e dimostrare le pratiche di gestione chiave.

Eliminazione delle chiavi

DeleteKey contrassegna le chiavi nel database per l'eliminazione dopo un periodo configurabile dal cliente. Dopo questo periodo la chiave viene eliminata irrimediabilmente. Si tratta di un meccanismo di sicurezza per impedire l'eliminazione accidentale o dolosa di una chiave. Le chiavi contrassegnate per l'eliminazione non sono disponibili per nessuna azione tranne RestoreKey.

Le chiavi eliminate rimangono nei backup dei servizi per 7 giorni dopo l'eliminazione. Non sono ripristinabili durante questo periodo.

Le chiavi che appartengono agli account AWS chiusi sono contrassegnate per l'eliminazione. Se l'account viene riattivato prima del termine di eliminazione, tutte le chiavi contrassegnate per l'eliminazione vengono ripristinate, ma disattivate. È necessario riattivarle dall'utente per poterle utilizzare per operazioni crittografiche.

Condivisione delle chiavi

Le chiavi possono essere condivise con altri account all'interno o all'esterno dell'organizzazione utilizzando AWS Resource Access Manager (<https://docs.aws.amazon.com/ARG/index.html>). Le chiavi possono essere raggruppate in una condivisione di risorse e poi condivise con un account o con utenti e ruoli IAM specifici all'interno di un account. È possibile specificare le autorizzazioni di utilizzo per ogni condivisione di risorse. Le autorizzazioni di condivisione sono limitate da una politica delle risorse chiave. Una chiave condivisa non consentirà un'azione limitata dalla propria politica. L'autorizzazione alla condivisione può essere revocata in qualsiasi momento.

Registrazione di log e monitoraggio

I registri di servizio interni includono:

- CloudTrail registri delle chiamate al servizio AWS effettuate dal servizio
- CloudWatch registri di entrambi gli eventi registrati direttamente nei log o CloudWatch negli eventi da HSM
- File di registro da HSM e dai sistemi di servizio
- Archivi di registro

Tutte le fonti di registro monitorano e filtrano le informazioni sensibili, incluse quelle relative alle chiavi. I log vengono esaminati sistematicamente per garantire che contengano informazioni riservate sui clienti e non contengano informazioni riservate.

L'accesso ai registri è limitato alle persone necessarie per completare i ruoli lavorativi.

Tutti i log vengono conservati in linea con le policy di conservazione dei log di AWS.

Operazioni con i clienti

AWS Payment Cryptography ha la piena responsabilità della conformità fisica dell'HSM agli standard PCI. Il servizio fornisce anche un archivio sicuro delle chiavi e garantisce che le chiavi possano essere utilizzate solo per gli scopi consentiti dagli standard PCI e specificati dall'utente durante la creazione o l'importazione. L'utente è responsabile della configurazione degli attributi chiave e dell'accesso per sfruttare le funzionalità di sicurezza e conformità del servizio.

Argomenti

- [Generazione delle chiavi](#)

- [Importazione delle chiavi](#)
- [Esportazione delle chiavi](#)
- [Eliminazione delle chiavi](#)
- [Rotazione delle chiavi](#)

Generazione delle chiavi

Quando si creano le chiavi, si impostano gli attributi utilizzati dal servizio per imporre l'uso conforme della chiave:

- Algoritmo e lunghezza della chiave
- Utilizzo
- Disponibilità e scadenza

I tag utilizzati per il controllo degli accessi basato sugli attributi (ABAC) vengono utilizzati per limitare l'utilizzo delle chiavi con partner o applicazioni specifici, inoltre è necessario impostare durante la creazione. Assicurati di includere politiche per limitare i ruoli autorizzati a eliminare o modificare i tag.

È necessario assicurarsi che le politiche che determinano i ruoli che possono utilizzare e gestire la chiave siano impostate prima della creazione della chiave.

Note

Le politiche IAM relative ai CreateKey comandi possono essere utilizzate per applicare e dimostrare il doppio controllo per la generazione delle chiavi.

Importazione delle chiavi

Quando si importano le chiavi, gli attributi per imporre un uso conforme della chiave vengono impostati dal servizio utilizzando le informazioni legate crittograficamente nel blocco chiave. [Il meccanismo per impostare il contesto chiave fondamentale consiste nell'utilizzare blocchi chiave creati con l'HSM di origine e protetti da una KEK condivisa o asimmetrica.](#) Ciò è in linea con i requisiti del PIN PCI e preserva l'utilizzo, l'algoritmo e la forza della chiave dell'applicazione di origine.

Oltre alle informazioni contenute nel blocco chiave, è necessario stabilire importanti attributi chiave, tag e politiche di controllo degli accessi al momento dell'importazione.

L'importazione di chiavi mediante crittogrammi non trasferisce gli attributi chiave dall'applicazione di origine. È necessario impostare gli attributi in modo appropriato utilizzando questo meccanismo.

Spesso le chiavi vengono scambiate utilizzando componenti di testo in chiaro, trasmesse dai custodi delle chiavi e quindi caricate con una cerimonia che prevede il doppio controllo in una stanza sicura. Questo non è supportato direttamente da AWS Payment Cryptography. L'API esporterà una chiave pubblica con un certificato che può essere importato dal proprio HSM per esportare un blocco chiave importabile dal servizio. Consente l'uso del proprio HSM per caricare componenti in testo non crittografato.

È necessario utilizzare Key check values (KCV) per verificare che le chiavi importate corrispondano alle chiavi di origine.

Le politiche IAM sull' ImportKey API possono essere utilizzate per applicare e dimostrare il doppio controllo per l'importazione delle chiavi.

Esportazione delle chiavi

La condivisione delle chiavi con partner o applicazioni locali può richiedere l'esportazione delle chiavi. L'utilizzo di blocchi chiave per le esportazioni mantiene un contesto chiave fondamentale con il materiale chiave crittografato.

I tag chiave possono essere utilizzati per limitare l'esportazione in KEK di chiavi che condividono lo stesso tag e lo stesso valore.

AWS La crittografia dei pagamenti non fornisce né visualizza componenti chiave in testo chiaro. Ciò richiede l'accesso diretto da parte dei custodi delle chiavi a dispositivi crittografici sicuri (SCD) testati PCI PTS HSM o ISO 13491 per la visualizzazione o la stampa. Puoi stabilire una KEK asimmetrica o una KEK simmetrica con il tuo SCD per condurre la cerimonia di creazione dei componenti chiave in testo chiaro sotto doppio controllo.

È necessario utilizzare i valori di controllo delle chiavi (KCV) per verificare che le chiavi importate dall'HSM di destinazione corrispondano alle chiavi di origine.

Eliminazione delle chiavi

È possibile utilizzare l'API di eliminazione della chiave per pianificare l'eliminazione delle chiavi dopo un periodo di tempo configurato. Prima di quel momento le chiavi erano recuperabili. Una volta eliminate, le chiavi vengono rimosse definitivamente dal servizio.

Le policy IAM sull' DeleteKey API possono essere utilizzate per applicare e dimostrare il doppio controllo per l'eliminazione delle chiavi.

Rotazione delle chiavi

L'effetto della rotazione delle chiavi può essere implementato utilizzando l'alias chiave creando o importando una nuova chiave, quindi modificando l'alias della chiave per fare riferimento alla nuova chiave. La vecchia chiave verrebbe eliminata o disabilitata, a seconda delle pratiche di gestione.

Quote per AWS Payment Cryptography

L'account AWS dispone delle seguenti quote predefinite, precedentemente definite limiti, per ogni servizio AWS. Salvo diversa indicazione, ogni quota è specifica per regione. Se per alcune quote è possibile richiedere aumenti, altre quote non possono essere modificate.

Nome	Predefinita	Adattate	Descrizione
Alias	Ogni regione supportata: 2.000	Sì	Il numero massimo di alias che puoi avere in questo account nella regione corrente.
Frequenza combinata di richieste del piano di controllo	Ogni regione supportata: 5 al secondo	Sì	Il numero massimo di richieste di piano di controllo al secondo che è possibile effettuare in questo account nella regione corrente. Questa quota si applica a tutte le operazioni del piano di controllo combinate.
Velocità combinata di richieste sul piano dati (asimmetrica)	Ogni regione supportata: 20 al secondo	Sì	Il numero massimo di richieste al secondo per le operazioni sul piano dati con una chiave asimmetrica che è possibile effettuare in questo account nella regione corrente. Questa quota si applica a tutte le operazioni sul piano dati combinate.

Nome	Predefinita	Adattata	Descrizione
Velocità combinata di richieste sul piano dati (simmetrica)	Ogni regione supportata: 500 al secondo	Sì	Il numero massimo di richieste al secondo per le operazioni sul piano dati con una chiave simmetrica che è possibile effettuare in questo account nella regione corrente. Questa quota si applica a tutte le operazioni sul piano dati combinate.
Chiavi	Ogni regione supportata: 2.000	Sì	Il numero massimo di chiavi che puoi avere in questo account nella regione corrente, escluse le chiavi eliminate.

Cronologia dei documenti per la AWS Payment Cryptography User Guide

La tabella seguente descrive le versioni della documentazione per AWS Payment Cryptography.

Modifica	Descrizione	Data
Rilascio di funzionalità	Aggiungere informazioni sugli endpoint VPC (PrivateLink) ed esempi iCVV.	30 maggio 2024
Rilascio di funzionalità	Sono state aggiunte informazioni sulle nuove funzionalità relative all'importazione/e sportazione di chiavi tramite RSA e all'esportazione di chiavi DUKPT IPEK/IK.	15 gennaio 2024
Versione iniziale	Versione iniziale della AWS Payment Cryptography User Guide	8 giugno 2023

Le traduzioni sono generate tramite traduzione automatica. In caso di conflitto tra il contenuto di una traduzione e la versione originale in Inglese, quest'ultima prevarrà.