Secure Packager and Encoder Key Exchange API Specification

Partner and Customer Guide

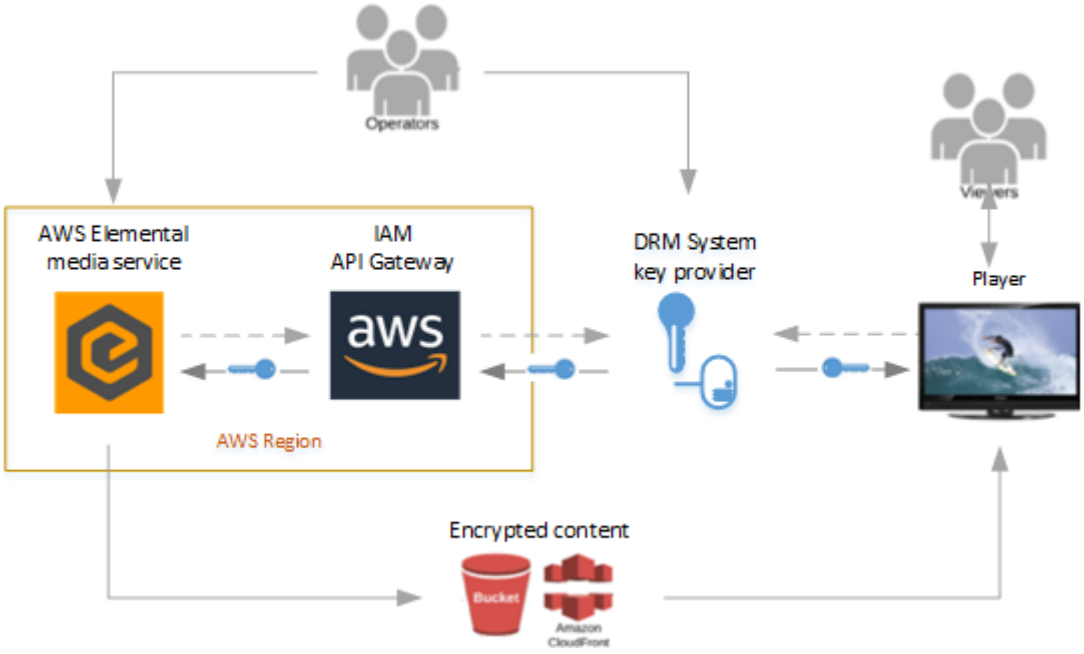
Secure Packager and Encoder Key Exchange API Specification: Partner and Customer Guide

Table of Contents

What Is Secure Packager and Encoder Key Exchange?

Secure Packager and Encoder Key Exchange (SPEKE) is part of the AWS Elemental content encryption protection strategy for media services customers. SPEKE defines the standard for communication between our media services and digital rights management (DRM) system key servers. SPEKE is used to encrypt video on demand (VOD) content through AWS Elemental MediaConvert and for live content through AWS Elemental MediaPackage.

The following illustration shows a high-level view of the AWS Elemental content encryption architecture.



These are the main services and components:

- **AWS Elemental media service**—Provides the encryption technology. The service receives encryption requests from its operator and retrieves the required keys from the DRM key server, through Amazon API Gateway. It delivers the encrypted content to Amazon S3 buckets or Amazon CloudFront. The AWS Elemental media service and the API Gateway must be instantiated in the same AWS region.
- **AWS IAM and API Gateway**—Manages customer trusted roles and proxy communication between the media service and the key server. API Gateway provides logging capabilities and lets customers control their relationships with the AWS Elemental media service and with the DRM system. Customers enable key server access through IAM role configuration. The API Gateway must reside in the same AWS region as the AWS Elemental media service.
- **DRM system key server**—Provides encryption keys to the AWS Elemental media services through a SPEKE-compliant API. Also provides licenses to media players for decryption.

How to Get Started

Are you a customer?

Partner with an AWS Elemental DRM solution provider to get set up to use encryption. For details, see *Customer Onboarding* (p. 3).

Are you a DRM solution provider or a customer with your own key server?

Expose a REST API for your key server in compliance with our AWS Elemental SPEKE specification. For details, see *SPEKE API Specification for DRM Solution Providers* (p. 5).

Customer Onboarding

Protect your content from unauthorized use by combining a digital rights management (DRM) system key server with your AWS Elemental media services and with your media players. Follow the steps in this chapter to get started using encryption with your AWS Elemental media services.

Step 1: Check Supported Technologies

The following support matrices show the DRM system support for each streaming protocol. Verify that your streaming protocol and the DRM system that you want are available for your live or VOD service.

AWS Elemental MediaPackage

	Microsoft PlayReady	Google Widevine	Apple Fairplay	AES-128
DASH	√ with key rotation	√ with key rotation		
Apple HLS			√ with key rotation	√ with key rotation
Microsoft Smooth	√			
CMAF Apple HLS			√ with key rotation	

AWS Elemental MediaConvert

	Microsoft Playready	Google Widevine	Apple Fairplay	AES-128
DASH	√	√		
Apple HLS			√	√
Microsoft Smooth	√			
CMAF Apple HLS				
CMAF DASH				

Step 2: Get On Board with a DRM Solution Provider

The following Amazon partners provide third-party DRM system implementations for AWS Elemental products. For details about each solution provider's offerings and information about how to contact them, follow the links to their Amazon Partner Network pages. The partners will help you get set up to use their solutions with AWS Elemental media services.

- Axinom
- BuyDRM

- castLabs
- Conax AS
- EZDRM
- INKA Entworks
- Irdeto
- Kaltura
- Verimatrix

We use a standard key exchange protocol, which is documented in our Secure Packager and Encoder Key Exchange (SPEKE) Digital Rights Management (DRM) specification at *SPEKE API Specification for DRM Solution Providers* (p. 5). Our DRM solution providers have integrated with AWS Elemental media services by exposing a SPEKE-compliant REST API.

SPEKE API Specification for DRM Solution Providers

This is the REST API specification for Secure Packager and Encoder Key Exchange (SPEKE). Use this specification to write a REST API for your digital rights management (DRM) system key server that is compatible with AWS Elemental MediaConvert and AWS Elemental MediaPackage.

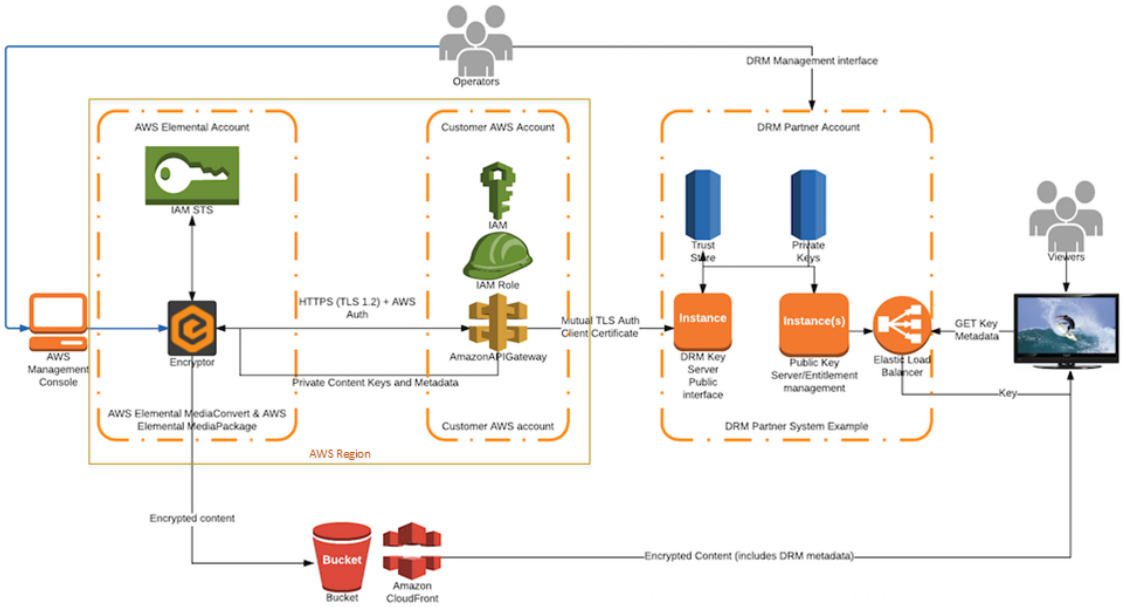
In a video streaming workflow, the encryption engine communicates with the DRM system key store to request content keys. These keys are highly sensitive, so it is critical that the key store and encryption engine establish a highly secure, trusted communication channel.

This specification addresses the following goals:

- Define a simple, trusted, highly secure interface that DRM vendors and customers can use to integrate with AWS Elemental products when content encryption is required.
- Cover VOD and live workflows, and include the error conditions and the authentication mechanisms that are required for robust, highly secure communication between AWS Elemental products and DRM key server endpoints.
- Include support for HLS, MSS, and DASH packaging and their common DRM systems (Fairplay, PlayReady, and Widevine/CENC).
- Keep the specification simple and extensible, to support future DRM systems.
- Use a simple REST API.

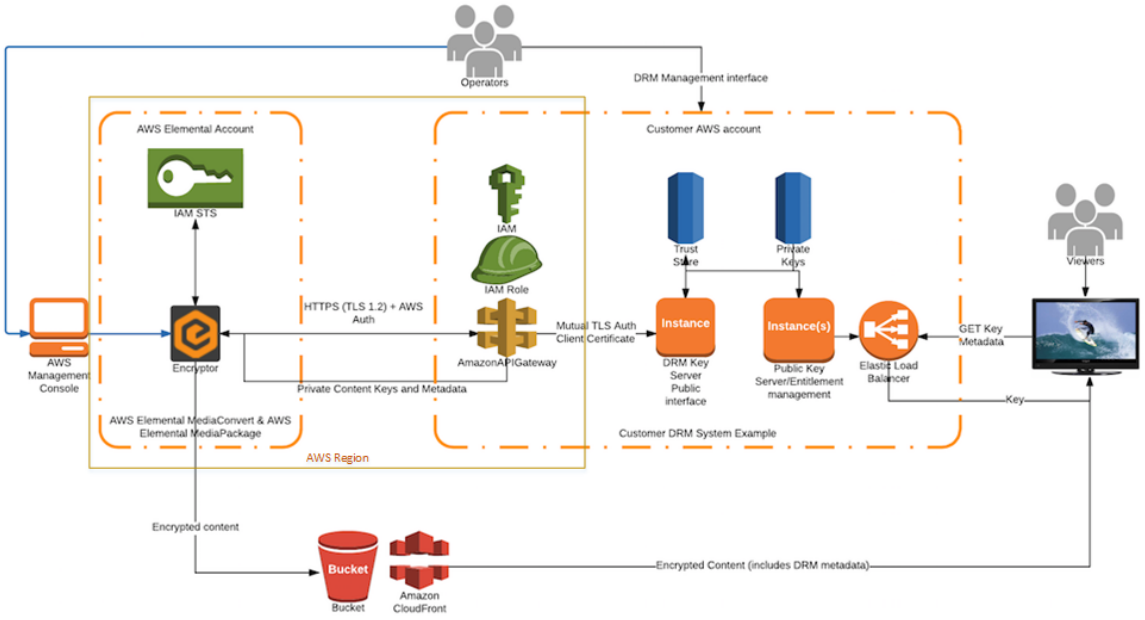
Architectural Overview

You can implement SPEKE for any DRM key server. The following illustration shows a typical architecture using a third-party key server.



- **AWS Elemental Account**—Provides the encryption technology. The service receives encryption requests from its operator and retrieves the required keys from the DRM key server, through Amazon API Gateway. It saves the encrypted content to Amazon S3 buckets or Amazon CloudFront.
- **Customer AWS Account**—Management of customer trusted roles in the AWS system and DRM key server and proxy communication between the media service and the key server. Customers enable key server access through IAM role configuration for their account. API Gateway provides logging capabilities and lets the customer control their relationships with the AWS Elemental service and with the DRM system. The API Gateway must reside in the same AWS region as the AWS Elemental encryptor media service.
- **DRM Partner Account**—Provides secure keys to the encryptor. Provides encryption keys to the AWS Elemental services through a SPEKE-compliant API. Provides secure licenses to media players for decryption on behalf of viewers.

The following illustration shows a typical architecture for a customer-implemented key server. In this case, the customer account and partner account are combined. The API Gateway and AWS Elemental media service must be instantiated in the same AWS region.



Are You New to SPEKE?

This section lists common SPEKE terminology and provides links to related services and specifications.

Terminology

- **ARN** – Amazon Resource Name. Uniquely identifies an AWS resource.
- **Content Key** – Cryptographic key used for encrypting part of the content.
- **Content Provider** – Publisher who provides the rights and rules for delivering protected media. The content provider might also provide source media (mezzanine format, for transcoding), asset identifiers, key identifiers (KID), key values, encoding instructions, and content description metadata.
- **Encryptor** – Video processing component (packaging stage as part of compressor or packager). For example, AWS Elemental MediaConvert and AWS Elemental MediaPackage.

- **Key Server** – Component of a DRM system that is used to provide keys to the encryptor.
- **Operator** – Person in charge of operating the overall system, including the encryptor and the DRM system.
- **Player** – Media player operating on behalf of a viewer. Gets its information from different sources, including the media manifest files, media files, and DRM licenses. Requests licenses from the DRM server on behalf of the viewers.

Related Services and Specifications

- AWS AssumeRole
- API Gateway Permissions
- AWS Sigv4
- DASH-IF CPIX specification
- DASH-IF System IDs

AWS Authentication for SPEKE

SPEKE requires AWS authentication through IAM roles for use with AWS Elemental media services. IAM roles are created by the DRM system service or by the operator who owns the DRM endpoint in an AWS account. Each role is assigned an Amazon Resource Name (ARN), which the AWS Elemental product operator provides in the service UI when requesting encryption. The role's policy permissions must be configured to give permission to access the key server API and no other AWS resource access. When the encryptor contacts the DRM key server, it uses the role ARN to assume the role of the key server account holder, which returns temporary credentials for the encryptor to use to access the key server.

One common implementation involves the operator or DRM vendor using Amazon API Gateway in front of the key server, and then enabling AWS_IAM authorization on the API Gateway resource. You can use the following policy definition example and attach it to a new role to give permissions to the appropriate resource. In this case the permissions are for all API Gateway resources.

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Effect": "Allow",
            "Action": [
                "execute-api:Invoke"
            ],
            "Resource": [
                "arn:aws:execute-api:us-west-2:*:*/*/GET/*"
            ]
        }
    ]
}
```

Finally, the role requires the addition of a trust relationship and the operator must be able to select the service.

The following example shows a role ARN that is created for accessing the key server:

```
arn:aws:iam::2949266363526:role/DRMKeyServer
```

For more information about the creation of a role, see AWS AssumeRole. For more information about signing a request, see AWS Sigv4.

SPEKE API

To work with AWS Elemental services, your key server must expose the REST API described in this specification. The encryptor makes requests to the API to exchange the payload with your key server.

SPEKE uses the DASH Industry Forum Content Protection Information Exchange Format (DASH-IF-CPIX) data structure definition for key exchange. DASH-IF-CPIX defines a schema to provide an extensible, multi-DRM exchange from the DRM system to the encryptor. This enables content encryption for all adaptive bitrate packaging formats at the time of content compression and packaging. Adaptive bitrate packaging formats include HLS, DASH, and MSS.

For detailed information about the exchange format, see the DASH Industry Forum CPIX specification at http://dashif.org/wp-content/uploads/2016/11/DASH-IF-CPIX-v2-0.pdf.

The SPEKE API payload response conforms to DASH-IF-CPIX with the following constraints and customizations:

- SPEKE follows the Encryptor Consumer workflow.
- SPEKE does not use the encrypted document feature. Instead, it relies on encryption at the transport layer, plus strong authentication.
- SPEKE requires the `ContentKeyUsageRule` filter, `KeyPeriodFilter` for rotating key workflows. SPEKE ignores all other `ContentKeyUsageRules`.
- SPEKE omits the `UpdateHistoryItemList` functionality. If the list is present in the response, SPEKE ignores it.
- SPEKE supports key rotation. SPEKE uses only the `ContentKeyPeriod` `@index` to track the key period.
- To support MSS Playready, SPEKE uses a custom parameter under the `DRMSystem` tag, `SPEKE:ProtectionHeader`.
- For HLS packaging, if the `URIExtXKey` is present in the response, then it contains the full data to be added in the URI parameter of the `EXT-X-KEY` tag of an HLS playlist, with no further signaling requirement.
- For HLS playlist, under the `DRMSystem` tag, SPEKE provides the optional custom parameters `speke:KeyFormat` and `speke:KeyFormatVersions`, for the values of the `KEYFORMAT` and `KEYFORMATVERSIONS` parameters of the `EXT-X-KEY` tag.

  The HLS IV always follows segment number unless explicitly specified by the operator.
- When requesting keys, the encryptor might use the optional `@explicitIV` attribute on the `ContentKey` element. The key server can respond with an IV using `@explicitIV`, even if the attribute is not included in the request.
- The encryptor creates the key identifier (`KID`). It does not change for a given content ID and key period. The key server must include the `KID` in the request document response.
- The key server might include a value for the `Speke-User-Agent` response header to identify itself for debugging purposes.
- SPEKE does not currently support multiple tracks or keys per content.

The AWS Elemental encryptor acts as a client and sends `POST` operations to the key server endpoint. The encryptor might send a periodic `heartbeat` request to ensure that the connection between the encryptor and the key server endpoint is healthy.

Live Workflow Method Calls

Request Syntax Example

The following URL is an example and does not indicate a fixed format.

```
POST https://speke-compatible-server/speke/v1.0/copyProtection
```

Request Body

A CPIX element.

Request Headers

Name	Type	Occurs	Description
`Authorization`	String	1..1	See AWS Sigv4
`X-Amz-Security-Token`	String	1..1	See AWS Sigv4
`X-Amz-Date`	String	1..1	See AWS Sigv4
`Content-Type`	String	1..1	application/xml

Response Headers

Name	Type	Occurs	Description
`Speke-User-Agent`	String	1..1	String that identifies the key server
`Content-Type`	String	1..1	application/xml

Request Response

HTTP CODE	Payload Name	Occurs	Description
`200 (Success)`	CPIX	1..1	DASH-CPIX payload response
`4XX (Client error)`	Client error message	1..1	Description of the client error
`5XX (Server error)`	Server error message	1..1	Description of the server error

Live Example Request

The following is an example listing only and cannot be run.

```
<cpix:CPIX id="abc123" xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc"
xmlns:speke="urn:aws:amazon:com:speke">
    <cpix:ContentKeyList>
        <cpix:ContentKey kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        explicitIV="OFj2IjCsPJFfMAxmQxLGPw==">></cpix:ContentKey>
    </cpix:ContentKeyList>
    <cpix:DRMSystemList>
        <!-- HLS AES-128 (systemId is implementation specific)-->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="81376844-f976-481e-a84e-cc25d39b0b33">
            <cpix:URIExtXKey></cpix:URIExtXKey>
            <speke:KeyFormat></speke:KeyFormat>
            <speke:KeyFormatVersions></speke:KeyFormatVersions>
        </cpix:DRMSystem>

        <!-- HLS SAMPLE-AES -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="94ce86fb-07ff-4f43-adb8-93d2fa968ca2">
            <cpix:URIExtXKey></cpix:URIExtXKey>
            <speke:KeyFormat></speke:KeyFormat>
            <speke:KeyFormatVersions></speke:KeyFormatVersions>
        </cpix:DRMSystem>

        <!-- Common encryption (Widevine)-->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">
            <cpix:PSSH></cpix:PSSH>
        </cpix:DRMSystem>

        <!-- Common encryption / MSS (Playready) -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="9a04f079-9840-4286-ab92-e65be0885f95">
            <speke:ProtectionHeader></speke:ProtectionHeader>
            <cpix:PSSH></cpix:PSSH>
        </cpix:DRMSystem>
    </cpix:DRMSystemList>
    <cpix:ContentKeyPeriodList>
        <cpix:ContentKeyPeriod id="keyPeriod_0909829f-40ff-4625-90fa-75da3e53278f" index="1" />
    </cpix:ContentKeyPeriodList>
    <cpix:ContentKeyUsageRuleList>
        <cpix:ContentKeyUsageRule kid="98ee5596-cd3e-a20d-163a-e382420c6eff">
            <cpix:KeyPeriodFilter periodId="keyPeriod_0909829f-40ff-4625-90fa-75da3e53278f" />
        </cpix:ContentKeyUsageRule>
    </cpix:ContentKeyUsageRuleList>
</cpix:CPIX>
```

Live Example Response

The following is an example listing only and cannot be run.

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc"
xmlns:speke="urn:aws:amazon:com:speke" id="abc123">
    <cpix:ContentKeyList>
        <cpix:ContentKey explicitIV="OFj2IjCsPJFfMAxmQxLGPw==" kid="98ee5596-cd3e-a20d-163a-e382420c6eff">
            <cpix:Data>
                <pskc:Secret>
                    <pskc:PlainValue>5dGAgwGuUYu4dHeHtNlxJw==</pskc:PlainValue>
                </pskc:Secret>
            </cpix:Data>
        </cpix:ContentKey>
    </cpix:ContentKeyList>
    <cpix:DRMSystemList>
        <!-- HLS AES-128 (systemId is implementation specific) -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="81376844-f976-481e-a84e-cc25d39b0b33">

            <cpix:URIExtXKey>aHR0cHM6Ly83azR5dHV4cTVkLmV4ZWN1dGUtYXBpLnVzLXdlc3QtMi5hbWF6b25hd3MuY29tL0VrZVN0Y
            WdlL2NsaWVudC9hYmMxMjMvOThlZTU1OTYtY2QzZS1hMjBkLTE2M2EtZTM4MjQyMGM2ZWZm</cpix:URIExtXKey>
            <speke:KeyFormat>aWRlbnRpdHk=</speke:KeyFormat>
            <speke:KeyFormatVersions>MQ==</speke:KeyFormatVersions>
        </cpix:DRMSystem>

        <!-- HLS SAMPLE-AES -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="94ce86fb-07ff-4f43-adb8-93d2fa968ca2">

            <cpix:URIExtXKey>aHR0cHM6Ly83azR5dHV4cTVkLmV4ZWN1dGUtYXBpLnVzLXdlc3QtMi5hbWF6b25hd3MuY29tL0VrZVN0Y
            WdlL2NsaWVudC9hYmMxMjMvOThlZTU1OTYtY2QzZS1hMjBkLTE2M2EtZTM4MjQyMGM2ZWZm</cpix:URIExtXKey>
            <speke:KeyFormat>Y29tLmFwcGxlLnN0cmVhbWluZ2tleWRlbGl2ZXJ5</speke:KeyFormat>
            <speke:KeyFormatVersions>MQ==</speke:KeyFormatVersions>
        </cpix:DRMSystem>

        <!-- Common encryption (Widevine) -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">

            <cpix:PSSH>AAAAanBzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7QAAAEoIARIQeSIcblaNbb7Dji6sAtKZzRoNd2lkZXZpbmVfdGV
            zdCIfa2V5LWlkOmVTSWNibGFOYmI3RGppNnNBdEtaelE9PSoCU0QyAA==</cpix:PSSH>
        </cpix:DRMSystem>

        <!-- Common encryption / MSS (Playready) -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="9a04f079-9840-4286-ab92-e65be0885f95">

            <speke:ProtectionHeader>CgMAAAEAAQAAAzwAVwBSAE0ASABFAEEARABFAFIAIAB4AG0AbABuAHMAPQAiAGgAdAB0AHAAOg
            AvAC8AcwBjAGgAZQBtAGEAcwAuAG0AaQBjAHIAbwBzAG8AZgB0AC4AYwBvAG0ALwBEAFIATQAvADIAMAAwADcALwAwADMALwBQ
            AGwAYQB5AFIAZQBhAGQAeQBIAGUAYQBkAGUAcgAiACAAdgBlAHIAcwBpAG8AbgA9ACIANAAuADAALgAwAC4AMAAiAD4APABEAE
            EAVABBAD4APABQAFIATwBUAEUAQwBUAEkATgBGAE8APgA8AEsARQBZAEwARQBOAD4AMQA2ADwALwBLAEUAWQBMAEUATgA+ADwA
            QQBMAEcASQBEAD4AQQBFAFMAQwBUAFIAPAAvAEEATABHAEkARAA+ADwALwBQAFIATwBUAEUAQwBUAEkATgBGAE8APgA8AEsASQ
            BEAD4ATwBXAGoAaAB0AHIAMwB1ADkAawArAHIAZABvADEASQBMAFkAMAByAGEAdwA9AD0APAAvAEsASQBEAD4APABDAEgARQBD
            AEsAUwBVAE0APgBCADMAQQA2AEEAMwB4AG0AdABkAEkAPQA8AC8AQwBIAEUAQwBLAFMAVQBNAD4APABMAEEAXwBVAFIATAA+AG
            gAdAB0AHAAOgAvAC8AcABsAGEAeQByAGUAYQBkAHkALgBkAGkAcgBlAGMAdAB0AGEAcABzAC4AbgBlAHQALwBwAHIALwBzAHYA
            YwAvAHIAaQBnAGgAdABzAG0AYQBuAGEAZwBlAHIALgBhAHMAbQB4AD8AUABsAGEAeQBSAGkAZwBoAHQAPQAxACYAYQBtAHAAOw
            BhAG0AcAA7AGEAbQBwADsAVQBzAGUAUwBpAG0AcABsAGUATgBvAG4AUABlAHIAcwBpAHMAdABlAG4AdABMAGkAYwBlAG4AcwBl
            AD0AMQA8AC8ATABBAF8AVQBSAEwAPgA8AC8ARABBAFQAQQA+ADwALwBXAFIATQBIAEUAQQBEAEUAUgA+AA==</speke:Protec
            tionHeader>

            <cpix:PSSH>AAADMHBzc2gAAAAAmgTweZhAQoarkuZb4IhflQAAAxAQAwAAAQABAAYDPABXAFIATQBIAEUAQQBEAEUAUgAgAHg
            AbQBsAG4AcwA9ACIAaAB0AHQAcAA6AC8ALwBzAGMAaABlAG0AYQBzAC4AbQBpAGMAcgBvAHMAbwBmAHQALgBjAG8AbQAvAEQAU
            gBNAC8AMgAwADAANwAvADAAMwAvAFAAbABhAHkAUgBlAGEAZAB5AEgAZQBhAGQAZQByACIAIAB2AGUAcgBzAGkAbwBuAD0AIgA
            0AC4AMAAuADAALgAwACIAPgA8AEQAQQBUAEEAPgA8AFAAUgBPAFQARQBDAFQASQBOAEYATwA+ADwASwBFAFkATABFAE4APgAxA
            DYAPAAvAEsARQBZAEwARQBOAD4APABBAEwARwBJAEQAPgBBAEUAUwBDAFQAUgA8AC8AQQBMAEcASQBEAD4APAAvAFAAUgBPAFQ
            ARQBDAFQASQBOAEYATwA+ADwASwBJAEQAPgBiAGgAdwBpAGUAWQAxAFcAdgBtADMARABqAGkANgBzAEEAdABLAFoAegBRAD0AP
            QA8AC8ASwBJAEQAPgA8AEMASABFAEMASwBTAFUATQA+AGEAVABtAFAASgBWAEMAVgBaADYAcwA9ADwALwBDAEgARQBDAEsAUwB
            VAE0APgA8AEwAQQBfAFUAUgBMAD4AaAB0AHQAcABzADoALwAvAHAAcgBsAHMALgBhAHQAdgAtAHAAcwAuAGEAbQBhAHoAbwBuA
            C4AYwBvAG0ALwBjAGQAcAA8AC8ATABBAF8AVQBSAEwAPgA8AEMAVQBTAFQATwBNAEEAVABUAFIASQBCAFUAVABFAFMAPgA8AEk
            ASQBTAF8ARABSAE0AXwBWAEUAUgBTAEkATwBOAD4ANwAuADEALgAxADQAMwA5AC4AMAA8AC8ASQBJAFMAXwBEAFIATQBfAFYAR
            QBSAFMASQBPAE4APgA8AC8AQwBVAFMAVABPAE0AQQBUAFQAUgBJAEIAVQBUAEUAUwA+ADwALwBEAEEAVABBAD4APAAvAFcAUgB
            NAEgARQBBAEQARQBSAD4A</cpix:PSSH>
        </cpix:DRMSystem>
    </cpix:DRMSystemList>
    <cpix:ContentKeyPeriodList>
        <cpix:ContentKeyPeriod id="keyPeriod_0909829f-40ff-4625-90fa-75da3e53278f" index="1" />
    </cpix:ContentKeyPeriodList>
    <cpix:ContentKeyUsageRuleList>
        <cpix:ContentKeyUsageRule kid="98ee5596-cd3e-a20d-163a-e382420c6eff">
            <cpix:KeyPeriodFilter periodId="keyPeriod_0909829f-40ff-4625-90fa-75da3e53278f" />
        </cpix:ContentKeyUsageRule>
    </cpix:ContentKeyUsageRuleList>
</cpix:CPIX>
```

VOD Workflow Method Calls

Request Syntax Example

The following URL is an example and does not indicate a fixed format.

```
POST https://speke-compatible-server/speke/v1.0/copyProtection
```

Request Body

A CPIX element.

Response Headers

Name	Type	Occurs	Description
`Speke-User-Agent`	String	1..1	String that identifies the key server
`Content-Type`	String	1..1	application/xml

Request Response

HTTP CODE	Payload Name	Occurs	Description
`200 (Success)`	CPIX	1..1	DASH-CPIX payload response
`4XX (Client error)`	Client error message	1..1	Description of the client error
`5XX (Server error)`	Server error message	1..1	Description of the server error

VOD Example Request

The following is an example listing only and cannot be run.

```
<cpix:CPIX id="abc123" xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc"
xmlns:speke="urn:aws:amazon:com:speke">
    <cpix:ContentKeyList>
        <cpix:ContentKey kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        explicitIV="OFj2IjCsPJFfMAxmQxLGPw==">></cpix:ContentKey>
    </cpix:ContentKeyList>
    <cpix:DRMSystemList>
        <!-- HLS AES-128 (systemId is implementation specific)-->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="81376844-f976-481e-a84e-cc25d39b0b33">
            <cpix:URIExtXKey></cpix:URIExtXKey>
            <speke:KeyFormat></speke:KeyFormat>
            <speke:KeyFormatVersions></speke:KeyFormatVersions>
        </cpix:DRMSystem>

        <!-- HLS SAMPLE-AES -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="94ce86fb-07ff-4f43-adb8-93d2fa968ca2">
            <cpix:URIExtXKey></cpix:URIExtXKey>
            <speke:KeyFormat></speke:KeyFormat>
            <speke:KeyFormatVersions></speke:KeyFormatVersions>
        </cpix:DRMSystem>

        <!-- Common encryption (Widevine)-->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">
            <cpix:PSSH></cpix:PSSH>
        </cpix:DRMSystem>

        <!-- Common encryption / MSS (Playready) -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="9a04f079-9840-4286-ab92-e65be0885f95">
            <speke:ProtectionHeader></speke:ProtectionHeader>
            <cpix:PSSH></cpix:PSSH>
        </cpix:DRMSystem>
    </cpix:DRMSystemList>
    </cpix:CPIX>
```

VOD Example Response

The following is an example listing only and cannot be run.

```
<cpix:CPIX xmlns:cpix="urn:dashif:org:cpix" xmlns:pskc="urn:ietf:params:xml:ns:keyprov:pskc"
xmlns:speke="urn:aws:amazon:com:speke" id="abc123">
    <cpix:ContentKeyList>
        <cpix:ContentKey explicitIV="OFj2IjCsPJFfMAxmQxLGPw==" kid="98ee5596-cd3e-a20d-163a-e382420c6eff">
            <cpix:Data>
                <pskc:Secret>
                    <pskc:PlainValue>5dGAgwGuUYu4dHeHtNlxJw==</pskc:PlainValue>
                </pskc:Secret>
            </cpix:Data>
        </cpix:ContentKey>
    </cpix:ContentKeyList>
    <cpix:DRMSystemList>
        <!-- HLS AES-128 (systemId is implementation specific) -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="81376844-f976-481e-a84e-cc25d39b0b33">

            <cpix:URIExtXKey>aHR0cHM6Ly83azR5dHV4cTVkLmV4ZWN1dGUtYXBpLnVzLXdlc3QtMi5hbWF6b25hd3MuY29tL0VrZVN0Y
            WdlL2NsaWVudC9hYmMxMjMvOThlZTU1OTYtY2QzZS1hMjBkLTE2M2EtZTM4MjQyMGM2ZWZm</cpix:URIExtXKey>
            <speke:KeyFormat>aWRlbnRpdHk=</speke:KeyFormat>
            <speke:KeyFormatVersions>MQ==</speke:KeyFormatVersions>
        </cpix:DRMSystem>

        <!-- HLS SAMPLE-AES -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="94ce86fb-07ff-4f43-adb8-93d2fa968ca2">

            <cpix:URIExtXKey>aHR0cHM6Ly83azR5dHV4cTVkLmV4ZWN1dGUtYXBpLnVzLXdlc3QtMi5hbWF6b25hd3MuY29tL0VrZVN0Y
            WdlL2NsaWVudC9hYmMxMjMvOThlZTU1OTYtY2QzZS1hMjBkLTE2M2EtZTM4MjQyMGM2ZWZm</cpix:URIExtXKey>
            <speke:KeyFormat>Y29tLmFwcGxlLnN0cmVhbWluZ2tleWRlbGl2ZXJ5</speke:KeyFormat>
            <speke:KeyFormatVersions>MQ==</speke:KeyFormatVersions>
        </cpix:DRMSystem>

        <!-- Common encryption (Widevine) -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">

            <cpix:PSSH>AAAAanBzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7QAAAEoIARIQeSIcblaNbb7Dji6sAtKZzRoNd2lkZXZpbmVfdGV
            zdCIfa2V5LWlkOmVTSWNibGFOYmI3RGppNnNBdEtaelE9PSoCU0QyAA==</cpix:PSSH>
        </cpix:DRMSystem>

        <!-- Common encryption / MSS (Playready) -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff"
        systemId="9a04f079-9840-4286-ab92-e65be0885f95">

            <speke:ProtectionHeader>CgMAAAEAAQAAAzwAVwBSAE0ASABFAEEARABFAFIAIAB4AG0AbABuAHMAPQAiAGgAdAB0AHAAOg
            AvAC8AcwBjAGgAZQBtAGEAcwAuAG0AaQBjAHIAbwBzAG8AZgB0AC4AYwBvAG0ALwBEAFIATQAvADIAMAAwADcALwAwADMALwBQ
            AGwAYQB5AFIAZQBhAGQAeQBIAGUAYQBkAGUAcgAiACAAdgBlAHIAcwBpAG8AbgA9ACIANAAuADAALgAwAC4AMAAiAD4APABEAE
            EAVABBAD4APABQAFIATwBUAEUAQwBUAEkATgBGAE8APgA8AEsARQBZAEwARQBOAD4AMQA2ADwALwBLAEUAWQBMAEUATgA+ADwA
            QQBMAEcASQBEAD4AQQBFAFMAQwBUAFIAPAAvAEEATABHAEkARAA+ADwALwBQAFIATwBUAEUAQwBUAEkATgBGAE8APgA8AEsASQ
            BEAD4ATwBXAGoAaAB0AHIAMwB1ADkAawArAHIAZABvADEASQBMAFkAMAByAGEAdwA9AD0APAAvAEsASQBEAD4APABDAEgARQBD
            AEsAUwBVAE0APgBCADMAQQA2AEEAMwB4AG0AdABkAEkAPQA8AC8AQwBIAEUAQwBLAFMAVQBNAD4APABMAEEAXwBVAFIATAA+AG
            gAdAB0AHAAOgAvAC8AcABsAGEAeQByAGUAYQBkAHkALgBkAGkAcgBlAGMAdAB0AGEAcABzAC4AbgBlAHQALwBwAHIALwBzAHYA
            YwAvAHIAaQBnAGgAdABzAG0AYQBuAGEAZwBlAHIALgBhAHMAbQB4AD8AUABsAGEAeQBSAGkAZwBoAHQAPQAxACYAYQBtAHAAOw
            BhAG0AcAA7AGEAbQBwADsAVQBzAGUAUwBpAG0AcABsAGUATgBvAG4AUABlAHIAcwBpAHMAdABlAG4AdABMAGkAYwBlAG4AcwBl
            AD0AMQA8AC8ATABBAF8AVQBSAEwAPgA8AC8ARABBAFQAQQA+ADwALwBXAFIATQBIAEUAQQBEAEUAUgA+AA==</speke:Protec
            tionHeader>

            <cpix:PSSH>AAADMHBzc2gAAAAAmgTweZhAQoarkuZb4IhflQAAAxAQAwAAAQABAAYDPABXAFIATQBIAEUAQQBEAEUAUgAgAHg
            AbQBsAG4AcwA9ACIAaAB0AHQAcAA6AC8ALwBzAGMAaABlAG0AYQBzAC4AbQBpAGMAcgBvAHMAbwBmAHQALgBjAG8AbQAvAEQAU
            gBNAC8AMgAwADAANwAvADAAMwAvAFAAbABhAHkAUgBlAGEAZAB5AEgAZQBhAGQAZQByACIAIAB2AGUAcgBzAGkAbwBuAD0AIgA
            0AC4AMAAuADAALgAwACIAPgA8AEQAQQBUAEEAPgA8AFAAUgBPAFQARQBDAFQASQBOAEYATwA+ADwASwBFAFkATABFAE4APgAxA
            DYAPAAvAEsARQBZAEwARQBOAD4APABBAEwARwBJAEQAPgBBAEUAUwBDAFQAUgA8AC8AQQBMAEcASQBEAD4APAAvAFAAUgBPAFQ
            ARQBDAFQASQBOAEYATwA+ADwASwBJAEQAPgBiAGgAdwBpAGUAWQAxAFcAdgBtADMARABqAGkANgBzAEEAdABLAFoAegBRAD0AP
            QA8AC8ASwBJAEQAPgA8AEMASABFAEMASwBTAFUATQA+AGEAVABtAFAASgBWAEMAVgBaADYAcwA9ADwALwBDAEgARQBDAEsAUwB
            VAE0APgA8AEwAQQBfAFUAUgBMAD4AaAB0AHQAcABzADoALwAvAHAAcgBsAHMALgBhAHQAdgAtAHAAcwAuAGEAbQBhAHoAbwBuA
            C4AYwBvAG0ALwBjAGQAcAA8AC8ATABBAF8AVQBSAEwAPgA8AEMAVQBTAFQATwBNAEEAVABUAFIASQBCAFUAVABfAFMAPgA8AEk
            ASQBTAF8ARABSAE0AXwBWAEUAUgBTAEkATwBOAD4ANwAuADEALgAxADQAMwA5AC4AMAA8AC8ASQBJAFMAXwBEAFIATQBfAFYAR
            QBSAFMASQBPAE4APgA8AC8AQwBVAFMAVABPAE0AQQBUAFQAUgBJAEIAVQBUAEUAUwA+ADwALwBEAEEAVABBAD4APAAvAFcAUgB
            NAEgARQBBAEQARQBSAD4A</cpix:PSSH>
        </cpix:DRMSystem>
    </cpix:DRMSystemList>
</cpix:CPIX>
```

Heartbeat

Request Syntax Example

The following URL is an example and does not indicate a fixed format.

```
GET https://speke-compatible-server/speke/v1.0/heartbeat
```

Request Response

HTTP CODE	Payload Name	Occurs	Description
`200 (Success)`	statusMessage	1..1	Message that describes the status

Document History

The following table describes the documentation for this release of SPEKE.

- **Latest documentation update:** November 27, 2017

Change	Description	Date
CMAF	Updated support matrix tables to include CMAF.	June 27, 2018
First release	First release of Secure Packager and Encoder Key Exchange, a specification for communication between a content encryption service and a DRM key provider. The DRM key provider exposes a Secure Packager and Encoder Key Exchange API to handle incoming key requests.	November 27, 2017

AWS Glossary

For the latest AWS terminology, see the AWS Glossary in the *AWS General Reference*.