Secure Packager and Encoder Key Exchange (SPEKE)

Open API Specification for Encoders, Transcoders, Packagers, and DRM Platforms

Lionel Bringuier – Director of Product Management, Video Delivery Services
Ken Shek – Specialist Solutions Architect M&E
What is the SPEKE API?

The Secure Packager and Encoder Key Exchange (SPEKE) is an open API specification which defines the standard for communication between encryptors and digital rights management (DRM) platforms.
Why do we need to use DRMs?

Protect and control access to content
- Monetize content by maintaining control and fulfillment

Market coverage
- Content producers protect Premium video content
- Sporting events example: FIFA WorldCup 2018

Playback Complexity
- Consumers watch content on various devices which all have specific Container/DRM requirements
- The DASH container offers Multi-DRM protected using Widevine and PlayReady
- Apple HLS is protected using Apple Fairplay
- Playback on Web Browsers, Multiscreen devices and Set-top boxes
**Key Terms**

**Encryptor**
- Encoders, transcoders, packagers

**CPIX**
- Content Protection Information Exchange format (DASH-IF)

**SystemID or schemeld**
- Unique ID for the underlying DRM vendor:
  - Microsoft PlayReady: 9a04f079-9840-4286-ab92-e65be0885f95
  - Google Widevine: edef8ba9-79d6-4ace-a3c8-27dcd51d21ed
- Registered at: https://dashif.org/identifiers/protection/

**Key ID (KID)**
- Identifier that points to the underlying Key similar to a hash table

**PSSH**
- Protection System Specific Header, as part of CENC (Common Encryption)
- Contains a reference to the KeyID, SystemID and custom data for that DRM vendor Stored as an MP4 box in fMP4
- Stored as base64 encoding for MP4 box in DASH MPD
SPEKE – Democratization of the video workflow

Encryptors
(Encoders, Transcoders and Packagers)
- Robust and lighter application
- Saves time, effort and cost of custom DRM API integration (4 weeks per custom integration)
- Savings in testing time and effort (~17% reduction in testing effort)
- Ability to test DRM workflow with reference servers

Content Providers
(MVPDS and Content distributors)
- Lowers barrier of DRM solution provider adoption
- Opportunity cost savings with quicker integration
- Ability to expand audience/device coverage

DRM Solution Providers
- Lowers barrier to adoption
- Custom integration cost and time savings
- Ability to establish proven workflows
The SPEKE Ecosystem

Several DRM solution providers have implemented SPEKE

SPEKE also enables customers to develop their own key management solution
SPEKE – CPIX Based Encryptor Consumer Model

- Packager
- Encryptor
- Key Server
- DRM System

CPIX Document V2
- Content Keys
- PSSH DRM signaling
- Key format and version

CPIX Document V1
- Key ID
- DRM system ID
SPEKE Request Sample – XML POST Over HTTP

```
xmlns:speke="urn:aws:amazon:com:speke">
  <cpix:ContentKeyList>
    <cpix:ContentKey kid="98ee5596-cd3e-a20d-163a-e382420c6eff" explicitIV="OFj2IjCsPJFfMAxmQxLGPw=">
      <cpix:DRMSystemList>
        <!-- Common encryption (Widevine) -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">
          <cpix:PSSH xmlns="urn:aws:amazon:com:speke"/>
        </cpix:DRMSystem>
        <!-- Common encryption / MSS (Playready) -->
        <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff" systemId="9a04f079-9840-4286-ab92-e65be0885f95">
          <speke:ProtectionHeader xmlns:speke="urn:aws:amazon:com:speke"/>
          <cpix:PSSH xmlns="urn:aws:amazon:com:speke"/>
        </cpix:DRMSystem>
      </cpix:DRMSystemList>
    </cpix:ContentKey>
  </cpix:ContentKeyList>
</cpix:CPIX>
```
SPEKE Response Sample – XML Over HTTP

<?xml version="1.0" encoding="UTF-8"?>
  <cpix:ContentKeyList>
    <cpix:ContentKey explicitIV="0Fj2IjCsPJffMAxmQxLGPw==" kid="98ee5596-cd3e-a20d-163a-e382420c6eff">
      <cpix:Data>
        <pskc:Secret>
          <pskc:PlainValue>5dGAgwGuUYu4dHeHTNl3w==</pskc:PlainValue>
        </pskc:Secret>
      </cpix:Data>
    </cpix:ContentKey>
    <!-- Common encryption (Widevine) -->
    <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff" systemId="edef8ba9-79d6-4ace-a3c8-27dcd51d0d1ed">
      <cpix:PSSH>AAAanBzc2gAAAAA7e+LqXnWSs6jyCfc1R0h7QAAAEAA==</cpix:PSSH>
    </cpix:DRMSystem>
    <!-- Common encryption / MSS (Playready) -->
    <cpix:DRMSystem kid="98ee5596-cd3e-a20d-163a-e382420c6eff" systemId="9a04f079-9840-4286-ab92-e65be0885f95">
      <speke:ProtectionHeader>CgMAAEAAQAAAzwA4wBSAE0ASABFAEERABFAIAB4A==</speke:ProtectionHeader>
      <cpix:PSSH>AAADNHBz2gAAAAAmgTweZhAOcarkuZ</cpix:PSSH>
    </cpix:DRMSystem>
  </cpix:ContentKeyList>
</cpix:CPIX>
DASH Manifest with multi-DRM signaling

```xml
<MPD xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <Period start="PT0S" id="1">
    <AdaptationSet mimeType="video/mp4" frameRate="30/1" subsegmentStartsWithSAP="true" bitstreamSwitching="false">
      <ContentProtection cenc:default_KID="886630ca-f637-4f0c-9908-9589af362c50" schemeIdUri="9a04f079-9840-4286-ab92-e65be0885f95">
        <cenc:pmsh xmlns:cenc="urn:mpeg:cenc:2013">
          "AAAAVnl3zc2gAAAAAmgTweZhAQqarkLZb4Ihf1QAAA56eAwAAAQABAJQDPABX+=="</cenc:pmsh>
      </ContentProtection>
      <ContentProtection cenc:default_KID="886630ca-f637-4f0c-9908-9589af362c50" schemeIdUri="edef8ba9-79d6-4ace-a3c8-27dcd51d21ed">
        <cenc:pmsh xmlns:cenc="urn:mpeg:cenc:2013">
          "AAAAan3zc2gAAAAA7e+LqXrWSs6jyCFC1R0h7QAAAEOIARIQeS1cb1aNbc7Dji6sAtKZzRoNd21kZXZpbnVf-dGVzdCJfa2V5LWlkOmVTSWNibGFOYnI3RGppNnBdEtoielE9PSoCUQyAA=="</cenc:pmsh>
      </ContentProtection>
      <SegmentTemplate>
        <Representation id="1" width="1280" height="720" bandwidth="5000000" codecs="avc1.64001f; scanType=progressive" />
      </SegmentTemplate>
    </AdaptationSet>
    <Period>
    </Period>
  </Period>
</MPD>
```
How Do I Get Started with SPEKE?

- SPEKE API Documentation: https://docs.aws.amazon.com/speke/latest/documentation/what-is-speke.html
SPEKE Reference Server

- Open source reference key server in GitHub AWS Labs project area
- Foundational example of a custom SPEKE key server, supporting HLS and DASH
- Provides pre-built CloudFormation templates and code for a turnkey installation
- Integrates API Gateway, Lambda, S3, CloudFront, Secrets Manager for key generation
  - Uses secret IV per stream (content ID)
  - Uses key derivation to produce encryption/decryption keys
- Participate at https://github.com/awslabs/speke-reference-server
- Fork the project and build your own key server
- Don’t hesitate to submit issues, questions, pull requests with improvements
Additional Resources:

- DASH-IF Implementation Guidelines: Content Protection Information Exchange Format (CPIX):

- Google Widevine:
  https://storage.googleapis.com/wvdocs/Widevine_DRM_Encryption_API.pdf

- Microsoft PlayReady:
  https://docs.microsoft.com/en-us/playready/

- Apple FairPlay Streaming:
Secure DRM Workflow with AWS Services
with Cognito, CloudFront, Lambda@Edge, S3, and Media services
VOD content encryption flow

1. Enumerate all clear contents needed to be protected

2. Submit Job to create DRM contents

3. SPEKE-compatible DRM providers

4. Store DRM, ABR HLS (and/or DASH)

5. Contents served by CloudFront
Architectural view

WEB CLIENT

1. Sign in
2. Federated Identity
3. Return temporary access credential
4. Fetch CF Key Pair ID for signing from Parameter Store
5. Request to sign URL with Key Pair Id
6. Redirect (302) upon signing completed
7. Content requests must contain signed cookies
8. Direct access to S3 is prohibited

WEB CLIENT

ORIGIN REQUEST

PARAMETER STORE

Amazon SSM

IAM policy with authorized access

FEDERATED IDENTITY

Amazon Cognito

SAML / OIDC

IDENTITY PROVIDER(S)

RETURN TEMPORARY ACCESS CREDENTIAL

PRIVATE ORIGIN

Amazon S3

Amazon CloudFront

DISTRIBUTION

OAID

Federated Identity

Return temporary access credential

© 2018, Amazon Web Services, Inc. or its affiliates. All rights reserved.
Levels of protection

Content
SPEKE compatible DRM encryption

Key Retrieval
API Gateway with IAM_AUTH

Origin
S3 Origin Access ID (OAID)

Content URL(s)
CloudFront Signed Cookies

User
Cognito Authentication
Thank you.

Lionel Bringuier – lbringui@amazon.com
Ken Shek – kenshek@amazon.fr