Dash-Industry-Forum / **dash.js**

Code Issues 191 Pull requests 20 Actions Projects **Wiki** Security

# Generate MPEG DASH content encrypted with MPEG CENC ClearKey

Edit New Page Jump to bottom

Zhongze Tang edited this page on Aug 12, 2019 · 3 revisions

## Prepare our video content

Let's assume our starting point is a regular mp4 file (let's call it video.mp4), and that we will use Bento4 tools for generating the content. The first thing we will need to do is creating a fragmented version of our regular mp4 file:

`mp4fragment video.mp4 video-fragmented.mp4`

## Apply content encryption

Our video file is now ready to be encrypted and packaged as a MPEG-DASH stream. Let's start with the encryption part. To do that, first, you need to define a couple of parameters that will be later used in the encryption process:

- key: Content encryption key.
- kid: Unique identifier of our content encryption key.

Both parameters are defined as 16 bytes and are usually expressed in hex string format. As an example, let's use the ones I defined right below. Use your own ones in for encrypting your content (you can just generate them randomly).

- key: 87237D20A19F58A740C05684E699B4AA
- kid: A16E402B9056E371F36D348AA62BB749

Once defined, let's encrypt our file: `mp4encrypt --method MPEG-CENC --key 1:87237D20A19F58A740C05684E699B4AA:random --property 1:KID:A16E402B9056E371F36D348AA62BB749 --key 2:87237D20A19F58A740C05684E699B4AA:random --property 2:KID:A16E402B9056E371F36D348AA62BB749 --global-option mpeg-cenc.eme-pssh:true video-fragmented.mp4 video-encrypted.mp4`

Short explanation of parameters:

- `--method MPEG-CENC`: Defining MPEG-CENC as the encryption method

- `--key 1:87237D20A19F58A740C05684E699B4AA:random`: using our key to encrypt the track #1.
- `--property 1:KID:A16E402B9056E371F36D348AA62BB749`: using our kid for track #1.
- `--key 2:87237D20A19F58A740C05684E699B4AA:random`: using our key to encrypt the track #2.
- `--property 2:KID:A16E402B9056E371F36D348AA62BB749`: using our kid for track #2.

Note here we are encrypting both video and audio tracks using the same key. You could define a different set of key/kid per track to increase your content security.

## Generate the MPEG-DASH stream

Ok, we have our content ready and encrypted. It is the time of packaging it as a MPEG-DASH stream. `mp4dash -o output video-encrypted.mp4` This will produce the MPEG-DASH packaged version of our file in the `output` folder.

## How to play our stream in dash.js

Easiest path is using dash.js ClearKey example as a reference and change the clearkey configuration to match our own key and kid parameters. To do that, we first need to define our key and key parameters as base64 strings. I used this web based tool to do the transformation but feel free to use your preferred method.

- key: 87237D20A19F58A740C05684E699B4AA -> hyN9IKGfWKdAwFaE5pm0qg
- kid: A16E402B9056E371F36D348AA62BB749 -> oW5AK5BW43HzbTSKpiu3SQ

Note: Please, note you need to remove the '==' padding characters from the resulting base64 transformation.

So, here we have the portion of code that will play our recently generated stream:

```
<!DOCTYPE html>
<html lang="en">
<head>
    <script src="../../dist/dash.all.debug.js"></script>
    <script>
        function init() {
            const protData = {
                "org.w3.clearkey": {
                    "clearkeys": {
                        "oW5AK5BW43HzbTSKpiu3SQ": "hyN9IKGfWKdAwFaE5pm0qg"
                    }
                }
            };
            var video,
                player,
                url = "http://localhost:8080/stream.mpd";

            video = document.querySelector("video");
            player = dashjs.MediaPlayer().create();
            player.initialize(video, url, true);
            player.setProtectionData(protData);
        }
    </script>
</head>
<body>
<div>
    <video></video>
</div>
<script>
    document.addEventListener("DOMContentLoaded", function () {
        init();
    });
</script>
</body>
</html>
```

That's it. Feel free to use these instructions to generate your own MPEG-DASH encrypted content.

+ Add a custom footer

▸ **Pages** 39

## Questions

Please post questions to dash.js Google Group

### Documentation

- Dash.js API Docs
- FAQ
- How to Release Dash.js
- Dash.js 3.0 Migration Doc
- Low Latency streaming

### Samples

View the latest sample players and example implementations.

### Minimum Test vectors

- Smoke test files

### Meeting Minutes

- Archives of our bi-weekly calls

### Background Info

- How to create DASH manifests
- Embedding an adaptive streaming video within your HTML5 application

- Embedding an adaptive streaming video within your HTML5 application
- Building an Open Source DASH-AVC/264 Player
- How to: Creating a DASH-264 Player
- Adaptive Bit Rate video delivery (MPEG-DASH)
- What is MPEG-DASH?

### Clone this wiki locally

```
https://github.com/Dash-Industry-Forum/dash.js.wiki.git
```