# Multimedia Security

DRM-Protected Adaptive Video Streaming Practical Work

2019-2020

In the last years, media distribution has evolved from physical distribution, such as CDs or DVDs to streaming services such as Youtube, Spotify or Netflix. Among other things, streaming services need to adapt the quality of the served content based on the bandwith available. Some platforms, such as Netflix, also adapt the quality of content based on the subsciption plan of their clients.

The Moving Picture Experts Group (MPEG) is a working group of the International Standards Organization (ISO) and the International Electrotechnical Commision (IEC) dedicated to the development and publishing of standards for media formats and distribution, particularly of images (JPEG standard), audio (MP3 standard) and video (MP4). Related to video streaming, the standard MPEG-DASH (Dynamic Adaptive Streaming over HTTP) is an adaptive bitrate streaming technique that enables high quality streaming of media content over the Internet delivered from conventional HTTP web servers. DASH supports the encryption of media via the Common Encryption (CENC) standard, and the World Wide Web Consortium (W3C) standard Encrypted Media Extensions allow the decryption of the media in the browser, with no additional plugins-

The Shaka player is a JavaScript open source DASH player from Google. It plays adaptive media formats (such as DASH and HLS) in a browser, without using plugins or Flash. Instead, Shaka Player uses the W3C open web standards Media Source Extensions and Encrypted Media Extensions.

More information on these technologies can be found on the complemetary reading section of Unit 3 in the moodle course page.

## 1 Task description

This practical work consists on implementing a web service that serves DRM-protected adaptive bitrate video and a website that plays such videos. Use the guide in the **PRACTICAL EXERCISES** section of the course moodle page to implement this. Once you have studied and followed the tutorial you will have to make some modifications to provide the following features:

1. Serve three different videos. You can download them from the following links:

- Elephant Dreams: `https://archive.org/download/ElephantsDream/ed_hd.mp4`. *This is the video used in the guided tutorial.*
- Sintel: `https://archive.org/download/Sintel/sintel-2048-stereo_512kb.mp4`
- Popeye for President: `https://archive.org/download/Popeye_forPresident/Popeye_forPresident_512kb.mp4`

This means you will have to produce three DASH video streams and associated manifests. Note that the *indexTest.html* file in the guided tutorial already provides a way to select the video stream by introducing the *manifest* file name associated to each stream.

2. Protect each of the three streams by **password**. This will require you to produce a different *drm.xml* file for each of the three streams. To generate both the **KID** and **KEY** values for each of the *drm.xml* files, use the Password-Based Key Derivation Function 2 (PBKDF2)[1]. You will have to include this functionality at the client side, too. A JavaScript library that implements PBKDF2 can be found here: `https://gist.github.com/calvinmetcalf/91e8e84dc63c75f2aa53`.

In summary, the client page should include a text field to input the name of the manifest file associated to each of the video streams, a second text field (or password field) to introduce the password to decrypt the video stream, a button to load the video, a bitrate selector (this is already done in the guided tutorial) and the video player itself.

# 2 Evaluation

You have to deliver three items:

- Your working directory (as explained in the guided tutorial) as a .zip file.
- Documentation.

The documentation file must include at least the following items:

- A detailed explanation of the changes made with respect to the results of the guided tutorial.
- The password for each of the video streams.

The evaluation of the practical work will be based on the quality of the delivered items, both software and documentation. Please make your code readable and include any necessary comments, make sure the documentation is clear and complete.

[1] `https://en.wikipedia.org/wiki/PBKDF2`