Research on the Video Copyright Protection

Yiran Guo*

Software engineering, Chongqing University of Posts and Telecommunications, Chongqing, China, 400065
**Corresponding author. Email: 2019213900@stu.cqupt.edu.cn*

ABSTRACT
Nowadays, the phenomenon of online video theft is emerging one after another. Many high-value videos created by video creators are obtained from the network and transmitted again, which makes the video creators suffer losses. The problem of video protection is becoming more and more urgent. Aiming at the field of video protection, this paper introduces how to protect videos through DRM technology. The composition and core steps of a DRM system are described. And through the design of a video-sharing platform, this paper gives the scheme to integrate DRM technology into practical application. The video-sharing platform is developed with the help of interfaces provided by DRM implementer Axinom and Alibaba cloud platform, which supports the core functions of video service, including video storage, video upload and video playback. It is very easy to expand complex business based on it. The video protection effect is evaluated by some common means of obtaining video from the network. Finally, from the perspective of integrating DRM technology into the existing video system, it shows that it is very cost-effective to use DRM technology for some enterprises relying on video services for profit.

Keywords: *Video Protection, DRM, Video Encryption, Video Sharing Platform*

1. INTRODUCTION

There are many possible ways to download videos from the website. Some websites use traditional video formats such as mp4, wmv, flv, etc. The video download address cannot be got directly from the HTML page. Other websites use DASH or HLS. Video download address cannot be got directly from the html page. However, by using the developer tools, the network requests can be captured. There is usually a m3u8 file for HLS or a mpd file for DASH. After getting the index file, some tools can help download the video segments and reorganize them into a mp4 file. Some websites use video encryption. But the encryption key is exposed in the front-end. They can use any encryption algorithm as long as the front-end and back-end are well negotiated. Even though the encryption algorithm is not public, users can still explore the source code and find out the encryption algorithm. Furthermore, the scheme does not prevent users from recording the screen. One popular method for dealing with video recording is to deter users through video watermarking, but this method cannot solve the problem fundamentally. All of these make enterprises relying on video services for profit face great challenges.

Aiming at the problem of video protection, this paper first gives a brief introduction to the mechanism of DRM including five components and three core steps, and then gives the design of a simple video sharing platform using DRM technology which shows the whole workflow of video upload, video storage, video encryption and video playback. This video platform realizes the core functions of video services, which is of great guiding significance for integrating DRM Technology into existing video services. On this platform, the video is encrypted in segments. Even if each segment can be downloaded locally, the video segment cannot be decrypted. In addition, it can prevent users from recording videos using some software and can limit users' playback in virtual machines. This is beyond the reach of traditional video protection technology. The video sharing platform can be accessed at https://videos.cqupt-gyr.xyz/.

2. EXPERIMENT

2.1 DRM Composition

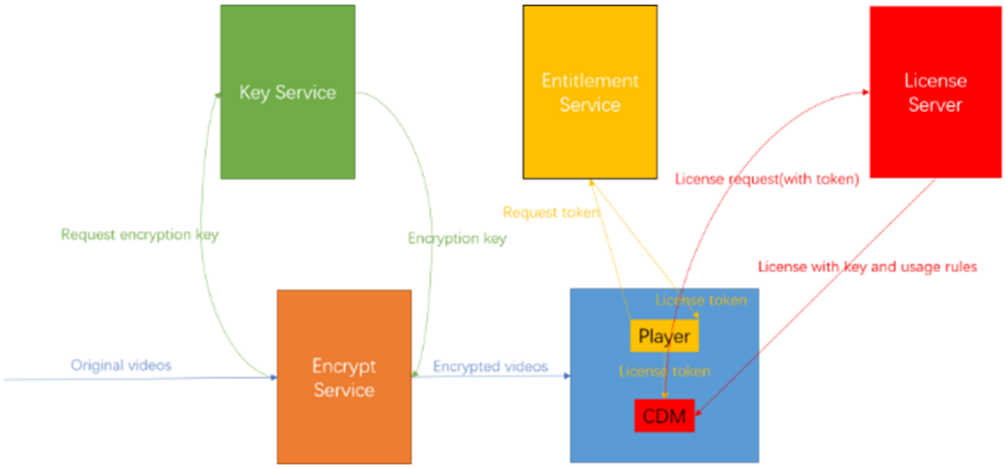

Figure 1. System composition [1]

There are mainly five important components in a DRM system: Encrypt Service, Key Service, Entitlement Service, License Server and Front End (includes player and CDM). The encrypt service is used to encode the video into DASH or HLS to facilitate streaming. At the same time, it will contact the key service to obtain the key and encrypt the video. The key service is used to generate keys for encryption. The entitlement service is used to authenticate the user's identity and sign a token to the front end so that the player can obtain a license from the license server. The license server is used to verify the token posted by the front end and issue DRM licenses to the front end. At the front end, the player plays video via CDM [2].

2.2 DRM Core Steps

2.2.1 Encrypt Video

To encrypt videos, key ID and key are needed. The key ID is used to put in the PSSH (Protection System Specific Headers) [3] and the key is used to encrypt the video. There are many key exchange protocols, such as Anevia, Harmonic, Speke, WidevineProtectionInfo [4]. The key ID and key can be contained by requesting a key server. Then the key and key ID are passed to a packager. The packager will use the key and key ID to encrypt the video. Usually, a mpd file and several m4s file will be outputted.

2.2.2 Authorize Video Play Requests

The authorization of video playback requests is realized through the entitlement service. The entitlement service will sign a token to valid users so that they can play the protected videos. The logic of authentication can be very flexible. A common example is to judge whether a user has logged in to the system. Once it is determined that the user has the permission to play the video, the next step is to issue a token to the user. Usually, a token contains some configuration information, including the effective time of the token, the effective time of the license, and some useful rules of the license (such as whether to allow persistence to the local, the security level of the license and the ID of the decryption key) [5]. Finally, the entitlement service will use the communication key to sign the token to prevent the token from being tampered.

2.2.3 Play Back

There are many players that support DRM, such as Shaka Player, Video.js Player and so on. Different players support DRM in different ways. But there are some steps that all players need. First, the address of the license server has to be configured. The next step is to intercept the request of license and add a specific HTTP request header and put the token issued by the entitlement service into it. The rest of the work is left to the player, which will hand over the obtained license to CDM for video playback.

2.3 The Design of a Video Sharing Platform

This section mainly introduces how to put DRM technology into practical application. DRM technology solves the problems of video encryption. To put DRM Technology into use, it is also necessary to solve the problems of video upload and video storage. The design of a simple video sharing platform based on DRM technology is given where everyone can upload videos and watch videos. This design shows how to combine DRM technology with video upload, video storage and video playback, and gives a general solution for video sharing platforms. The source code of this project can be found at https://github.com/gyr66/drm_server.

The video sharing platform has two main parts: video upload and video playback.

2.3.1 Video Upload

To ensure video playback and upload bandwidth, the uploaded videos are stored in an OSS (Object Storage Service) server instead of a back-end server. The storage space is divided into two buckets. A private bucket, which can only be written, is used to store the original video uploaded by the user; A public bucket, which only allows reading, is used to store encrypted video segments. Video playback will only interact with public buckets, and video upload will only interact with private buckets [6].

On the client side, Browser.js SDK which supports breakpoints retransmit is used to directly upload videos to the OSS server. For security reasons, every upload request must carry a short-term token. The back-end server registers in the RAM (Resource Access Management) server in advance to obtain permission to issue tokens. When the client uploads video, the back-end server first issues a short-term token for it. The client can upload the video to the OSS private bucket only if it has a valid token. To avoid token expiration, the client needs to regularly request the back-end server to update the token.

After the video is uploaded, the OSS server will send a callback request to the back-end server. The back-end server will pull the video just uploaded by the user from the OSS server for encryption. To encrypt the video, the back-end server first needs to request the key and key ID from the key server. Then it will call the packaging tool (in this project, Shaka packer is used) for video encryption. When the encryption is done, the back-end server uploads the encrypted video segments to the OSS public bucket and cleans up the local files. Finally, the back-end server uploads the video metadata to RDS (Relational Database Service) for later queries. The whole upload process is shown in figure 2.

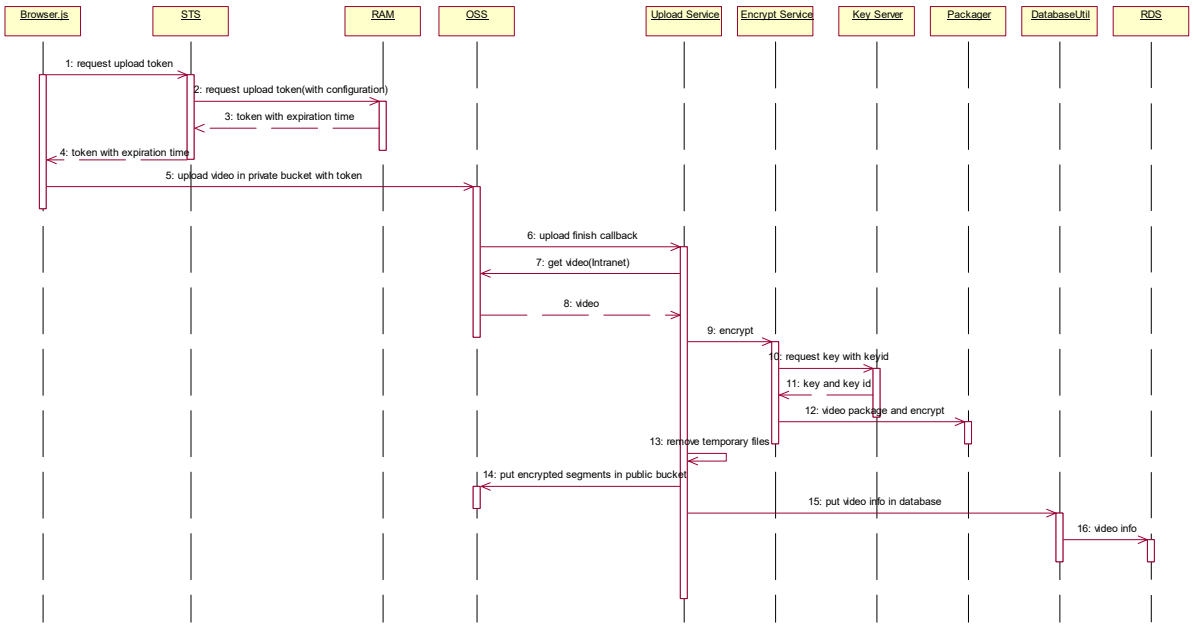

Figure 2. Video upload sequence diagram

2.3.2 Video Playback

When switching to the video playback part, the browser will first make a GET request to the back-end server to get all the information on the videos that have been uploaded before. The back-end server will query the RDS (Relational Database Service) through the database until and pass the results to the browser. The browser then gets the thumbnails from the OSS server and shows the thumbnails on the page. When the user clicks the thumbnail of a video, the browser will request the mpd file of the video and some mp4 files. At the same time, the browser will make a playback request to the token service (located at back-end server) to get a playback token. Later it will request a license from the license server with this token. Finally, it will pass the returned license to the CDM for playback. The whole playback process is shown in figure 3.

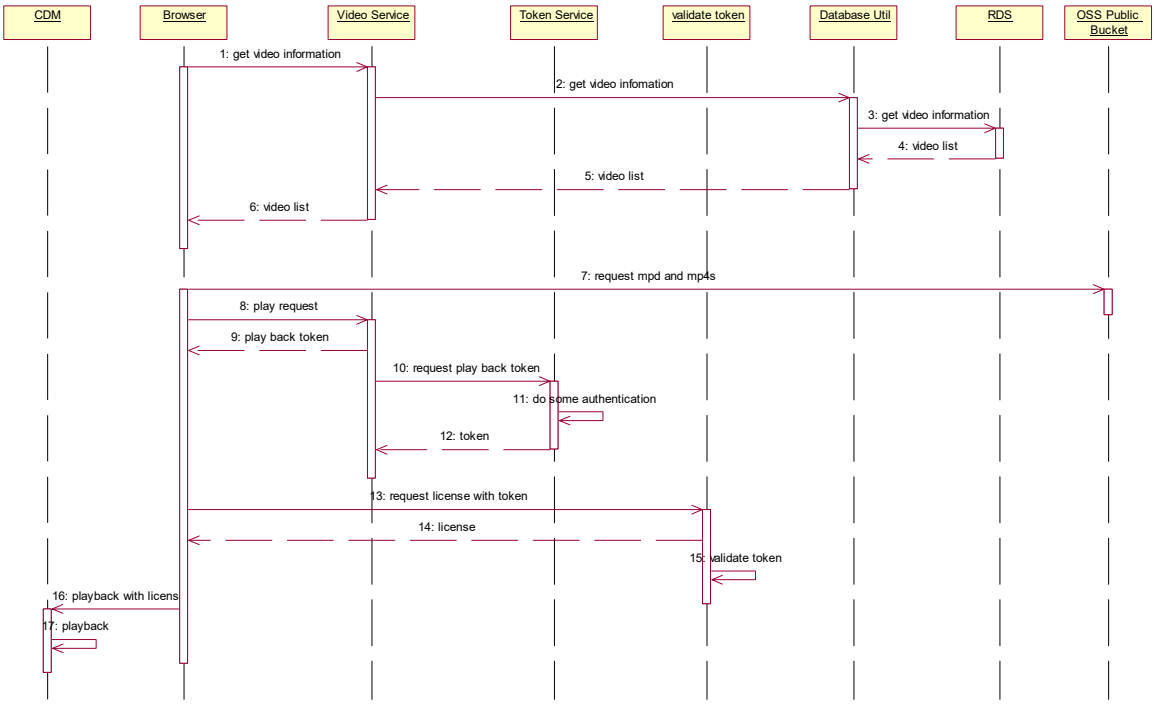

Figure 3. Video playback sequence diagram

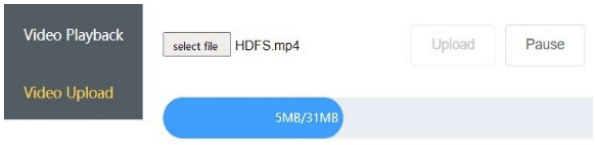

Figure 4. Video upload screenshot

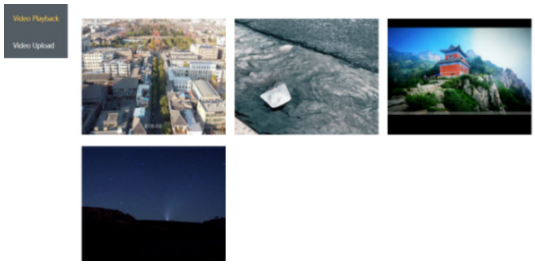

Figure 5. Video playback screenshot

2.4 Evaluation

In this part, the video protection effect of the developed video sharing platform will be evaluated.

For the video platform developed above, the problems mentioned in the introduction can all be well solved. The user can get the license sent by the license server. But the license can only be understood by the CDM. The structure of the license is not public (shown in figure 6). Nobody knows how to crack it except the DRM provider. So even though the user downloads the video, they cannot decrypt it. What is more important is that screen recording can be blocked from the bottom (even in the virtual machine) as figure 7 shows.

Figure 6. The license cannot be decrypted

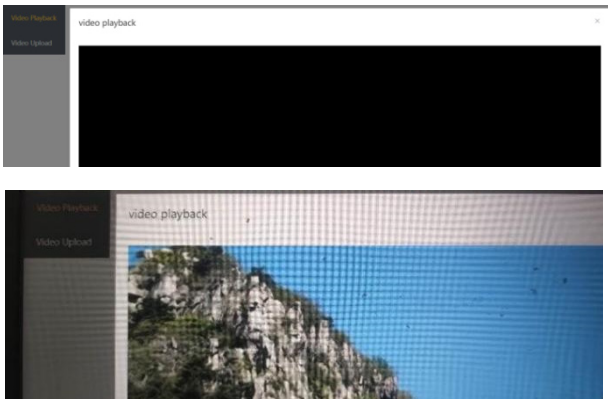

Figure 7. The picture above is captured by screenshot software which is blocked. The picture below is the actual video playback picture taken by camera.

3. CONCLUSION

After evaluating the video sharing platform, it was discovered that using DRM technology to protect video can effectively block some commonly used methods of obtaining videos from the network. However, the implement of DRM technology such as widevine, playready, fairplay is a very time-consuming job. At present, there are three mainstream DRM technologies, widevine, fairplay and playready, which are applicable to different platforms. Video service enterprises need to implement three DRM technologies at the same time to adapt to multiple playback platforms, which is very laborious. Also, the technology is updated very quickly, the implementation is required to be up to the new standard. The best way for video service enterprise is to purchase services provided by DRM implementers. After that, it is very easy to integrate DRM Technology into the existing video system. It only needs to modify the video processing service, the front-end player and provide the service of issuing tokens based on the original authentication service. Therefore, for enterprises that rely on video services for profit, it is very cost-effective to use DRM technology.

Aiming at the short board of video protection of domestic video websites, this paper introduces the composition of DRM technology, and illustrates how to put DRM technology into practical use through the design of a video sharing platform. It has certain guiding significance for how to integrate DRM Technology with existing video services.

REFERENCES

[1] Axinom. What is DRM? https://portal.axinom.com/mosaic/documentation/drm/what-is-drm, 2022-03-11/2022-03-27.

[2] James Irwin, Digital Rights Management: The Open Mobile Alliance DRM specifications, Information Security Technical Report, Volume 9, Issue 4, 2004, Pages 22-31, ISSN 1363-4127, https://doi.org/10.1016/S1363-4127(05)70037-6.

[3] Pramod A. Jamkhedkar, Gregory L. Heileman, Digital rights management architectures, Computers & Electrical Engineering, Volume 35, Issue 2, 2009, Pages 376-394, ISSN 0045-7906, https://doi.org/10.1016/j.compeleceng.2008.06.012.

[4] Matthew Peterson, 6 - Digital Rights Management, Editor(s): Matthew Peterson, In QuickTime Developer Series, Interactive QuickTime, Morgan Kaufmann, 2004, Pages 51-57, ISBN 9781558607460, https://doi.org/10.1016/B978-155860746-0/50010-7.

[5]google. https://developers.google.com/widevine/drm/overview, /2022-03-27.

[6] Alibaba cloud. https://help.aliyun.com/document_detail/32069.html, 2021-01-27/2022-03-27.