PlaySafe: A Digital Rights Management System for Media Content Consumption

Calvin Yee Keen Lau and Fang-Fang Chua(✉)

Faculty of Computing and Informatics, Multimedia University, Cyberjaya, Malaysia
ffchua@mmu.edu.my

Abstract. Digital right management (DRM) is widely used by online streaming service provider that allows them to manage and control the access rights of their copyrighted content. The existing DRM technology requires the developer to gain approval from the DRM provider company in order to access to their proprietary Software Development Kit and tools. This issue had caused some software developers not being able to get their approval to implement the DRM technology as the DRM's proprietary Software Development Kit and tools are not publicly available and often causing the delay of their software release. A DRM system, PlaySafe, is proposed as an alternative DRM system for copyrighted content management. The objective of this research is to design and develop a DRM system to facilitate media file encryption and decryption processes. The requirements elicitation and gathering process from the documentations provided by the existing DRM technology provider is being used as the requirements for the newly proposed DRM system. PlaySafe provides the content provider with features such as encryption, key rotation, automated HLS streaming packaging, and set streaming quality constraints which are cross platform, self-hosting supported and open source. The proposed backend system runs on a modular Docker container with virtualization which allows the content provider to host the system on their own local machines. Lastly, PlaySafe is an open-source project which allows anyone with the right knowledge to modify and add extra feature that they need since the source code is accessible by the public.

Keywords: Application Programming Interface (API) · Digital Rights Management (DRM) · AES-128 · Encryption · Decryption

1 Introduction

Digital content piracy includes but not limited to video, song, movie, eBook, research article, software, game or anything that is available online as a soft copy. As for the entertainment industry that produces songs and movies, the piracy issue regarding this industry has been skyrocketed ever since the covid-19 pandemic has started as most people have been staying at home the whole time and has plenty of free time to watch movies and listen to music [1]. Forbes has mentioned that there has been more than 40% traffic increase on websites that shares pirated media contents [1]. Moreover, piracy in the entertainment industry has costs those studio companies to lose about $29 billion

S.-C. Haw and K. Sonai Muthu (Eds.): CITIC 2022, 10, pp. 171–181, 2022.
https://doi.org/10.2991/978-94-6463-094-7_14

USD of revenue in a year and piracy is also one of the factors that causes an estimated of 230,000 jobs lost in the entertainment industry [2].

Popular media streaming services and the entertainment industry uses Digital Rights Management (DRM) technologies to combat with piracy. The Digital Rights Management technology is designed to control and limit the usage of any proprietary hardware and copyrighted content.

The workings of the DRM technologies are based on the concept of encryption. In a nutshell, the encryption algorithm works by scrambling the data bits until it becomes unreadable, this process is known as encryption. Next, only the entity that has the corresponding decryption key can decrypt the data bits back into its original forms. The DRM technologies such as Widevine, PlayReady, and FairPlay will do so by encrypting all the content and store it onto a server and only the authorized and eligible user can decrypt the content for consumption. The most widely used DRM technology by all the streaming services nowadays are Google's Widevine, Apple's FairPlay and Microsoft's PlayReady. The Widevine DRM are mostly being used in Google's products such as their Chrome browser, Android OS and Chromecast devices, meanwhile the FairPlay DRM are only available to all Apple product.

All the DRM technology in the streaming industry uses sophisticated methods to securely stream the content to the user's device from their content server, and these methods usually requires the device's Original Equipment Manufacturer (OEM) to embed special security module or processor into the device itself [3]. All software will need to get approved by the DRM providers to playback the content that are being protected using the mentioned DRM technologies. Next, upon getting approval from the DRM providers, the developers are allowed to integrate the proprietary code to allow their software to playback those DRM protected content. The approval process sometimes could cause a third-party independent developer to scrape away the whole project just because of the failure of getting approved by the DRM provider [4].

Digital right management (DRM) are widely used by online streaming service provider and DRM solutions should be easy and hassle free to implement by any software developers. However, the DRM solutions provider in the market requires device and software approval from the DRM solutions provider in order for the developer to get access to their proprietary Software Development kit (SDK) and tools to be used in the implementation. As some developers might not get their software approved, this had caused the software developers to have blockers to implement the DRM solutions into their own software as their proprietary SDK and tools are not publicly available and often caused the delay of their software release.

The objective of this proposed work is to firstly identify the media content consumption behaviors and digital rights management requirements. The requirements will be used to design and develop a Digital Rights Management System to facilitate media file encryption and decryption processes. PlaySafe practices cross platform, self-hosting supported and is open source. The cross platform support allows the content provider to host the system on their own server despite with different kinds of operating system installed.

2 Literature Review

2.1 Media Content Consumption Behaviour

The covid-19 pandemic has significantly changed the way on how the Americans consumes movies and the all the cinemas' sales and revenue had been declining rapidly as more people watch movies at their own home due to the new stay at home norms being enforced by the government [5]. Ever since the movie studios and productions company had seen a decline of revenue and sales at the cinema, they begin to develop their own direct-to-consumer media publishing and distribution services which is known as streaming service platforms. Moreover, many production studios are forced to release their movie directly to consumer through streaming services as based on the survey done by Deloitte, only 18% of Americans had consumed movie at a cinema ever since the pandemic had hit [5]. According to CNBC, a new data from a piracy tracking firm Muso had shown a 43% spike in the visit of movie pirating site among Americans as this is all due to the government enforced lockdown and people had been consuming movie over the internet. Moreover, Walt Disney had witnessed that the number of illegal downloads that occurs for the release of their movie titled "Mulan" had outpaced most of the other Disney released movie.

Peukert et al. [6] has collected and constructed a rich dataset from Boxofficemojo which provides statistic data regarding commercial movie industry. Since the Megaupload file hosting provider does not provide a user interface that allows users to browse through a list of uploaded contents, they have to look for a link portal which provides content meta-data and links to download the contents. Peukert et al. [6] collected information from Megavideo website on whether a movie was available for download have found that 60% of the majority movies provides link to the Internet Movie Database (IMDB) website which provide user with movie meta-data like cast, critics, awards, and trailers [6].

2.2 Encryption with Advanced Encryption Standard (AES-128-CBC)

Encryption is a process of converting plain text file into a scrambled text file which converts the readable file into gibberish. The process of decryption is to convert the gibberish file back into its original form with an corresponding encryption key that had initially used to encrypt the file. There are two types of encryption which are symmetric and asymmetric encryption. The symmetric encryption requires a single key to encrypt and decrypt a file, whereas an asymmetric encryption requires a public key to encrypt a file and a secret key to decrypt a file [7]. The advantage of using symmetric encryption is that it provides a much faster speed compares to asymmetric encryption.

The National Institute of Standards and Technology (NIST) has selected the Advanced Encryption Standard (AES) which are known by the original name as Rijndael to be used as the data encryption standard to be used by the government to securely store and encrypt their data. The AES encryption uses a symmetric key to both encrypt and decrypt a file [8]. The AES is a block cipher where it encrypts the data in the form of 128 bits block of data at once at a time. The Fig. 4 shows the workflow of the AES

algorithm where a plaintext or data with the size of 128 bits is being XORed with the initial 128 bit round key to produce the output for SubBytes ().

The SubBytes is a stage where the data gets transformed into an array of hexadecimals and then mapped into a new final state array by using the S-Box table. Next, the output from the SubBytes stage will be passed into the Mix Columns stage where the matrix multiplication process will be performed onto the previously generated output. The constant matrix will be used to multiply against the previous final state array output to form a new matrix. After the Mix Column stage, the cipher key for that round will be added to the state array generated from the previous state by using XOR and if this is the last round of key round, the after result of adding the round key to the state array will be taken as the result of a cipher text. Besides, to make sure that the encryption cipher text is randomized and unpredictable each time when the same plaintext data is being encrypted, the AES Cipher Block Chaining mode will be used to ensure that each time when the same plaintext data is being encrypted, the ciphertext output will not be the same as previously generated output.

2.3 Feature Comparison with Existing DRM Technologies

3 existing DRM technology providers have been reviewed (as shown in Table 1). These DRM technology are all being used by most major streaming services to allow their media content being securely delivered to the legitimate end user. All of the DRM including the proposed PlaySafe DRM solutions comes with encryption key rotation feature which replaces the encryption with a new key periodically as this is to prevent the key from being used to decrypt the content in case the key got leaked or stolen. AES-128-CBC encryption algorithm are being utilized by all of the DRM solutions as the main advantage for AES encryption algorithm is speed as it can both encrypt and decrypt the data significantly faster than RSA encryption algorithm.

Widevine DRM [9] provides a packager tool namely Shaka-packager but it only supports transmuxing the video and not transcoding; FairPlay DRM [10] will require the content be packaged using its provided tools and no 3rd party tools are allowed, and as for PlayReady DRM [11], it does not provide any packaging tools but only the specifications and requirements on how the video should be encrypted.

PlaySafe comes with an automated HLS stream packager which the user only needs to submit the video to the server and the server will automatically package the video. Besides, all of the DRM technology supports set constraints on streaming quality as it allows user to stream the content with quality constraints in place. FairPlay DRM [10] is not cross-platform supported as it is a proprietary DRM solution from Apple which mainly reserved only for Apple made devices. All three of the existing DRM technology is not open-source and does not support self-hosting license key and content delivery server due to security concern. These DRM technology provide the user with advanced and high end protection features as it uses hardware level encryption as compared to the proposed PlaySafe technology which only uses software to simulate the encryption process. In comparing with software level encryption, hardware level encryption is more difficult to temper. For the software level encryption, it is comparably easier for extraction and encryption key prediction which makes it less secure than hardware level encryption. The self-hosting content delivery server and key management server comes with some

Table 1. Features comparison between DRM Technology

Features	DRM technology			
	Widevine [9]	FairPlay [10]	PlayReady [11]	PlaySafe
Key rotation support	✓	✓	✓	✓
Encryption algorithm	AES-128-CBC	AES-128-CBC	AES-128-CBC	AES-128-CBC
Automated HLS stream packaging	Requires manual video packaging with provided tools but still require 3rd party tools for transcoding	Requires manual video packaging with provided tool	Does not provide any tools or packager but rather provide documents and specifications regarding video packaging	Supports Automated HLS Stream packaging
Set streaming quality constraint	✓	✓	✓	✓
Account login/registration	✓	✓	✓	✓
Stream encrypted Content	✓	✓	✓	✓
Cross-platform	✓	☐	✓	✓
Open-source solution	☐	☐	☐	✓
Public API	☐	☐	☐	✓
Support self-hosting backend DRM server	☐	☐	☐	✓

compromise in terms of security. If the security parameters such as the file access control are not being implemented properly, the exposed encryption key has a higher risk of allowing the key from being stolen by a malicious entity.

3 Proposed Solution

The iOS streaming application has 5 core components which are MainView, ViewController, MediaPlayer, ViewModel and NetworkService. Figure 1 shows the overall architecture of the PlaySafe DRM system.

The main processes for content streaming consists of authentication, license key request and content fetching. The application will first request an auth token from the authentication server by sending the user's identity information and generate an auth

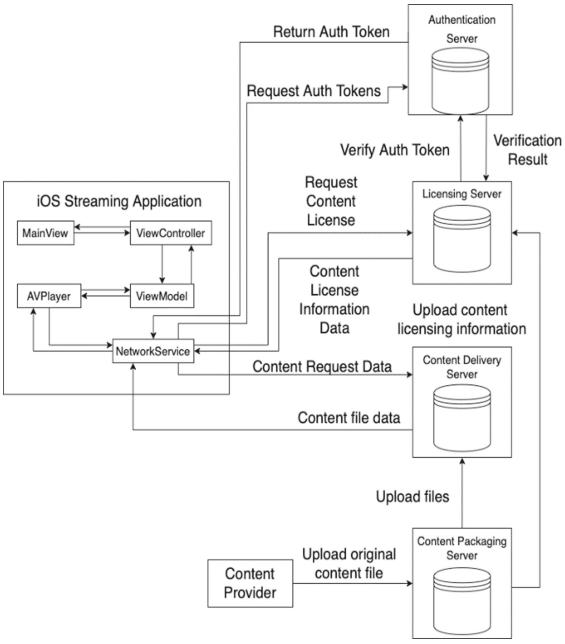

Fig. 1. PlaySafe overall architecture

token for the valid user. The application embeds the auth token into the license request and then sends the license request to the licensing server, and the licensing server will verify if the licensing request is valid and send the content license data back to the application. Next, the application will fetch the encrypted content file from the content delivery server, the content will be decrypted using the decryption key embedded in the content license data. Furthermore, the content provider will upload their media content through the content packaging server which will then utilize the Automated HLS stream packing process. The automated HLS stream packaging process works by applying the transcoding and transmuxing process onto the original video file uploaded by the client.

The transcoding process is a process where the video will get resized into 6 different resolutions by default which are 144p, 240p, 360p, 480p, 720p and 1080p, and the transcoding library used are libx264 which comes together with the ffmpeg which is an open-source tool that contains a suite of libraries and programs for processing video, audio, and other multimedia files and streams. Subsequently, the transcoded video will then go through a compression process by using the ffmpeg tools which it compresses the video into different bitrate that are suitable for each resolution. After the compression is done, the output result will have 6 different videos with different quality being generated, and each of the video will go through the transmuxing process.

During the transmuxing process, the video file will be converted in small chunks of HLS file format for streaming. Then, a random encryption key will be generated by the openssl tool. Next, ffmpeg will use the generated key to encrypt the chunked HLS file. Once the encryption is done during the transmuxing process, the ffmpeg will then

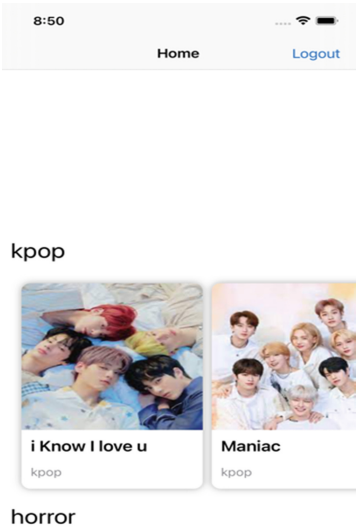

Fig. 2. Home screen page

generate a master playlist file which contains the metadata on how to play the stream, which will be read by the player on the client side. On the client side, the iOS streaming application will then use the master playlist URL to make request to the server to fetch the master playlist file in order to start the playback. The content of the master playlist URL contains the location of the chunked HLS file which allows the player to download the chunked data into its buffer memory while the playback is ongoing. The master playlist also contains the information required to decrypt the video whereby the encryption metadata are encryption method used to encrypt the video, URL address to retrieve the decryption key from the server, and initialization vector for decryption process use. The iOS streaming app uses Apple's AVPlayer library to stream the encrypted content on server, and before the player attempts to stream and decrypt the content, an authorization header will be injected into the player's HTTP request header so that the server will return the key for decryption.

4 Implementation Results

The iOS PlaySafe Streaming Demo Application allows the users to stream and browse media content on their devices. As shown in Fig. 2, the users can browse for all the media content that is available to them and they can tap on the content that they want to consume.

Figure 3 shows that the application allows the user to stream contents. The content that is being streamed in the screenshot are encrypted, meaning that it requires an encryption key from the server to decrypt the stream before seeing any video or hearing any audio

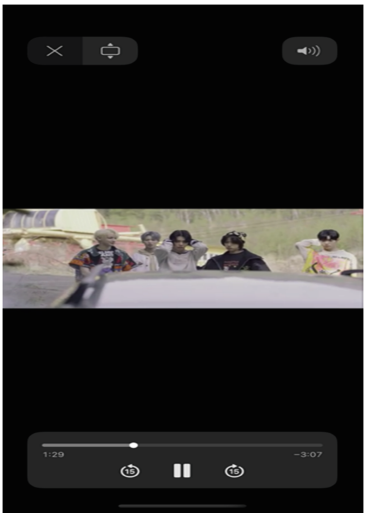

Fig. 3. Stream content page

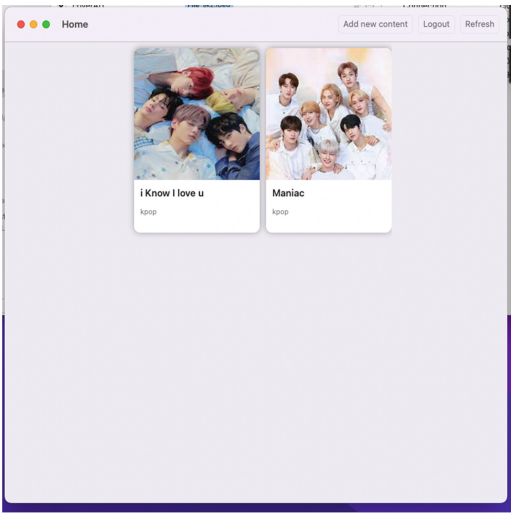

Fig. 4. Home screen for PlaySafe desktop app

song. The content provider PlaySafe desktop application allows the content provider to create, upload and manage their media content. The content provider will need to register an account with the content provider privilege to use the application. Figure 4 shows the home page for the content provider to view all the media content owned and uploaded by themselves, and they can also logout and add new content to the list by clicking the button from the toolbar.

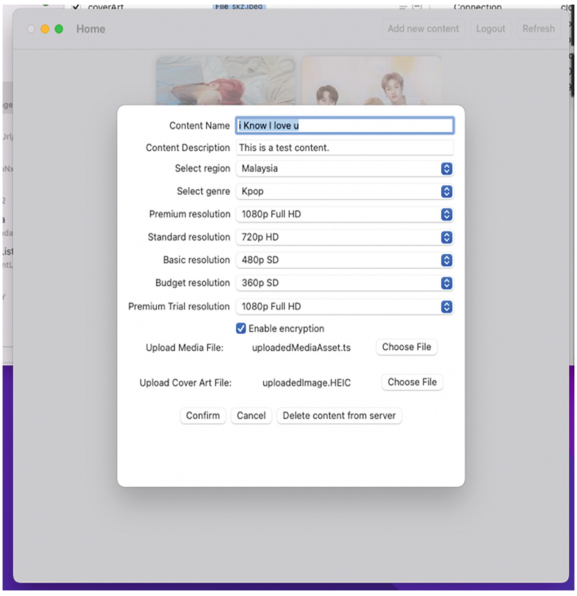

Fig. 5. Edit media content metadata

Figure 5 shows that the content provider can edit the metadata of their content by clicking on the media content to be edited. Moreover, they can also set the streaming constraints for different type of user packages.

5 Conclusion

In conclusion, the proposed work helps to identify the usage and purpose of Digital Rights Management in the entertainment industry where it is being used to ensure that the copyrighted content will not easily be duplicated illegally without permission. Besides, the media content consumption behaviors of consumers have been identified during the covid-19 phase which is significantly different from the pre covid-19 phase where all the cinemas had been closed temporarily and most people have been streaming movie and media contents online. The main motivation for the proposed PlaySafe DRM system is that the DRM solutions provider in the market requires device and software approval from the DRM solutions provider in order for the developer to get access to their proprietary Software Development kit (SDK) and tools to be used in the implementation. As some developers might not get their software approved, this had caused the software developers to have blockers to implement the DRM solutions into their own software as their proprietary SDK and tools are not publicly available and often caused the delay of their software release. The limitation of PlaySafe is that the automated HLS stream packaging process is slow and utilizes large amount of memory resources.

The future work includes exploring methods to improve the speed and efficiency in terms of system resources. Next, since the user authorization with JWT token feature in PlaySafe uses JWT token to authorize the user, an authorization header which includes

the JWT token secret is sent together with the HTTP request through the internet without any form of encryption being applied onto the request header. This vulnerability allows the attacker to perform attacks such as man-in-the-middle attack to intercept the HTTP request and to extract the JWT token which can then be disguised as an authorized user for malicious use. This could further be improved by applying asymmetric encryption to the HTTP request before sending it through the internet so that only the legit entity can decrypt and read the content.

References

1. Escandon, R. (2020, April 27). *Forbes*. Retrieved from Film Piracy Has Been Skyrocketing As People Stay Home: https://www.forbes.com/sites/rosaescandon/2020/04/27/film-piracy-has-been-skyrocketing-as-people-stay-home/?sh=6746f7ae7c81
2. Gonzalez, O. (2019, June 18). *Cnet*. Retrieved from Digital video piracy costs movie and TV industry at least $29 billion a year, study says: https://www.cnet.com/tech/mobile/digital-video-piracy-costs-the-movie-and-tv-industry-at-least-29-billion-study-says/
3. Triggs, R. (2019, June 14). *Android Authority*. Retrieved from Widevine digital rights management explained: https://www.androidauthority.com/widevine-explained-821935/
4. Claburn, T. (2019, April 3). *The Register*. Retrieved from No Widevine DRM for you! Developer left with two years of work stymied by Google snub: https://www.theregister.com/2019/04/03/googles_widevine_drm/
5. Arkenberg, C. (2020, December 1). *Deloitte*. Retrieved from Digital media trends The future of movies: https://www2.deloitte.com/us/en/insights/industry/technology/future-of-the-movie-industry.html
6. Peukert, C. (2017). Piracy and box office movie revenues: Evidence from Megaupload. *Science Direct* (p. 189 - 193). Frederiksberg: Science Direct
7. [7]Jonsson, J., & Kaliski, B.S. (2003). Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1. *RFC, 3447*, 1-72
8. Information Technology Laboratory (National Institute of Standards and Technology). (2001). *Announcing the Advanced Encryption Standard (AES).* Gaithersburg, MD: Computer Security Division, Information Technology Laboratory, National Institute of Standards and Technology
9. Google. (2017). *Google Widevine.* Retrieved from Google Widevine: https://developers.google.com/widevine
10. Apple. (2016). *Apple FairPlay.* Retrieved from Apple FairPlay: https://developer.apple.com/streaming/fps/
11. Microsoft. (2015). *Microsoft PlayReady.* Retrieved from Microsoft PlayReady: https://www.microsoft.com/playready/documents/