streeme

DRM Overview

CONTENT HANDLING GUIDELINES

Table of Contents

1. GENERAL DESCRIPTION

The Streeme IPTV platform uses a number of mechanisms for protection of the content, users' data, and the system itself:

1) **Source code encryption.** All software modules of the platform are installed with encrypted binaries and source code parts in order to prevent hacking, reverse engineering, and unauthorized modifications.

2) **Data encryption.** All valuable data, such as passwords of the users, are stored in encrypted form. Moreover, even the system administrator can't decrypt these passwords and read them. There is only a possibility for the system administrator to send a request to the user to set the new password.

3) **Command encryption.** All requests/responses and other commands between system parts are transmitted over secure HTTPS protocol with AES-128 encryption in order to prevent interception and modification of transactions between parts of the platform.

4) The platform actively uses **protection mechanisms** provided by the **software framework**, which was used for implementation of the web-interface, and **security mechanisms** provided by the **database platform**.

5) **IP-filtration.** Access to the servers, where the content is being transformed and stored, is restricted for certain IP-addresses only, which prevents possibility to obtain direct access to the origin-servers by end-users.

6) Integrated **DOS/DDOS-attacks preventions** mechanism allows the system to stay in service and prevent attempts to compromise security of the platform by the key or password generation techniques.

7) **Logging** of all activities on two independent levels: by the OS means and by internal logging system of the Streeme IPTV/OTT-platform.

8) **Monitoring** of the state of **hardware resources**, which allows detection of suspicious activities of the processes consuming CPU resources or network traffic.

9) **Monitoring** of the **users' activities** including unusual network traffic consumption, such as sessions with more than average duration (uninterrupted watching for more than the average watch-time).

2. CONTENT PROTECTION FOR HLS WITH AES-128 ENCRYPTION

With the increase of piracy, protecting media content is one of the key concerns of many publishers. In this section, the most popular method for content protection with the HTTP Live Streaming (HLS) protocol AES-128 content encryption is described.

2.1. Content Protection

The need for content protection has been recognized by many different streaming protocols, which have added support for content protection in various forms and flavors. AES-128 encryption has been present in the HLS specification from the first draft of the protocol, putting content protection high on

the priority list. In fact, there are two encryption schemes which are supported by HLS:

- AES-128 encryption: This means media segments are completely encrypted using the Advanced Encryption Standard with a 128-bit key. It also allows for the usage of Initialization vectors to optimize the protection.
- Sample-AES: In this case, the individual media samples are encrypted using the AES-standard. With this encryption level, the stream container is not fully encrypted. Also, how the encrypted samples are encapsulated, depends on the media format of the segment.

In practice, AES-128 is the most commonly used method for HLS encryption. This method is also often the easiest to achieve using standard streaming servers and tools.

2.2. Safety of the AES-128 Protection

The first question when dealing with content protection is often: "How safe is this protection?". In order to understand this, let's look at what AES-encryption really is. AES is a symmetric encryption algorithm. It was designed to be efficient in both hardware and software. The algorithm is used worldwide and was adopted as the standard encryption algorithm by the U.S. government for encrypting sensitive data. Furthermore, it is the basis of most of the DRM systems available, for example Microsoft PlayReady, Widevine, and Verimatrix. The usage of AES encryption recently became part of the common encryption standard for MPEG-DASH as well. In general, it might be safe to say this level of AES encryption will not be broken soon.

The AES encryption itself can be declared safe. However, encryption is only as safe as its weakest point. It is also necessary to have a look at the security of the decryption key. This is the area on which many DRM technologies focus. They deem key protection essential and often employ very obscure or complex schemes to retrieve decryption keys. With AES-128 content protection, key retrieval has been kept simple, making it easy to implement. It also leaves plenty of freedom to make key protection as simple or advanced as possible.

2.3. Protection of the Decryption Key

The HLS specification mentions only one aspect of key retrieval: the URL from which the key can be loaded should be a part of the manifest file. Protecting this resource is up to the publisher itself. Most often, we see a number of different approaches to protecting the decryption key:

- Protecting the manifest: This relies on hiding the URL to the decryption key. It does not provide a high level of security as the URL might leak or could be intercepted on the network.
- Using authentication cookies: Authentication cookies can be sent by the player with the key request. This allows the key server to check which user is requesting the key. If the user is not allowed to access the stream, the key will not be returned. As a result, only users which have proper authentication will receive the decryption key.

- Leveraging signed URLs: Signed URLs can be used by providing unique manifests to each user. A user-specific manifest will then contain a link to the decryption key, containing an authentication token. The server can then check the authentication token and determine if the key can be accessed, or not.

2.4. Using AES-128 Encryption in Practice

Using AES-128 encryption can be done by encrypting your media files and signaling this using the EXT-X-KEY-tag within the manifest file. This tag signals the URL to the decryption key. It should be placed before the first segment, which is encrypted with the given key. There are two extremes in which this tag can occur:

1) One time on top of the manifest. This means all segments are encrypted with the same decryption key. In case the decryption key is intercepted, the entire stream can be decrypted.

2) Before each segment with a different URL. This approach allows encryption of each segment with a different key. A key allows the decryption of a single segment, which contains only a few seconds of media information.

Between these two extremes, there is a freedom to choose the frequency of refreshing the encryption keys required for each individual case. By default, the Key Distribution Server generates the new key every 24 hours.

3. DRM ALGORITHM DESCRIPTION

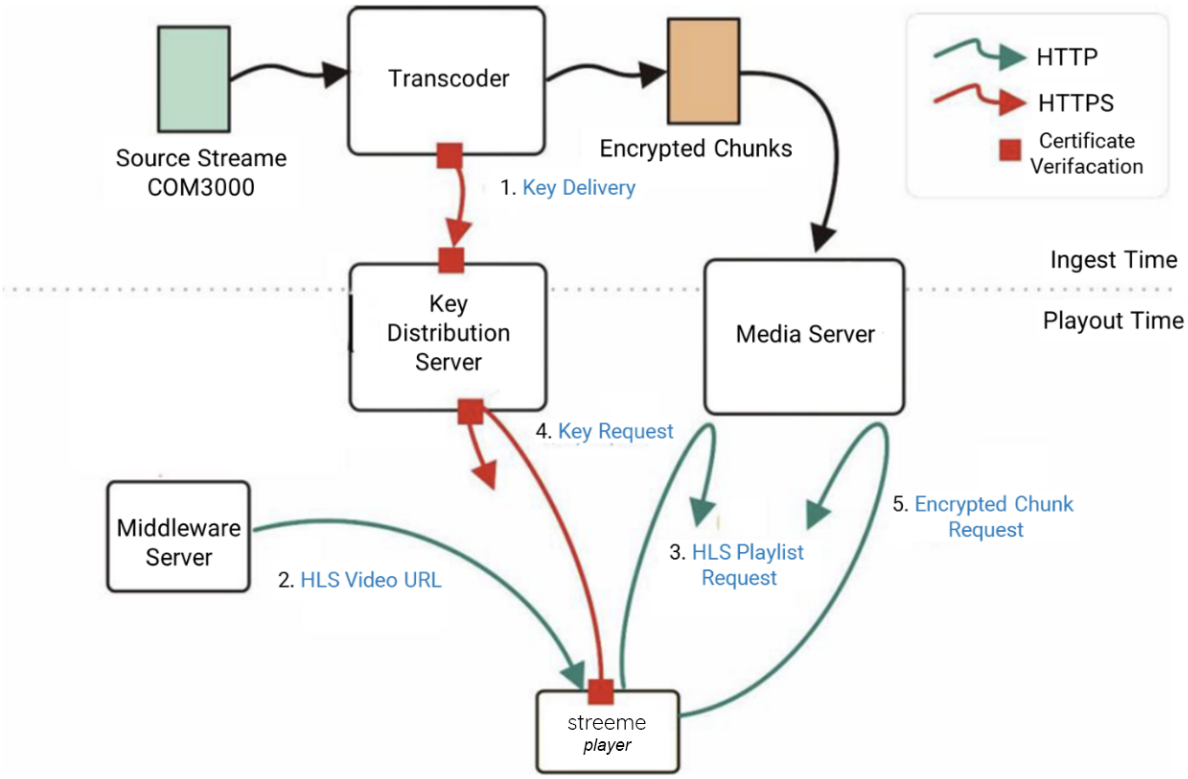

Figure 1- DRM functioning algorithm model

1) Streeme transcoder divides incoming content into 10 seconds parts (blocks), encrypts the blocks with AES algorithm, and delivers the encryption key securely to the Key Distribution Server.

2) Streeme middleware server provides the end user's player application with an HTTP URL to desired video stream.

3) Streeme Player application requests the M3U-playlist from the Streeme Media Server using the URL provided on the previous step. The Media Server includes a X-EXT-KEY URL in the playlist, which points to the Key Distribution Server. Fetch of the playlist can be made over secured HTTPS protocol for additional protection.

4) The player application requests security certificate from the Key Distribution Server in order to prevent a man-in-the-middle attack. After successful certificate verification the player requests decryption key for the content from the key distribution server over secured HTTPS protocol. The key distribution server requests and verifies the client's certificate in order to identify a rogue client, which is trying to obtain the decryption key. In case of successful verification, the server returns the key for the asset; otherwise, the server rejects the incoming request.

5) The player requests the encrypted video blocks from the media server. The player can then decrypt the content and play it.

The key distribution server has a list of IP-addresses of all encryption servers, which are eligible to request the keys. This list of IP-addresses is configured upon the system installation and updated by validated personnel (administrator of the IPTV platform) upon the system expansion. Additionally, the encryption server and the key distribution server validate each other by checking SSL-certificates upon establishing a secure HTTPS-connection.

The administrator of the IPTV platform must pass the two-steps authorization mechanism and validate the connection to the encryption server by his email.

Each channel on the encryption server is processed with the unique encryption key. Moreover, the same channel on different encryption servers will be encrypted with different keys. This means that even if a decryption key for a certain channel will be compromised, this will not affect the other channels processed at the same server and will not affect the same channel on the other servers.

By default, encryption keys are changed every 24 hours. This interval may be changed by the administrator of the IPTV platform according to requirements of the content provider.

Each request for content by the player is validated by the mechanism of tokens. The token is generated based on the following parameters:

- Master token, which is provided by the key distribution server
- Unique ID of the user, which is assigned automatically when user registers in the IPTV
- Platform Type of content, live or VOD
- Name of the content
- Expiration time, the default value is 24 hours
- IP-address of the user, if IP-filtering mechanism is enabled

4. DOS/DDOS-ATTACKS PREVENTION

In order to prevent DOS/DDOS-attacks, the IPTV Platform controls the source and the frequency of incoming requests. Each request, which may be handled by the Platform, has a number of allowed attempts per 60 seconds. If the number of allowed attempts will be exceeded, then the IP-address, from which these attempts were made, will be blocked for a certain period of time.
Additionally, there is a possibility for the system administrator to create a white-list and a black-list of IP-addresses to control the input traffic and robustness of the Platform.

5. USERS AUTHENTICATION AND IDENTIFICATION

The system supports the following techniques that guarantee secure identification of users and their devices:

Users are authenticated by unique log-in/password combination, which is transmitted from player application to the middleware server over secure HTTPS-protocol with AES-128 encryption. Additionally, users may be authenticated in the platform by their device ID, which is read by the application from device's firmware and can't be modified by an end-user or by some other means.

There is also a limit on the number of simultaneously connected devices from one user account.

6. DISTRIBUTION PLATFORM

6.1. Geo-filtering Solution

The IPTV/OTT Platform uses "DB-IP.com" service for the geo-filtering functionality. The service's back end is updated every 10 days depending on the volume of pending updates, and the database downloads are updated monthly usually on the 1st day of each month.

7. END USER DEVICES

7.1. DRM Solution Robustness

The platform does not store any content on end-user devices except for the minimum amount of 10-30 seconds, which is required for playback.

Content is transmitted to device being encrypted by the DRM system with AES-128 algorithm and only an authorized player application can decrypt it at the time of playback. If the device is tampered with and chunks of content remain in the device's memory they will not be played by 3rd-party players unless properly decrypted.

7.2. Identification of End-User Devices

Each player application reports a unique ID of the device it's running on into the middleware at the time of user login. The ID is read from the firmware of the device and cannot be accessed by an end-user.

7.3. Detection of Clone Devices

Each player application reports the ID of the device it's running on to the Middleware at the time of user login. The ID is read from the firmware of the device and cannot be accessed by the end-user. In any case, we do not accept redundant device IDs.

8. DVR, DOWNLOADS AND TRANSFER TO DEVICES

Content protection from downloading and/or transferring to another device:

- No capability to download and transfer content to another device;
- No storage use in the device for content;
- No capability to download and transfer content to another device.

9. NETWORK/TRANSMISSION TECHNOLOGY

The transmission medium(s) employed for signal delivery:

- Twisted-pair copper;
- Coax;
- Fiber;
- Wireless network.

Protocols used to deliver content to the consumer – HTTP and HLS.

10. CONTENT STORAGE RULES

Default buffering on the player side: 10-30 seconds as per HLS protocol specifications.
Default buffering on the server side: about 1 minute.
Optional encrypted storage may be configured in the platform to store 1 or more hours for pause, rewind, and playback of live channels. Storage space for the catch up service is allocated only in the head-end server.

Optional Network Video Recorder (NVR) service may be configured in the Platform in order to provide end-users with the ability to record desired TV programs for later viewing. Storage space for the NVR service is allocated only in the head-end server.

11. BREACH RESPONSE

Initial communication between the player application used by the customers and the IPTV platform is performed via secure HTTPS protocol, which assures correct authentication of the users. Users are recognized in the platform by their login/password combination, which may be replaced by their social network account. After that, the player application then reads the unique device identifier from the firmware and sends this ID to the core of the IPTV platform. This ID cannot be modified by the users and assures the identity of the device used to access the IPTV service. The platform has an option to limit the number of simultaneously connected devices from the same end-user account so that users cannot have multiple set-top boxes or other devices to obtain access to many different channels simultaneously, which would look like an attempt to steal and illegally redistribute content. Moreover, links to the origins of content are never transmitted to the player application therefore users can't get these links and put them into 3rd party applications for illegal content consumption.

Security of the platform is constantly monitored by the maintenance team. In case of security alert the team will work by the following general scenario:

1) Take immediate measures to stop the breach, ban suspicious user account and/or filter network traffic coming from/to suspicious IP-address;
2) Take immediate measures to restore availability of the service for other end-users;
3) Analyze nature of the breach and discover the core of the issue;
4) Depending on the results of analysis at the previous step develop and apply patches and updates required to cure the breach and prevent similar attacks in future.

11.1. Communication Process to Notify IPTV Operator of any Breaches

If for some reason the IPTV platform will be hacked or illegally used by unauthorized users, then appropriate a ticket for the incident will be created in internal CRM system and the Operator will be notified about the case by the Operator's personal project manager or technical engineer. There is a tickets support sub-system integrated in the IPTV platform, which allows Streeme customers to create and monitor progress on all incidents. Streeme assigns the highest priority to all issues and tickets regarding any security matters. Streeme assigns appropriate technical resources to track and resolve such issues immediately, which might require a work in close connection with the Operator's local personnel until complete issue resolution.