Watermarking Adaptive Video Streams By Manipulating m4s Segments

Kalinga Swain
Concordia University, Montreal

Jonathan Lupague
Concordia University, Montreal

ABSTRACT

Intellectual property integrity is constantly in danger due to digital piracy; hence, creative countermeasures are essential. Conventional watermarking approaches are frequently insufficient against skilled pirates who employ sophisticated techniques. This research introduces a groundbreaking technique to embed binary watermarks in the form of messages during the live ingestion of video streams from servers. This research presents a system developed to encode live binary watermarks seamlessly into .m4s chunks, enhancing the security of digital content without compromising quality. The accompanying decoder, which can be implemented on the player side, facilitates the retrieval of the embedded message during playback. This multifaceted methodology not only bolsters content protection against the prevalent issue of piracy but also establishes superior resilience against removal or distortion techniques. By embedding binary watermarks dynamically during live ingestion, our approach introduces a layer of adaptability that contributes to the robustness of the protection mechanism. This comprehensive approach aims to fortify content protection in the face of rampant piracy, providing superior resilience against removal or distortion techniques. The effectiveness of this watermarking technique is rigorously benchmarked against existing methods across diverse scenarios, indicating its potential as the groundwork for future innovations in this field.

Author Keywords

Integrity; Privacy; Piracy; Authentication

CCS Concepts

•**Privacy** → **Authentication;**

INTRODUCTION

In the contemporary digital landscape, the preservation of intellectual property faces an escalating threat posed by the rampant issue of piracy. The ubiquity of unauthorized redistribution, exemplified by pirates capturing live streams and rebroadcasting them on unauthorized platforms, not only results in substantial revenue loss for content creators but also dilutes the overall viewing experience for legitimate audiences.

CHI'20, April 25–30, 2020, Honolulu, HI, USA

ISBN 978-1-4503-6708-0/20/04…$15.00

DOI: https://doi.org/10.1145/3313831.XXXXXXX

Compounded by tactics such as deep linking and sharing on social media platforms, pirates exploit direct links to live streams, enabling unauthorized access and exacerbating the challenges of content control.

Furthermore, the monetization of piracy introduces an additional layer of complexity. Pirates capitalize on their activities by charging users for access to pirated live streams or by integrating advertisements into unauthorized content, compounding the financial impact on content creators and distributors. This escalating threat demands innovative countermeasures that transcend the limitations of conventional approaches.

Comparatively, live streaming piracy presents unique challenges in monitoring and combating illicit activities due to the real-time nature of the content. Pirates adeptly adapt to countermeasures, necessitating effective and dynamic solutions to safeguard digital assets. The financial implications of both live streaming and Video-on-Demand (VOD) piracy are substantial, directly impacting subscription and advertising revenues. However, the immediacy of live streaming piracy poses a distinct challenge to user engagement, as viewers seek free alternatives in real-time.

Dynamic Adaptive Streaming over HTTP (DASH)

Dynamic Adaptive Streaming over HTTP (DASH), also known as MPEG-DASH, is an adaptive bitrate streaming technique that enables high quality streaming of media content over the internet delivered from conventional HTTP web servers. Here's an overview of its background, implementation, and recent works:

Background

DASH was developed by the Moving Picture Experts Group (MPEG) and was published as an international standard in 2012. It is the first adaptive bitrate HTTP-based streaming solution that is an international standard. The development of DASH was motivated by the need for a universal standard that could be used across different devices and platforms, as prior to DASH, there were various proprietary adaptive streaming technologies such as Apple's HLS (HTTP Live Streaming), Microsoft's Smooth Streaming, and Adobe's HDS (HTTP Dynamic Streaming).

The key advantage of DASH is its ability to adapt the quality of the video stream in real-time, according to the available network conditions and the capacity of the client device. This adaptability minimizes buffering and ensures an optimal user experience.

Implementation

DASH works by dividing the content into a sequence of small HTTP-based file segments, each containing a short interval of playback time of content such as video, audio, or other data. These segments are served over HTTP and can be hosted on any web server. Here are the basic components of a DASH implementation:

Media Presentation Description (MPD): An XML document that describes the location of media files, available qualities, segment durations, and other metadata necessary for playback.

Segments: The actual multimedia files, which are divided into small, multi-bitrate chunks of a few seconds each. These can be in any codec/container that HTTP supports (like MP4 or WebM).

DASH Client: The player or device that requests the segments based on the MPD file and the user's current network conditions.

The DASH client dynamically requests the appropriate segment with the suitable bitrate, enabling it to adapt to changing network conditions while minimizing the risk of buffer underrun.

Recent Works

Recent works in DASH focus on optimizing performance, improving user experience, and integrating cutting-edge technologies. Some of these advancements include:

1. Machine Learning Machine learning algorithms are being explored to predict network conditions and user behavior to optimize segment requests.

2. Low-Latency Streaming Efforts are being made to reduce the latency that is typical with segment-based streaming to enable real-time applications like live sports and interactive services.

3. Multi-CDN Switching

4. To improve reliability and performance, DASH players can switch between different CDN (Content Delivery Network) providers during a streaming session.

5. Content Protection New methods for integrating DRM (Digital Rights Management) within the DASH ecosystem are being developed to protect copyrighted content.

6. VR and 360° Video DASH is being expanded to support immersive video formats, requiring new ways to handle the significantly larger file sizes and bandwidth demands.

7. Hybrid Delivery Combining DASH with broadcast techniques like ATSC 3.0 to offer a seamless experience across different delivery methods.

8. Standards Convergence There is ongoing work to harmonize DASH with other streaming standards like CMAF (Common Media Application Format) for broader compatibility and efficiency.

DASH remains at the forefront of streaming technology due to its adaptability, efficiency, and broad industry support. Research and development continue to improve DASH, ensuring it meets the evolving demands of content providers and viewers.

STATEMENT OF THE PROBLEM

In light of these challenges, the imperative to fortify content protection mechanisms is evident. The introduction of invisible watermarks emerges as a pioneering solution, poised to address the deficiencies of conventional approaches. This research aims to implement an innovative technique, seamlessly embedding binary watermarks in the form of messages during the live ingestion of video streams, bolstering content protection without compromising quality.

Motivations

The motivation behind this endeavor lies in the multifaceted benefits that invisible watermarking brings to the forefront. By introducing this adaptive technique, the following objectives can be achieved:

1. Tracking and Attribution:
   - Invisible watermarks provide a covert means to track the origin of unauthorized content, enabling swift attribution and targeted counteraction against piracy.
2. Legal Action:
   - The embedded watermarks serve as indisputable evidence in legal proceedings, empowering content creators to pursue legal action against individuals or entities engaged in unauthorized distribution.
3. Preserving Content Integrity:
   - The integration of invisible watermarks safeguards the integrity of digital content, ensuring that viewers experience it as intended by creators, free from the distortions introduced by piracy.

EARLIER WORKS

1. Low-on-Latency-plus (LoL+)

   The paper introduces "Low-on-Latency-plus (LoL+)," an enhanced solution building upon the earlier "Low-on-Latency (LoL)" approach for low-latency live streaming applications. LoL+ addresses limitations in LoL by incorporating dynamic features, improving adaptability, and ensuring robust deployment. Three key areas of improvement are highlighted: parameter adaptability, accommodation of diverse use cases, and refinement of adaptive playback speed control.

   The motivation stems from the increasing demand for low-latency services in live streaming applications like Twitter's Periscope and Amazon's Twitch. As traditional solutions such as Adobe's RTMP fade away, HTTP adaptive streaming (HAS) using DASH and HLS dominates the market. However, the target for low-latency streaming is five seconds or less, presenting unique challenges in the streaming workflow.

The proposed LoL+ solution leverages the Common Media Application Format (CMAF) standard and HTTP/1.1 chunked transfer encoding (CTE) to reduce latency. It introduces sophisticated player enhancements, including a bitrate selection module, playback speed control module, and accurate throughput measurement module tailored for CTE. These collectively address challenges in throughput measurements, bitrate selection, buffer management, and playback speed adaptation.

Key contributions include the formulation of the weight selection problem as an assignment problem, the design of a learning-based adaptive bitrate (ABR) algorithm adapting a Self-Organizing Map (SOM) model, and the implementation of LoL+ integrated into the official dash.js player (v3.2.0). The solution is validated through trace-driven experiments, showcasing its superiority over four state-of-the-art ABR algorithms for low-latency live streaming.

In conclusion, LoL+ is positioned as a specialized player for low-latency streaming, delivering good Quality of Experience (QoE) for various latency targets. The open-source availability and integration into dash.js highlight its potential contribution to other player implementations, fostering further advancements in the field of low-latency live streaming.

2. Steganography

Background

Steganography in live streaming involves the concealment of information within the audio, video, or other data streams to hide the existence of the embedded content. The primary goal is to make the presence of the hidden information imperceptible to viewers while allowing authorized users to extract or decode the concealed data. In the context of live streaming, this practice may have various applications, including copyright protection, content authentication, and data embedding.

Recent Works

Some notable trends involving steganography in live streaming include:

(a) Audio and Video Steganography Techniques: Ongoing research explores advanced audio and video steganography techniques, aiming to embed information seamlessly into multimedia content without compromising perceptual quality.

(b) Machine Learning-Based Steganalysis: The development of machine learning algorithms for steganalysis continues to evolve. Researchers are working on robust methods to detect hidden information in multimedia streams, which could impact the effectiveness of steganography techniques.

(c) Real-Time Steganography: Considering the real-time nature of live streaming, recent works might focus on developing steganography methods optimized for low latency and high-speed data embedding to keep pace with the streaming content.

(d) Deep Learning Applications: Integration of deep learning approaches in steganography and steganalysis is an area of active research. This involves leveraging neural networks to enhance the security and resilience of hidden information.

(e) Dynamic Embedding Techniques: Recent works might explore dynamic steganography techniques that adapt to changing streaming conditions, making the detection and removal of hidden content more challenging for adversaries.

The design for "Watermarking Adaptive Video Streams By Manipulating m4s Segments" presents an innovative approach built on the foundation of "LoL+" (Low-on-Latency-plus). Notably, this novel watermarking technique incorporates elements of steganography to embed information within adaptive video streams by manipulating MPEG-DASH (.m4s) segments. This integration with LoL+ underscores the adaptability and extensibility of the original low-latency solution, showcasing its versatility in supporting advanced functionalities like steganographic watermarking in the context of adaptive video streaming.

WATERMARKING ADAPTIVE VIDEO STREAMS BY MANIPULATING M4S SEGMENTS

Architecture

Figure 1 illustrates the architecture and flow of a low-latency video streaming system, that is similar to the achitecture of LoL+ implementation.

Here's a step-by-step explanation of the different components and how they interact:

1. OBS Studio This is the source of the input feed, likely a video capture or streaming software that sends out the live video feed.

2. ffmpeg transcoder and ll-dash packager ffmpeg is a widely used software for handling multimedia data. It's being used here to transcode the video into a format suitable for streaming. The 'll-dash' packager then takes this transcoded video and packages it for the DASH (Dynamic Adaptive Streaming over HTTP) protocol, optimized for low latency (ll).

3. Encode Script When encoding is being done [using the input video being split into smaller chunks], a new encode script will add a binary message to the video chunks. This binary message will be used later when the chunks are received into the client side.

4. LL-DASH toolchain This represents the entire process involving the ffmpeg transcoder and ll-dash packager, ensuring the video is in the correct format and packaged for streaming.

5. node-gpac-dash This component use a node.js implementation of GPAC's DASH streaming (where GPAC is a multimedia framework).

6. LL-DASH stream This is the resulting video stream, now in a format ready for low-latency delivery over the internet.

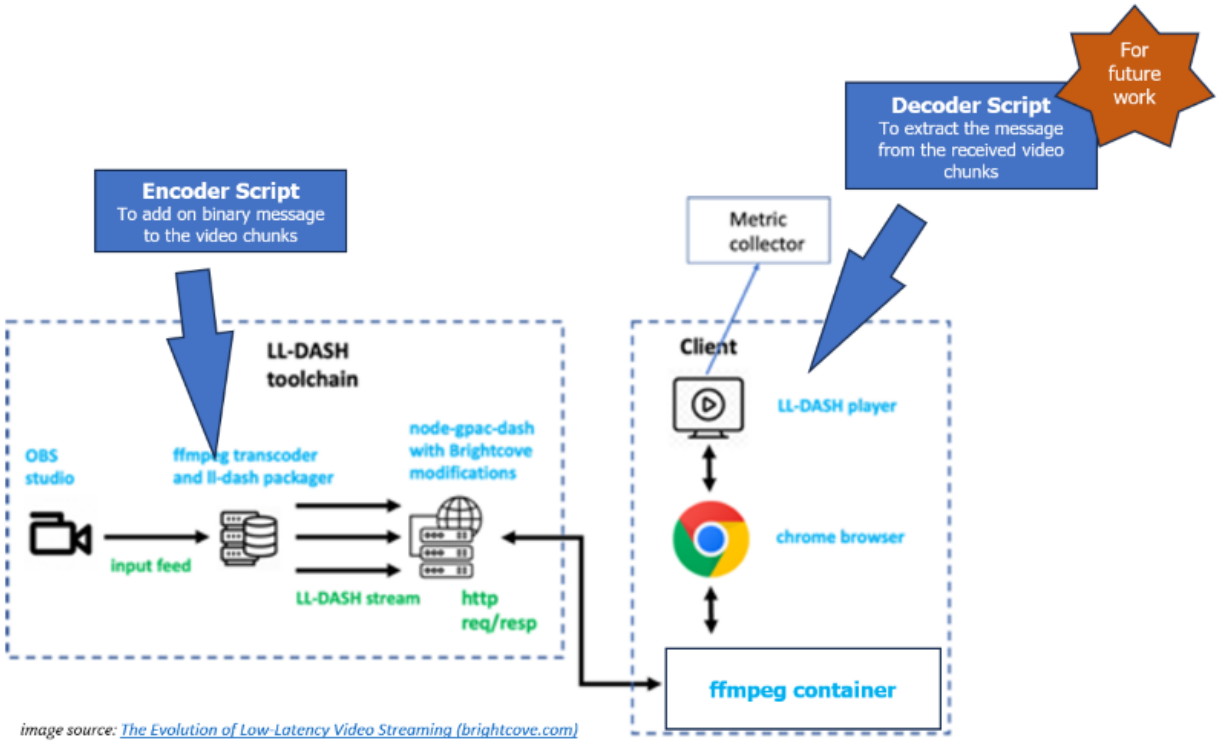


Figure 1. Generic architecture of LL-DASH toolchain and the client. In blue are the enhancements for Watermarking Adaptive Video Streams By Manipulating m4s Segments, where Encode and Decoder scripts are added in the encoder and player, respectively.

7. HTTP req/resp Represents the standard protocol for the client's browser to request (req) and receive (resp) video data.

8. ffmpeg container This is an isolated environment (like a Docker container) where the ffmpeg process runs.

9. Client LL-DASH player The client-side player, which is responsible for playing the LL-DASH stream. It likely has capabilities to handle low-latency streaming effectively.

10. Decoder Script Paired with the Encoder Script, this script on the client side extracts the message from the received video chunks. The purpose is to check the integrity of the video chunks received. The binary message extracted by the decoder script should be the same as the binary message used by the encode script. [1]

11. Metric collector This component collects various metrics regarding the streaming quality, user experience, etc., which can be used for analytics and improving the service.

The arrows indicate the flow of data through the system, from the initial video capture to the end-user playback. The dashed lines suggest that the Metric collector and the Decoder Script are not directly in the stream's path but are ancillary services that support the streaming system. The 'For future work' label next to the Decoder Script suggests that this component is planned for future implementation or will be enhanced.

[1]Note that in this paper and within the testing done, the team has only build the decoder script but has not fully integrated with the client side, i.e., the dash.js player.

Algorithm

As earlier stated, 2 new scripts were developed for this paper. Figure 2 provides the pseudo-code of both the encode and decoder scripts.

For the purpose of testing, the binary message used was simple, e.g., simple string which was converted into binary representation by the encode script, then was added (along with a Marker into the video chunks) and were written into different folder that will be used for encoding (into different bitrate-resolution representations).

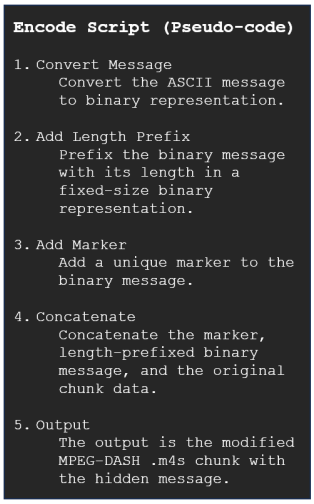
In the decode script, each chunk videos received will be checked for the presence of the Marker and the binary message.

In relation to this, some of the more challenging features are to be added in the future, which will be tackled in the next section. Some of these future works are:

- Integrate the decode script into the player
- use a sophisticated binary message (AI/ML generated message, or by using RSA encryption)

CONTRIBUTIONS

The research presented in the paper introduces a pioneering approach to enhancing the security of live video streaming without compromising on performance. By incorporating digital watermarks—essentially added messages within the video stream—the system ensures a higher level of security against unauthorized use or redistribution. One of the key achievements of this method is that it adds no extra overhead to the



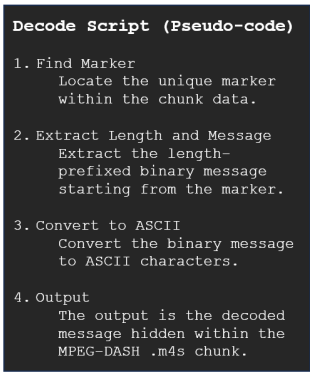


Figure 2. New scripts to (1) add binary message to each chunk video; and (2) to extract the binary message from each chunk videos received by the player.

streaming performance, maintaining the efficiency of the service even with the security enhancements. Furthermore, the solution is designed to be fully compatible with existing Adaptive Bitrate Streaming (ABR) systems, which are widely used for delivering content that can dynamically adjust to changing network conditions. This compatibility ensures that the proposed system can be integrated seamlessly into existing streaming infrastructures. Notably, the paper claims that this is the first system of its kind according to the latest available information, positioning it at the forefront of innovation in secure live streaming technology. Preliminary designs and testing have demonstrated the viability of this approach, marking a significant advancement in the field.

FUTURE WORKS

For the continuation of the research presented, the paper outlines critical areas of future development to enhance the robustness and security of live streaming technology further. A primary goal is to integrate the decoder into the client-side player. While the preliminary groundwork has been laid with the creation of code capable of decoding messages from video chunks, subsequent efforts will focus on embedding this decoding logic into the player itself. This integration aims to enable seamless, real-time decryption of the digital watermark during video playback, ensuring a secure and uninterrupted viewing experience.

In addition to integrating the decoder, the research team aims to elevate the security measures by encrypting the digital watermark. Prospective strategies include the application of established encryption algorithms such as AES or RSA. These cryptographic standards would make unauthorized access to the embedded messages significantly more challenging. Furthermore, there is a proposition to harness the potential of artificial intelligence or machine learning to generate dynamic, sophisticated messages that vary over time or per session. This would add an additional layer of complexity to the encryption, deterring potential breaches with continuously evolving security measures. The implementation of these advanced encryption techniques represents a forward-thinking approach to safeguarding digital content in the realm of live video streaming.

REFERENCES

Catching the Moment With LoL+ in Twitch-Like Low-Latency Live Streaming Platforms
Authors:

- Abdelhak Bentaleb, Member IEEE
- Mehmet N. Akcay, Senior Member, IEEE
- May Lim, Senior Member, IEEE
- Ali C. Begen, Senior Member, IEEE
- Roger Zimmermann, Senior Member, IEEE

Steganography-Tools
Author:

- Priyansh Sharma

Video-Steganography
TEAM AEGIS:

- Tan Kae Chuan | @itxKAE
- Mirza Haziq Bin Mohammed Yusoff @m1rzyq
- Muhammad Syaifulnizar Bin Izam | @syaifulnizarrr
- Christopher Gwee Soon Chai | @ManOCoolture
- Ong Xing Hao | @XinhaoSITICT