ADJUSTING BIT-STREAM VIDEO WATERMARKING SYSTEMS TO COPE WITH HTTP ADAPTIVE STREAMING TRANSMISSION

Antoine Robert, Omar Alvarez, and Gwenaël Doërr

Technicolor R&D France – Security & Content Protection Labs
E-mail: firstname.lastname@technicolor.com

ABSTRACT

HTTP adaptive streaming has become the dominant solution for broadband streaming delivery. Essentially, the server delivers small chunks of content in response to queries from a player, that can alternate between different qualities of content depending on its rendering capabilities and the network condition. In this paper, we highlight that the frequent switches of this popular delivery mechanism are deemed to interfere with the payload modulation of bit-stream video watermarking systems. We then propose a generic strategy to cope with such transmission and showcase its efficiency in practice with a watermarking system that operates directly in the H.264/AVC CABAC bit-stream. Eventually, we evaluate the robustness of the proposed solution against standard attacks.

Index Terms— HTTP adaptive streaming, traitor tracing, watermarking.

1. INTRODUCTION

Only a decade ago, video content was primarily intended to be watched on TV sets in living rooms. Today, content consumption is expected to be possible on virtually any device with rendering capabilities, possibly on the go. These rendering devices may have various characteristic, e.g. with respect to screen resolution, CPU power, battery life, Internet connectivity, etc. The challenge is then to devise delivery mechanisms that can serve such huge variety of devices while relying on the same distribution infrastructure.

There are essentially three alternative approaches to tackle this issue: scalable video encoding, dedicated RTP/RTSP media servers, and HTTP Adaptive Streaming (HAS) [1]. HAS essentially builds on top of HTTP, the standard Internet protocol, and thereby inherit several of its features. First, it does not require a specific port address and thus seamlessly go through most firewalls. Second, the native caching capabilities of the HTTP infrastructure permits to save a sizable amount of bandwidth. Third, HTTP being built on top of TCP/IP, lost data packets are retransmitted, allowing guaranteeing a given quality of service.

On the server side, video content is encoded at different bit-rates and resolutions in order to accommodate for the diversity of end-devices as well as various network conditions. Moreover, the content is also split in short segments, e.g. 2 to 10 seconds, to be able to dynamically adjust the quality of the video upon request. In other words, a single movie can be seen as a collection of $Q \cdot S$ files, where Q and S stand for the number of qualities and segments respectively. These files can be stored independently (*physical* files) or wrapped in a single container (*logical* files). In the latter case, the server needs to generate the requested chunks on the fly and thus slightly deviates from a standard HTTP server. The HAS server also hosts for each content a playlist file, referred to as the *manifest*, that lists all available chunks for this content [2].

	Adobe HDS	**Apple HLS**	**Microsoft IIS**	**MPEG DASH**
Video codec	H.264, VP6	H.264	H.264, VC-1	H.264 + others
Audio codec	AAC, MP3	AAC, MP3	AAC, WMA	AAC + others
Container	MP4/FLV	TS	MP4	MP4/TS
File type	logical	physical	logical	logical/physical
Manifest	.F4F	.M3U8	.SMIL	.PDM
Player	Flash, Air	iOS, QuickTime	Silverlight	MPEG players
Protection	Flash Access	AES-128	PlayReady	flexible
Chunk duration	2-4 sec	10 sec	2-4 sec	flexible

Table 1. Main features comparison of major HAS technologies.

One baseline principle of HAS is that the server should be as simple as possible, ideally reduced to serving requested files, and that all intelligence should be moved to the client side. To stream a video, the end-device therefore first retrieves the corresponding manifest file and starts requesting chunks, based on its rendering capabilities. At any moment, the player can decide to switch to a different quality for subsequent requested chunks. It is the responsibility of the player to initiate such switches, e.g. relying on an analysis of its internal buffer that kind of reflects the network conditions.

The economics and scalability of HTTP delivery largely outweigh the additional encoding and storage costs induced by HAS. As a result, HAS is gradually emerging as a *de facto* standard in the broadband environment and several technologies have been released over the last few years, including Adobe's HTTP Dynamic Streaming (HDS), Apple's HTTP Live Streaming (HLS), Microsoft's Smooth Streaming (IIS) [3], and 3GPP/MPEG DASH. Table 1 summarizes some key differences between them with respect to some HAS features.

While this new streaming mechanism holds great promises for the consumers' experience, it may introduce undesired side effects for underlying technologies e.g. digital watermarking. In Section 2, we briefly review prior work and exemplify how the HAS quality switching mechanism comes into the way of the payload modulation in bit-stream watermarking systems. We then introduce a generic framework that can be used to fix existing algorithms and accommodate for HAS transmission. In Section 3, we apply the proposed paradigm to a video watermarking system that operates directly in the H.264/AVC CABAC bit-stream and showcase the validity of the proposed approach through various robustness tests. Eventually, we summarize the findings of our study and outline directions for future work in Section 4.

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

2. HAS-OBLIVIOUS BIT-STREAM WATERMARKING

Forensic watermarking refers to serving clients with uniquely watermarked content in order to facilitate traitor tracing, if a copy is later found on an unauthorized distribution network. Today, this mechanism is routinely employed in professional environment, e.g. for digital cinema, for screeners, for premium video-on-demand services, etc. In anticipation of potential extensions relying on HAS transmission, it may be necessary to revise the serialization system depending on the underlying watermarking technology.

2.1. State-of-the-Art

Prior work [4] focused on modifications at the server side to deliver serialized copies to the end-device. The main idea is that the server now has, for each video chunk, several pre-watermarked versions associated to alternate payloads. When a device requests the manifest file for a content, the server produces a client-specific manifest that discards all chunks that do not encode the identifier of the client. As a result, whatever is the HAS transmission path across the different qualities, the resulting sequence of chunks encode the ID of the client. While this strategy is backward-compatible with legacy players, it incurs significant storage overhead, both at the server side and in the network through the Internet caching mechanism.

The dual approach consists in serializing content at the client side. This transition may be seamless if the watermarking process operates at the signal level. However, such algorithms are usually computationally intensive which may be undesired on resources-constrained mobile devices. This limitation motivated the recent introduction of new watermarking algorithms that directly operates into the bit-stream [5, 6, 7, 8].

2.2. HAS and Bit-stream Watermarking

Such watermarking systems are typically decomposed in three steps. To begin with, the content is analyzed (i) to pinpoint positions where the bit-stream could be modified and (ii) to identify an alternate value, e.g. a few bytes, that could be used instead of the original one. This pre-processing step typically includes some partial decoding to check the induced distortion and discard all modifications that would not respect a specified fidelity-robustness trade-off. The resulting sequence of triplets (`offset`, `origValue`, `altValue`) is then formatted, e.g. to specify over how many changes a payload bit is spread and/or to integrate a pseudo-random sequence that will guide the substitution convention in the last step. Eventually, the original bit-stream and the formatted metadata are forwarded to a substitution engine to actually embed the watermark payload.

Since the whole watermarking process is dependent on the original bit-stream, it is necessary to forward the pixel-domain equivalent of the triplets (`offset`, `origValue`, `altValue`) to perform watermark demodulation. Such information can for instance translate as a frame index in the video, a block index in the frame, and some original/alternate value for a feature derived from this block, e.g. the average luminance value. For such demodulation to operate successfully, even after modifications of the video, the tested video needs to be perfectly synchronized in space and time with the original master file. This can be achieved for instance using content fingerprints [9, 10]. In other words, detection is semi-blind which is acceptable only in the traitor tracing application use case.

To watermark a HAS-mastered content, one could apply the baseline watermarking algorithm to all the bit-streams that the HAS content is composed of. However, previous work reported that the throughput of bit-stream watermarking systems is tightly related to the content mastering parameters, including the encoding bit rate [11]. In other words, the number of embedding changes per second will be different for each quality of the HAS master. As illustrated in Figure 1, the switches across the different qualities are deemed to disrupt the payload modulation convention and the detector is no longer able to extract the embedded watermark.

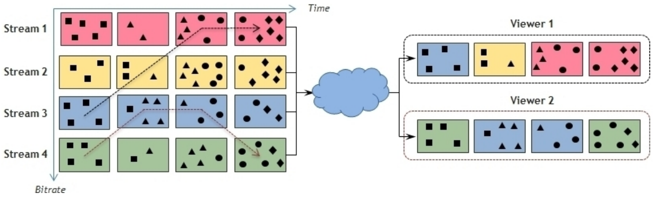

Fig. 1. HAS-induced payload modulation disruption. The content is encoded at different bit rates and segmented in temporally aligned chunks. The pre-processing step identifies embedding locations for each stream and the formatting step sets the payload modulation convention, e.g. five changes per payload bit in this example. During HAS transmission, the player requests chunks at different qualities. The re-assembled content no longer respects the payload modulation convention and the watermark cannot be extracted.

2.3. Harmonization of the Watermarking Throughput

The payload modulation disruption originates from the fact that the pre-processing step yields different watermarking throughputs at different qualities. In other words, the payload is embedded more or less quickly depending on the encoding bit-rate. To alleviate this issue, an intuitive solution is to harmonize the watermarking throughput across the different qualities. For instance, there should not be a problem anymore if all temporally aligned chunks share the same number of embedding locations.

This can be achieved by incorporating a module before the formatting step. Its role is (i) to derive some HAS-compliant statistics, based on the outputs of the pre-processing engine applied to all bit-streams, and (ii) to feed the formatting process with this information to adjust its behavior. For example, this HAS-supervision module could look at the number of embedding changes for each set of temporally aligned chunks and record the minimum value. Afterwards, this value could be used to guide the formatting process, e.g. by discarding unwanted embedding sites and thus match this target minimum value for any chunk in this set. In other words, the embedding rate is harmonized across the different qualities, based on the critical HAS-path which consists of the concatenation of chunks with the fewest number of changes. This strategy is depicted in Figure 2 for reference.

Ignoring spatio-temporal resynchronization which is outside of the scope of this paper, the watermark demodulation procedure also needs to be updated. Pre-processing the Q bit-streams yielded Q alternate metadata files that could be used at detection and, in most cases, none of them can be used on its own after HAS transmission. The detection engine needs to infer the quality of each collected chunk in order to use the associated metadata. Let us assume that the detector can output a confidence score per chunk $\mathbf{c}$, e.g.

$$S_{\mathbf{c}} = \sum_{p \in \mathcal{P}(\mathbf{c})} \frac{\left|\sum_{e \in \mathcal{E}(p)} \mathrm{b}(e)\right|}{\sqrt{|\mathcal{E}(p)|}} \qquad (1)$$

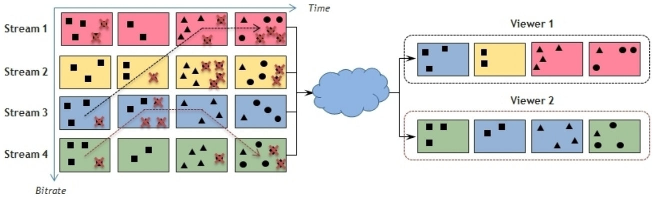

Fig. 2. Minimum-path payload modulation strategy. After pre-processing the bit-streams of the HAS master, the supervision module inspects all temporally aligned chunks and records the minimum number of embedding changes. Using this worst-case value, the formatting process then discards all superfluous embedding sites that could disrupt the payload modulation convention. As a result, any HAS-path respects the 5 changes per bit convention.

where $\mathcal{P}(\mathbf{c})$ is the set of payload bits embedded (maybe not fully) in the chunk $\mathbf{c}$, $\mathcal{E}(p)$ is the set of embedding positions associated to the payload bit p, and $\mathrm{b}(e) \in \{-1, 1\}$ is the demodulated bit at the embedding position e. A large score indicates that all embedding positions vote for the same bit value, which is unlikely to occur at random or with mismatching metadata. This confidence score can be used as an indicator by the detector to identify which quality is the most likely, after attempting detection in a chunk of the test video with all possible metadata. Extracting the watermarking payload then reduces to combining the information obtained from each chunk using the most responding metadata.

3. EXPERIMENTAL RESULTS

To validate the proposed solution, we conducted a number of experiments using a video watermarking system that operates in the H.264/AVC CABAC bit-stream [6, 11]. In a nutshell, the algorithm modifies large-valued *mvd* coefficients and thereby affects the average luminance value of some macro-blocks. This modulation mechanism is exploited to transmit the watermark, by spreading each payload bit over 80 changes. To account for the diversity of content in the real word, we considered a data set composed of four 30-minutes long videos and two short sequences taken from a mixture of feature and animation movies in HD quality (1920 × 1080, YUV 4:2:0, 24 frames per second). H.264 encoding has been performed using the publicly available `x264` encoder. To simulate HAS mastering, we considered seven qualities corresponding to encoding the master video with various compression ratios r (1:300, 1:200, 1:150, 1:100, 1:75, 1:50 and 1:25) and identical GOP structure, to guarantee that the subsequent segmentation in chunks can be temporally aligned.

3.1. Impact on the Watermarking Throughput

The proposed minimum-path payload modulation strategy eliminates by construction a number of watermark embedding sites. The watermarking throughput is therefore expected to drop to some extent. Table 2 reports the measured average loss of throughput, in percentage of the original embedding rate for each of the individual stream that the HAS master is composed of.

The first observation is that, statistically, no particular quality is significantly more hampered than the other ones. This is in line with the findings reported in [11] which clearly highlighted that the relationship between the watermarking throughput and the encoding parameters is not trivial. In other words, the critical chunks that bridle the embedding rate are not concentrated in one quality but are fairly well distributed across qualities. On the other hand, the loss of embedding rate appears to be tightly related to the chunk duration, shorter chunks being more constraining. For instance, the watermarking throughput is cut by 16% for 10-seconds chunks compared to 19% for 2-seconds chunks. This reduction implies that the algorithm will require more content to embed a full payload. The nominal embedding rate of this algorithm is about 1.25 payload bit per second for HD content, i.e. a 64-bits ID can be embedded in 51 seconds in average, assuming that there is no error correction overlaid on top. The throughput cut, induced by the minimum-path payload modulation strategy required to accommodate for HAS transmission, indicates that the same ID will be embedded in a full minute instead. This remains acceptable in most application use cases.

r	1:300	1:200	1:150	1:100	1:75	1:50	1:25
d=10s	14.12%	11.86%	16.88%	16.12%	17.35%	19.37%	16.56%
d=2s	17.16%	14.98%	19.84%	19.12%	20.31%	22.25%	19.53%

Table 2. Average percentage of watermarking throughput reduction when the minimum-path payload modulation strategy is applied. The loss is reported for each quality of the HAS content and for different chunk durations d.

The decoding strategy heavily relies on the assumption that a single quality will respond significantly more than the others. This hypothesis should hold if most of the physical embedding positions, aka. the macro-blocks modified by the alteration of the bit-stream, are different for the alternate qualities of the HAS content. As a sanity check, we verified how often different qualities rely on the same macro-blocks to watermark the content. Our measurements revealed that, in average, the overlap does not exceed 11%. In other words, watermark extraction for different qualities relies on different parts of the video content and the cross-talk should remain negligible.

3.2. Watermark Extraction Performances

To benchmark the proposed watermarking solution, we simulated various random HAS transmission paths and watermarked all selected chunks with the formatted embedding metadata produced by the pre-processing steps. We then concatenated the sequence of watermarked chunks altogether to obtain a copy of the original video. This copy then undergoes downsizing (D) and/or recompression (R) before being fed to the watermark detector. To evaluate the impact of the path estimation mechanism proposed in Subsection 2.3, we performed watermark demodulation with either the estimated HAS-path or the ground truth.

Figure 3 depicts the decoding performances at various stages of the watermark extraction process for the sequence *WallE*. As could be anticipated, the final bit error rate is heavily dependent on the accuracy of the estimated path. With short-length segments, there are fewer embedding sites per chunk and the decoding process is much more sensitive to errors. When the attacking strength exceeds a threshold, the accuracy of the estimated path collapses and the bit error rate rockets up, notably deviating from performances achieved using the ground truth. In contrast, longer chunks allow obtaining a more reliable estimation of the path and, as a result, decoding performances are nearly as good as with the ground truth.

This behavior has been reproduced for a number of other video sequences. Short HAS segmentation consistently yields poorer robustness due to the inaccuracy of the transmission path estimation process. This being said, even with 2-seconds chunks, the payload bit error rate remains below 5% for all the video sequences in our

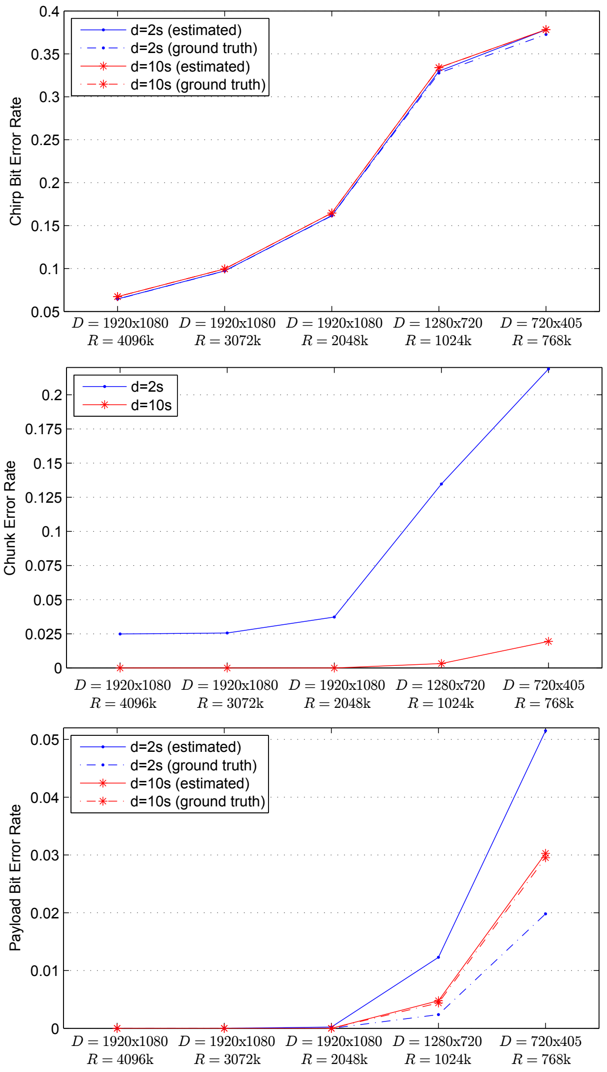

Fig. 3. Decoding errors at different stages of the watermark extraction process for the sequence *WallE*.

data set. Such level of error can be easily recovered using standard error correcting codes, although it implies that a full ID will take longer to embed/detect.

4. CONCLUSION AND PERSPECTIVES

This paper exemplified how HTTP adaptive transmission could disrupt watermarking systems that operate directly in the bit-stream. Essentially, the continuous switches between alternate qualities come into the way of the underlying payload modulation strategy. To cope with such emerging streaming mechanisms, we introduced a supervision module whose objective is to harmonize the watermarking throughput of the different qualities. This solution introduces a new challenge at the detector side since the extraction of the payload now requires an estimation of the path to use the correct metadata. While this estimation process may not be perfect, our experiments show that the error rate remains acceptable for a number of realistic scenarios. When possible, long-duration HAS chunks should be preferred since (i) the watermarking throughput is less impacted and (ii) the path estimation process is more stable.

Future work will investigate how to adjust this baseline solution to further improve performances. At the detector side, for instance, segments are processed independently and evenly for the moment. However, chunks with few embedding sites are more prone to introducing errors and should be weighted accordingly. Moreover, a number of payload bits are spread over two chunks. Such information could be exploited to disambiguate some cases during the path estimation in order to guarantee the coherence of the payload at the boundaries of the chunks. Conversely, the proposed harmonization strategy currently incurs a reduction of about 20% of the watermarking throughput. There may be alternate approaches that make a better use of these embedding sites that are discarded at the moment. For instance, it may be interesting to dynamically adjust the spreading length used at each quality to synchronize the pace of the payload modulation while making full use of the embedding capacity.

5. REFERENCES

[1] Jan Lee Ozer, *Producing Streaming Video for Multiple Screen Delivery*, Doceo Publishing, Apr. 2013.

[2] Christian Timmerer and Christopher Müller, “HTTP streaming of MPEG media,” in *Proc. of the Streaming Day*, Sept. 2010.

[3] Z. Zambelli, *IIS Smooth Streaming Technical Overview*, Microsoft Corporation, Mar. 2009.

[4] Dmitri Jarnikov and Jeroen M. Doumen, “Watermarking for adaptive streaming protocols,” in *Proc. of Secure Data Management*, Sept. 2011, vol. 6933 of *LNCS*, pp. 101–113.

[5] Dekun Zou and Jeffrey A. Bloom, “H.264/AVC substitution watermarking: A CAVLC example,” in *Media Forensics and Security I*, January 2009, vol. 7254 of *Proc. of SPIE*.

[6] Dekun Zou and Jeffrey A. Bloom, “H.264 stream replacement watermarking with CABAC encoding,” in *Proc. of IEEE ICME*, Jul. 2010, pp. 117–121.

[7] Niels J. Thorwirth, “Efficient watermarking approaches of compressed media,” U.S. patent 8,515,123, Aug. 2013.

[8] Thomas Stütz, Florent Autrusseau, and Andreas Uhl, “Inter-frame H.264/CAVLC structure-preserving substitution watermarking,” Tech. Rep. 2013-02, University of Salzburg, 2013.

[9] Séverine Baudry, Bertrand Chupeau, and Frédéric Lefebvre, “A framework for video forensics based on local and temporal fingerprints,” in *Proc. of IEEE ICIP*, Nov. 2009, pp. 2889–2892.

[10] Séverine Baudry, “Frame-accurate temporal registration for non-blind video watermarking,” in *Proc. of the ACM Workshop on Multimedia and Security*, Sept. 2012, pp. 19–26.

[11] A. Robert and G. Doërr, “Impact of content mastering on the throughput of a bit stream video watermarking system,” *Proc. of IEEE ICIP*, Sept. 2013.