

COMPRESSED VIDEO STREAM WATERMARKING FOR PEER-TO-PEER BASED CONTENT DISTRIBUTION NETWORK

Dekun Zou, Nicolas Prigent, and Jeffrey Bloom

Thomson Corporate Research

ABSTRACT

Peer-to-peer content distribution provides high network throughput with relatively low server cost and scales better than traditional content distribution networks with respect to the number of clients. It is ideal for applications requiring large data transfer such as Video-on-Demand or live video streaming. Traditional forensic watermarking systems are based on the assumption that each user receives a unique copy of video content to allow for tracking of pirated content back to the original recipient. However, P2P-based distribution systems are typically designed to supply identical copies to each user. This paper proposes a new watermarking framework that is suitable for P2P-based content delivery systems. Watermarking algorithms that are appropriate for this system are also discussed.

Index Terms— Peer-to-Peer, Video Watermarking, H.264/AVC, Content Distribution Network

1. INTRODUCTION

A Peer-to-Peer (P2P) network is a system in which networked hosts, called peers, share their resources (e.g., computing power and bandwidth) to achieve a common goal. Compared to traditional client-server systems, P2P provides a more scalable service at a lower cost.

P2P has a reputation as the most popular method to share pirated content. Ferguson shows that 2006 P2P traffic accounted for more than 60% of the all Internet traffic [1]. It is widely believed that a large percentage of this traffic is pirated content. However, it is important to distinguish between the P2P technology which can provide efficient distribution and applications of P2P for sharing pirated content. As a technology, P2P content distribution is envisioned as the next multimedia network distribution channel for legitimate content [2], especially for large content such as HD video. Many legitimate P2P content distribution systems such as Joost [3], BBC iPlayer [4], and Open Media network [5] are available today.

One of the main concerns of content owners in using P2P to distribute their content (or any distribution method for that matter), is the susceptibility of the system to piracy. Digital rights management (DRM) systems, such as CSS for DVD and AACS for Blu-ray, are widely used for content

distribution. In fact, Windows Media DRM has already been used to protect some content legally distributed over the BitTorrent P2P system [10].

Two issues here should not be confused. One issue that has received much attention is the use of P2P to distribute pirated content, independent of the source of that piracy. The second issue is the ability of a legitimate distribution system to prevent (or at least resist) leaks of the content. An acceptable P2P content distribution network (CDN) will not be the source of piracy. This paper addresses the second issue, not the first.

DRM can protect content until it is decrypted and presented for consumption. At that point the content is typically unprotected. In addition, most DRM implementations are vulnerable to attacks that render the content unprotected. Digital watermarking has recently been used as a complementary protection mechanism enabling pirated content to be tracked back to the authorized user responsible for the unauthorized distribution [6].

Most watermarking techniques work directly in the signal domain or the transformed signal domain. However, since content distributed over P2P are likely to be in an entropy-encoded compressed domain, we show in the next section that these techniques are not suitable for a P2P CDN system without a customized player. Some watermarking algorithms work in “compressed streams”. These usually first perform a partial decompression, more specifically an entropy decode, to expose the syntax elements (coefficients, motion vectors, modes, etc.); perform the watermarking in the syntax domain; and finally apply an entropy encode to obtain the marked compressed stream (e.g., [7] for MPEG, [8] for MPEG2, and [9] for H.264/AVC). Hence, we will show that they do not fit in our P2P CDN watermarking framework. In this article, we will discuss H.264/AVC based watermarking schemes that can be used in the proposed framework.

Section 2 discusses the system framework that can achieve digital watermarking in a P2P-based CDN. Section 3 discusses a watermark for H.264/AVC encoded video streams that fits the P2P CDN framework. Finally, in Section 4, we discuss the security of our proposal.

2. VIDEO WATERMARKING FRAMEWORK FOR P2P VIDEO DELIEVERY SYSTEM

In a P2P-based video-on-demand (VoD) or streaming system, the content is divided into small units called chunks. Unlike traditional client-server model, a peer can get each chunk from any peer in the network and there is no control on where a particular client gets content. The key feature is that all the peers will download the exact same version of a piece of content. However, in traditional forensic watermarking systems, the content distributed to a client should be unique to that client since these differences carry the forensic watermark information.

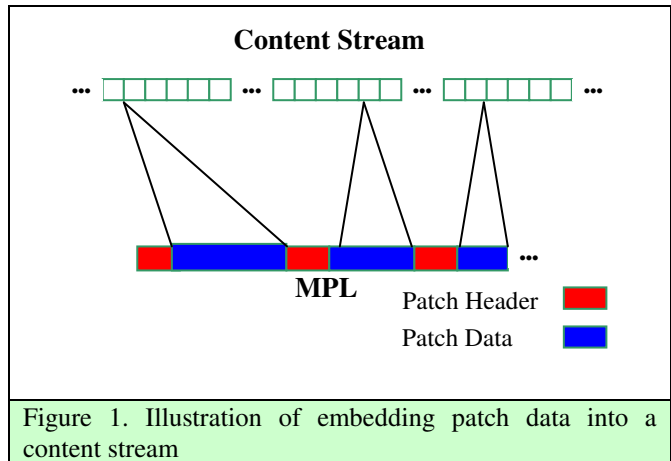
2.1. System Description

The system proposed in this paper is a hybrid P2P in which most of the content is distributed through a P2P network and a small portion is distributed through a client-server network. The content downloaded via the P2P network is not complete. To be more specific, a very small portion of the content is missing. Therefore, the content downloaded via a P2P network cannot play or will offer a bad experience if played by itself. For example, the content may freeze occasionally or stop playing completely. In order to obtain the complete content, a small data file must be downloaded from a server. We define the Media Patch List (MPL) as the portion required to complete the stream. This MPL fixes the stream and makes it completely playable.

In addition to repairing the content, the MPL serves the addition role of enabling watermark embedding. In one scenario, different clients receive different versions of the MPL for the same content.

The MPL is distributed to the users via a different distribution channel, typically through a client-server point-to-point mode. Thus, a crucial requirement is that the MPL should be small enough for a server to handle a large number of users. The basic unit in an MPL is called a patch. Figure 1 illustrates the structure of the MPL and the usage of MPL entries. A patch consists of a fixed length header and a variable length data field. The patch header contains the location in the stream where this patch should be inserted, and the length of the data field. The data field is the actual data to be inserted into the stream. The embedder reads the MPL entries and, for each entry, inserts the data field of this patch into the stream at the location specified by the header of this entry.

The distribution of the MPL can differ according to the application. For VoD applications, the MPL can be transported to the client during the client setup/authentication process. However, for live content streaming, the server cannot know the MPL before live streaming starts. Therefore, it should be streamed to clients as long as the program lasts. For each client, a separate connection may be needed to transmit the MPL entries.



Since the video content is compressed, the patch has to be inserted as shown in Figure 1 such that the stream is still compliant to the compression standard (including the entropy encoding). However, since the patch is built to be different for different users, there should be at least two version of each patch for each location. The challenge is to design a watermark algorithm that provides at least two different versions of patch data such that both versions should satisfy the following:

1. the modified compressed stream must be compliant with the specific compression standard,
2. the motion image sequence obtained by decompressing the modified compressed stream must be perceptually indistinguishable from the motion image sequence obtained by decompressing the original compressed stream, and
3. the modification to the stream must result in a measurable change in the motion image sequence obtained by decompressing the modified compressed stream.

Watermarking algorithms that operate in the syntax domain, such as those described in [7], [8], and [9], are not suitable for this framework because advanced compression standard like H.264/AVC use context adaptive entropy coding so that the change to a syntax value in the compressed syntax domain will affect the entropy coding process of the elements that follow. Thus, the entropy encoded bitstream would be different. This would require the P2P-based CDN to be overwhelmingly composed of patches (thus reverting the system back to a client-server system). In Section 3, we will discuss an H.264/AVC based watermarking algorithm that satisfies the above requirements.

3. WATERMARK EMBEDDING

H.264/AVC is one of the most efficient video compression methods and its use is becoming widespread. Two entropy coding methods are supported in H.264/AVC: a variable length code, CAVLC and an arithmetic code, CABAC. We

will describe a CAVLC-based method and expect CABAC-based methods to be published in the near future.

3.1. CAVLC-based Watermarking Approach

In our previous work [12], we proposed a CAVLC-based watermarking approach. In that algorithm, the intra-prediction mode of a macroblock is changed to carry watermark information. To be specific, the field that carries the intra-prediction mode information, `mb_type`, is changed. The reason we choose `mb_type` is that, unlike most of the syntax elements in CAVLC that use adaptive VLC, `mb_type` is entropy coded with non-adaptive VLC. Therefore, a change in the `mb_type` field will not affect the decoding of subsequent syntax elements in the stream. Please refer to [12] for the detailed algorithm. Here, we present some proof-of-concept experimental results.

The system that generates the patches is based on the JM H.264/AVC decoder [12]. This system obtains a complete list of alternative intra-prediction modes for all 16x16 intra-predicted blocks. A simplified cost analysis model first estimates the fidelity for a block by measuring the total absolute luminance change of the block when the alternative intra-prediction mode is used. The measure of estimated robustness used is the luminance change of the whole block under the alternative intra-prediction mode. Thresholds for both the fidelity measure and the robustness measure are used to identify changes that will produce changes of sufficient fidelity and robustness. These changes are then used as patches.

In the current implementation, portions of the stream corresponding to the selected intra-prediction modes are removed from the stream. Then, for each patch, two versions are created: one with the original intra-prediction mode data and one with the alternative intra-prediction mode. For a particular client, the server provides a unique combination of original and modified patches. The client has no way to determine whether an individual patch contains original or modified data.

A simple test was performed with a 6 minute clip from the movie "Independence Day." Each frame is 1920x1080 at 24 frames per second. The test clip consists of 8640 frames compressed at 15Mbps with H.264/AVC using CAVLC entropy coding.

In our testing, the basic unit of payload is a HEX symbol. We apply a simple spread spectrum channel coding method. Sixteen 300-bit long binary reference sequences are predefined. Each sequence is assigned to one of the 16 HEX symbols. To embed a symbol we use its associated 300-bit long reference sequence where each bit is embedded into a 16x16 intra-predicted block. The content used in this test supported the embedding of 18 HEX symbols.

Detection is accomplished with a correlation-based method. The extracted bit sequence is divided into segments of 300 bits. Each segment is correlated against each of the predefined 300-bit reference patterns. The symbol

corresponding to the pattern with highest correlation is reported as the recovered symbol.

Robustness to downsizing and compression were tested. `ffmpeg` [13] is used for downsizing attacks and `DIVX` for compression. Table 1 shows the results of these tests. The first column is the attack. The second column is the number of symbol errors out of 18 embedded symbols. The third column lists the average correlation value across the 18 recovered symbols.

In these tests, the proposed watermarking algorithm withstood downsizing to CIF without any symbol errors although the average correlation did decrease significantly. Recompression of this downsized content further degraded the correlation and began to introduce symbol errors. This proof-of-concept evaluation demonstrates that the proposed system is feasible and may be sufficiently robust for many applications.

Attack	Number of Symbol Errors	Average Correlation
No Attack	0	0.9685
Downsize to 960x540	0	0.8133
Downsize to 480x270	0	0.2778
Downsize to CIF (352x288)	0	0.1804
Downsize to CIF & compressed to 1Mbps divx	1	0.1148
Downsize to CIF & compressed to 780k divx	2	0.1181
Downsize to CIF & compressed to 300k divx	3	0.1026

Table 1. Robustness test results against downsizing and subsequent recompression

3.2. A better solution – CABAC-based

In H.264, another important entropy coding method is CABAC. CABAC is an arithmetic coding scheme that achieves improved compression performance by maintaining individual contexts for each syntax element type and by adapting these contexts with each coded element. Coding of one syntax element causes the associated context to adapt. H.264/AVC defines 460 separate contexts that are maintained during encoding and decoding. In general it is impossible to recover the decoding process from a damaged entropy coded bitstream, even if only 1 bit is missing. This property of the arithmetic code makes it perfectly suitable in our proposed watermarking framework. Referring to Figure 1, the client will be unable to decode the received stream without the MPL. When the arithmetic decoding process reaches a location where data is missing, it will not detect any error. Instead, it will continue to decode using the wrong context and the decoded content will be garbage. In preliminary testing with such data, we see the decoder crash, not at the point where the data is removed, but at a much later position. This property makes a stream analysis attack

very difficult. H.264/AVC watermarking algorithms based on CABAC will be published in the near future.

4. SECURE SYSTEM DISCUSSION

The solution we present in this article is a reactive security measure that enables the insertion of a watermark to diversify content distributed over a P2P CDN. It enables the tracing of pirated content and serves as a deterrent to future piracy. Since the solution will output a watermarked H.264/AVC stream to a standard player, our proposal ensures H.264 compliance. Even if an attacker is able to tap into the stream as it is being passed to the player, the captured video will be watermarked and will be traceable back to the user.

Like in many forensic watermarking applications, our solution can be envisioned as a second line of defense. To increase the security of the system, the embedding process should be performed in a secure environment. This secure environment will be responsible for communicating with the MPL server, decrypting the received MPL, and inserting the patches into the stream. A complementary protection would be to encrypt the transport stream prior to P2P distribution. As such, we could use a proactive DRM mechanism to protect the media data chunks.

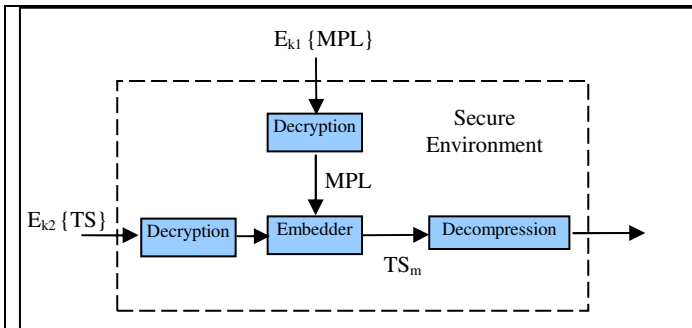


Figure 2. Secure player architecture.

Additional security can be obtained by limiting access to the clear-text (but still watermarked) H.264/AVC output of the P2P layer. We can rely on a link encryption to send the player an encrypted stream, but the overhead in establishing a secure channel between the P2P layer and the player may be too high. Another solution is to incorporate part of the player, specifically the decoder, into the secure environment. The output of the secure environment is now the decoded and watermarked pixel data. A typical secure system is shown in Figure 2. While such a secure system architecture could be considered challenging in a PC environment, it would be easier to achieve in a proprietary environment such as home routers for which manufacturers propose P2P clients today [11].

Finally, an attacker that gains access to the decompressed digital content or the media stream before decompression may collude with other users to remove the watermark. The

proposed algorithm in the basic form cannot prevent the collusion attack. However, using an anti-collusion coding layer such as [15][16] prior to watermark embedding would help resist such collusion attacks.

5. CONCLUSIONS

In the P2P-based CDN, all the clients receive the same copy of the content. This paper proposed a watermarking framework for a P2P-based content distribution network. In addition to any DRM-based content protection, this watermark acts as a second line of defense that enables forensic tracking. Our framework uses a Media Patch List to fix the stream while embedding the watermark information. The properties of watermarking algorithms that fit in the proposed system were presented along with an example algorithm and proof-of-concept experimental results.

6. REFERENCES

- [1] D. Ferguson, "Trends and statistics in peer-to-peer," presented at Workshop on Technical and Legal Aspects of Peer-to-Peer Television, 2006.
- [2] P. Rodriguez, S.M. Tan, C. Gkantsidis., "On the feasibility of commercial, legal P2P content distribution," *ACM SIGCOMM Computer Communication Review*, Volume 36, Issue 1 (January 2006).
- [3] www.joost.com
- [4] <http://www.bbc.co.uk/iplayer/>
- [5] <http://www.omn.org/>
- [6] I. Cox, M. Miller, and J. Bloom, "Digital Watermarking: Principles & Practice", San Mateo, CA: Morgan Kaufman, 2001
- [7] D. Simitopoulos, S.A. Tsaftaris, N.V. Boulgouris, M.G. Strintzis, "Fast MPEG watermarking for copyright protection", 9th International Conference on Electronics, Circuits and Systems, 15-18 Sept. 2002 Page(s):1027 - 1030 vol.3.
- [8] T. Chung, M. Hong, Y. Oh, D. Shin, S. Park, "Digital watermarking for copyright protection of MPEG2 compressed video", *IEEE Transactions on Consumer Electronics*, Volume 44, Issue 3, Aug. 1998 Page(s):895 - 901
- [9] M. Noorkami, R.M. Mersereau, "Compressed-domain video watermarking for H.264", *IEEE ICIP 2005*. 11-14 Sept. 2005, Vol. 2, page(s): II- 890-3.
- [10] Bittorrent Inc. FAQ, What is Windows DRM? Why do I need to authorize my computer to view a file? Do I have to do this every time? <http://www.bittorrent.com/btusers/nowplaying/faq/ten-usage#3n104>
- [11] BitTorrent Device Partners, <http://www.bittorrent.com/devices/?csrc=splash>
- [12] Dekun Zou; Bloom, J.A.; "H.264/AVC stream replacement technique for video watermarking" *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008. March 31 2008-April 4 2008 Page(s):1749 - 1752.
- [13] JM H.264/AVC Software: <http://iphone.hhi.de/suehring/tml/>
- [14] FFMPEG: <http://ffmpeg.mplayerhq.hu/>
- [15] G. Tardos; "Optimal probabilistic fingerprint codes." *Proc. of the 35th annual ACM symposium on theory of computing*, San Diego, CA, USA; (2003)Pages: 116 - 125
- [16] S. He and M. Wu, "Collusion-Resistant Video Fingerprinting for Large User Group", *IEEE Trans. on Information Forensics and Security*, Vol. 2, No. 4, pp. 697-709, Dec 2007.