

논문 2026-3-1 <http://dx.doi.org/10.29056/jsf.2026.09.01>

A/B 변형 세그먼트의 중복 저장 절감을 위한 CDN Edge 기반 포렌식 워터마킹 아키텍처

이재욱*, 이지은*, 박병찬*, 신용태**†

CDN Edge-Based Forensic Watermarking Architecture for Reducing Redundant Storage of A/B Variant Segments

Jewook Lee*, Jieun Lee*, Byeongchan Park*, Yongtae Shin**†

요 약

OTT 서비스에서는 불법 유통 콘텐츠의 유출 경로를 식별하기 위해 포렌식 워터마킹 기술이 활용되고 있다. 기존 A/B 워터마킹 방식은 CDN 캐시 재사용에 유리하지만, 각 콘텐츠의 해상도별 세그먼트마다 A와 B 두 개의 변형 세그먼트를 저장해야 하므로 저장 공간 및 운영 관리 부담이 증가하는 한계가 있다. 본 논문에서는 단일 공통 세그먼트와 WEM(Watermark Embedding Metadata)을 저장하고, 사용자 요청 시 CDN Edge에서 WMID(Watermark Identifier)를 삽입하는 요청별 포렌식 워터마킹 아키텍처를 제안한다. 영상 10개를 4개 해상도로 인코딩하여 실험한 결과, 저장량을 7.48 GB에서 3.74 GB로 줄여 약 49.98%의 저장 공간을 절감하였다. 또한 warm-cache 상태에서 동시 사용자 10명, 50명, 100명 조건 모두 기존 방식과 동일하게 100%의 캐시 적중률을 유지하였다. 이를 통해 제안 방식이 기존 A/B 워터마킹 방식의 캐시 재사용성을 유지하면서 저장 공간 부담을 완화할 수 있음을 확인하였다.

Abstract

Forensic watermarking is used in OTT services to identify the source of illegally distributed content. Conventional A/B watermarking supports CDN cache reuse but requires two variant segments for each content resolution and segment, increasing storage and management overhead. This paper proposes a request-based forensic watermarking architecture that stores a single common segment and Watermark Embedding Metadata (WEM) and embeds a Watermark Identifier (WMID) at the CDN edge upon user request. Experiments with 10 videos encoded at four resolutions showed that storage was reduced from 7.48 GB to 3.74 GB, corresponding to a 49.98% reduction. Under warm-cache conditions, the proposed method also maintained a 100% cache hit ratio for 10, 50, and 100 concurrent users. These results indicate that the proposed architecture reduces storage overhead while preserving the cache reusability of conventional A/B watermarking.

한글키워드 : OTT, A/B 포렌식 워터마킹, CDN Edge, 공통 세그먼트, WEM

keywords : OTT, A/B Forensic Watermarking, CDN Edge, Common Segment, WEM

* 숭실대학교 컴퓨터학과

** 숭실대학교 컴퓨터학부

† 교신저자: 신용태(email: shin@ssu.ac.kr)

접수일자: 2026.09.05. 심사완료: 2026.09.11.

게재확정: 2026.09.20.

1. 서론

인터넷 동영상 서비스(Over-The-Top, 이하 OTT 서비스)는 DASH, HLS와 같은 HTTP 기반 적응형 스트리밍 기술을 사용하여 다양한 네트워크 환경에서 동영상 콘텐츠를 제공한다. CDN은 대규모 사용자의 동시 접속을 처리하기 위한 핵심 인프라로, 콘텐츠 세그먼트를 CDN Edge에 캐싱하여 전송 지연과 오리진 서버의 부하를 줄일 수 있다[1-2]. 이에 사용자는 가까운 CDN Edge 서버로부터 빠르게 콘텐츠를 수신할 수 있다. 최근 OTT 콘텐츠의 불법 복제와 무단 재배포가 증가하면서 유출 경로를 추적하기 위한 포렌식 워터마킹의 필요성이 커지고 있다. 포렌식 워터마킹은 콘텐츠에 사용자 또는 세션 식별 정보를 삽입하여 불법 유통 발생 시 유출 경로를 추적하는 기술이다[3]. 그러나 사용자별로 워터마크가 적용된 세그먼트를 동적으로 생성하는 방식은 CDN 캐시 재사용이 어렵고, 대규모 접속 환경에서는 오리진 서버 부하와 네트워크 비용이 증가할 수 있다[4]. 이러한 문제를 해결하기 위해 각 콘텐츠 세그먼트에 대해 A와 B 변형 세그먼트를 사전에 생성하고, 사용자의 워터마크 비트열에 따라 두 변형 세그먼트 중 하나를 선택하여 전송하는 A/B 워터마킹 방식이 제안되었다[5]. 이 방식은 CDN 캐싱을 활용하면서 사용자별 워터마크 시퀀스를 구성할 수 있으나 각 콘텐츠의 해상도 및 세그먼트마다 A와 B 두 개의 변형 세그먼트를 저장해야 하므로 콘텐츠 수, 해상도 및 세그먼트 수가 증가할수록 저장 공간 및 운영 관리 부담이 증가하는 한계가 있다[5-7].

이에 본 논문에서는 기존 A/B 워터마킹 방식의 CDN 캐싱 효율을 유지하면서 중복 저장을 줄이기 위해 단일 공통 세그먼트와 WEM(Watermark Embedding Metadata)을 저장하고,

사용자 요청이 발생하면 CDN Edge에서 요청 토큰으로부터 생성한 WMID(Watermark Identifier)를 공통 세그먼트에 삽입하는 CDN Edge 기반 요청별 포렌식 워터마킹 아키텍처를 제안한다.

2. 관련 연구

2.1 OTT 전송과 CDN Edge 캐싱

CDN Edge 캐싱반복 요청되는 콘텐츠 세그먼트를 사용자와 가까운 위치에서 제공하여 오리진과 백홀 네트워크의 부하를 줄이는 방식이다. 선행 연구[1]에 따르면 iflix 요청 데이터셋 실험에서 ColabCache의 캐시 적중률이 DRL-CCCT(Deep Reinforcement Learning algorithm for Collaborative Video Caching and Transcoding)보다 23% 높고, 백홀 트래픽과 접근 지연이 각각 14%, 17% 감소했다고 보고하였다. 또한 선행 연구[2]에서는 합성 워크로드 환경에서 Edge 캐싱, MEC(Multi-access Edge Computing) 오프로딩 및 Q-learning 기반 적응형 라우팅을 결합한 구조가 약 88%의 캐시 적중률을 기록했다고 보고하였다. 이는 반복 요청되는 콘텐츠를 CDN Edge에서 캐싱하는 구조가 캐시 재사용성을 높이고, 오리진 및 백홀 네트워크의 전송 부하를 줄이는 데 효과적임을 보여준다. 이에 본 논문에서는 사용자 요청 간 공통으로 재사용되는 세그먼트를 CDN Edge에 캐싱하여 캐시 재사용성을 유지한다[8].

2.2 A/B 포렌식 워터마킹

A/B 포렌식 워터마킹 방식은 각 세그먼트에 대해 비트 0과 1에 대응하는 A와 B 변형 세그먼트를 사전에 생성하고, 사용자 또는 세션별 워터마크 비트열에 따라 CDN Edge에서 하나의 변형 세그먼트를 선택하여 전송하는 방식이다[5]. 선행

연구 [6, 7]에서는 라이브 스트리밍 환경에서 사용자별 토큰에 따라 CDN Edge에서 A/B 변형 세그먼트를 선택하고 두 변형 세그먼트를 캐싱하는 구조를 제시하였으며, A/B 선택열을 이용하여 유출 사용자와 제공 시점을 추적하는 사례를 제시한다. 또한 shield cache와 CDN 설정을 최적화하여 5xx 응답 오류를 줄이고, KPI(Key Performance Indicator)를 6% 향상시켰다[3]. 이는 A/B 워터마킹 환경에서 캐시 계층과 CDN 설정의 최적화가 워터마킹 서비스 안정성을 향상시키는 데 효과적임을 보여준다. 그러나 기존 A/B 워터마킹 방식은 각 콘텐츠의 해상도별 세그먼트마다 A와 B 두 개의 변형 세그먼트를 저장하는 구조이므로 콘텐츠 수, 해상도 및 세그먼트 수가 증가할수록 저장 공간 및 운영 관리 부담이 증가하는 한계가 있다.

3. 제안 아키텍처

본 논문에서는 기존 A/B 워터마킹 방식의 중복 저장 구조를 보완하기 위해 토큰 기반 식별과 CDN Edge 캐싱을 활용한 요청별 포렌식 워터마킹 아키텍처를 제안한다. 제안 방식은 사용자 정보와 무관한 단일 공통 세그먼트를 CDN Edge에 캐싱하고, 사용자 요청에 따라 WMID를 공통 세그먼트에 삽입하도록 구성한다. 이를 위해 공통 세그먼트와 함께 WEM을 저장하며, WEM은 각 WMID 비트에 대응하는 사전에 세그먼트에 구성한 16비트 워터마크 삽입 예약 영역 정보를 저장하는 메타데이터로 정의한다. 전체 아키텍처 구조는 그림 1과 같다.

제안 아키텍처는 CDN Edge에서 공통 세그먼트와 WEM을 조회한 후 사용자별 WMID를 적용하고, 워터마크가 삽입된 세그먼트를 생성하여 사용자에게 전달한다.

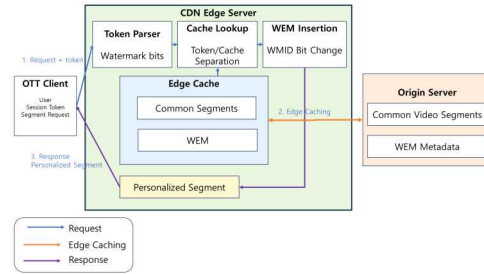


그림 1. 제안 아키텍처 구조
Fig. 1. Proposed Architecture Structure

3.1 요청별 워터마크 삽입 절차

제안 방식에서 사용자가 워터마크 토큰 (Watermark Token)을 포함한 세그먼트 요청을 전송하면 CDN Edge는 캐시에서 공통 세그먼트를 조회하고, 요청 토큰의 하위 16비트를 이용하여 WMID를 생성한다. 이때, 서로 다른 요청 토큰으로부터 동일한 WMID가 생성될 수 있다. 따라서 실제 서비스 환경에서는 요청 토큰, WMID 및 사용자 식별정보 간의 대응관계를 별도로 관리하며, 추출된 WMID와 관리 정보를 대조함으로써 해당 사용자 또는 세션을 식별한다. WEM의 바이트 위치는 사전 전처리 과정에서 각 세그먼트에 워터마크 삽입 예약 영역을 사전에 구성하며, 이 중 16개의 바이트 위치를 WMID의 각 비트와 대응시킨다. WMID 비트가 1인 경우 해당 위치의 바이트를 0x01과 XOR하여 변형하고, 비트가 0인 경우에는 해당 바이트를 변경하지 않는다. 사용자(u)의 요청 토큰을 T_u 일 때, 16비트 WMID(W_u)는 다음 식 (1)과 같다.

$$W_u = T_u \bmod 2^{16} \quad (1)$$

WMID의 i 번째 비트 $b_i(u)$ 는 최상위 비트부터 순차적으로 추출하며, 다음 식 (2)와 같이 정의한다.

$$b_i(u) = \left\lfloor \frac{W_u}{2^{15-i}} \right\rfloor \bmod 2, i = 0, \dots, 15 \quad (2)$$

k 번째 세그먼트에서 WEM의 i 번째 오프셋을 $p_{k,i}$, 공통 세그먼트를 B_k , 사용자(u)의 응답 세그먼트를 $S_{u,k}$ 라 할 때, 삽입 연산은 다음 식 (3)과 같다.

$$\begin{aligned} S_{u,k}[p_{k,i}] &= B_k[p_{k,i}] \oplus b_i(u), i = 0, \dots, 15 \\ S_{u,k}[j] &= B_k[j], \quad j \notin \{p_{k,0}, \dots, p_{k,15}\} \end{aligned} \quad (3)$$

여기서 $\oplus$ 은 WMID 비트가 1인 경우 해당 바이트와 0x01을 연산하여 최하위 비트를 반전시키는 것을 의미한다. 삽입된 WMID는 공통 세그먼트를 참조하여 유출 세그먼트와 WEM 위치의 바이트 값을 비교함으로써 추출한다. 이는 불법 복제 및 무단 재배포 과정에서 세그먼트 일부 구조 또는 데이터가 변경되더라도 WEM 위치의 바이트 값이 유지되는 경우, WMID를 추출할 수 있다. 유출 세그먼트 L_k 와 공통 세그먼트 B_k 의 WEM 위치를 비교하여 k 번째 세그먼트에 삽입된 WMID의 i 번째 비트 $b_i^{(k)}$ 를 추출하며, 이는 다음 식 (4)와 같다.

$$b_i^{(k)} = (L_k[p_{k,i}] \oplus B_k[p_{k,i}]) \bmod 2 \quad (4)$$

4. 실험 및 결과

4.1 실험 환경

본 논문에서는 제안 방식의 저장량, 캐시 적중률 및 CDN Edge CPU 사용률을 기존 A/B 워터마킹 방식과 비교 분석하였다. 실험에는 입력 영상 파일 10개를 각각 360p와 540p, 720p, 1080p의 4개 해상도로 인코딩하였으며, 동시 사용자 수는

10명, 50명, 100명으로 설정하였다. Origin Server는 Nginx를 이용하여 구성하였으며, CDN Edge 기능은 OpenResty를 이용하여 구현하였다. 세부 실험 환경은 표 1과 같다.

표 1. 실험 환경
Table 1. Experimental Environment

항목	환경
OS	Windows 11
CPU	AMD Ryzen 9 7940HX
RAM	64 GB
Docker	Docker 29.6.2
Python	Python 3.11
Origin Server	Nginx 1.25-alpine
CDN Edge	OpenResty 1.29.2.3
Encoder	FFmpeg

4.2 실험 결과

4.2.1 저장량 비교

본 논문에서는 기존 A/B 워터마킹 방식과 제안 방식의 저장량을 비교하였다. 기존 A/B 워터마킹 방식의 변형 세그먼트 저장량 (D_{AB})은 전체 콘텐츠와 해상도에 대해 생성된 A 변형 세그먼트 D_A 와 B 변형 세그먼트 D_B 의 합으로 다음 식 (5)와 같이 정의한다.

$$D_{AB} = D_A + D_B \quad (5)$$

제안 방식의 저장량(D_T)은 공통 세그먼트 저장량 D_S 와 WEM 저장량 D_W 의 합으로 구성되며, 다음 식 (6)과 같이 정의한다.

$$D_T = D_S + D_W \quad (6)$$

기존 A/B 워터마킹 방식 대비 제안 방식의 저장

공간 절감률(R_S)은 다음 식 (7)과 같이 계산한다.

$$R_S = \frac{D_{AB} - D_T}{D_{AB}} \times 100 \quad (7)$$

다음 표 2는 기존 A/B 워터마킹 방식과 제안 방식의 저장량을 비교한 결과이며, 소수점 둘째 자리까지 반올림하여 나타내었다.

표 2. 기존 A/B 워터마킹 방식과 제안 방식의 저장량 비교

Table 2. Comparison of storage amounts between the existing A/B Watermarking method and the proposed method

구분	저장량
기존 A/B 워터마킹 방식 저장량 (GB)	7.48
공통 세그먼트 저장량 (GB)	3.74
WEM 저장량 (MB)	1.09
제안 방식 총 저장량 (GB)	3.74
저장 공간 절감률 (%)	49.98

기존 A/B 워터마킹 방식의 A와 B 변형 세그먼트의 총 저장량은 약 7.48 GB로 측정되었다. 제안 방식은 약 3.74 GB의 공통 세그먼트와 약 1.09 MB의 WEM을 저장하며, 이를 포함한 총 저장량은 약 3.74 GB로 측정되었다. 이는 제안 방식이 기존 A/B 워터마킹 방식 대비 약 49.98%의 저장 공간을 절감하였음을 나타낸다.

4.2.2 캐시 적중률 및 CDN Edge CPU 사용률 비교

본 논문에서는 기존 A/B 워터마킹 방식과 제안 방식의 캐시 적중률 및 CDN Edge CPU 사용률을 비교하였다. 캐시 적중률 비교는 A/B 변형 세그먼트와 공통 세그먼트 및 WEM을 CDN

Edge 캐시에 사전 적재한 warm-cache 상태에서 수행하였다. 캐시 적중률(C)은 전체 요청 수 R 과 캐시 적중 요청 수 H 를 이용하여 다음 식 (8)과 같이 정의한다.

$$C = \frac{H}{R} \times 100 \quad (8)$$

제안 방식의 CDN Edge CPU 사용률(C_u)은 다음 식 (9)와 같이 정의한다.

$$C_u = \frac{1}{N} \sum_{i=1}^N C_{u,i} \quad (9)$$

다음 표 3은 동시 사용자 수에 따른 기존 A/B 워터마킹 방식과 제안 방식의 캐시 적중률 및 CDN Edge CPU 사용률을 비교한 결과이며, 소수점 둘째 자리까지 반올림하여 나타내었다.

표 3. 동시 사용자 수에 따른 캐시 적중률 및 CDN Edge CPU 사용률 비교

Table 3. Comparison of cache hit rate and CDN Edge CPU utilization by concurrent user count

구분	10명	50명	100명
기존 A/B 워터마킹 방식 캐시 적중률 (%)	100	100	100
제안 방식 캐시 적중률 (%)	100	100	100
기존 A/B 워터마킹 방식 CDN Edge CPU 사용률 (%)	1.45	3.58	6.25
제안 방식 CDN Edge CPU 사용률 (%)	3.98	16.90	28.52

기존 A/B 워터마킹 방식과 제안 방식은 warm-cache 상태의 모든 동시 사용자 조건에서 100%의 캐시 적중률을 기록하였다. 이를 통해 제안 방식이 A/B 변형 세그먼트 대신 단일 공통 세그먼트를 사용하는 구조에서도 기존 방식과 동일한 캐시 적중률을 유지함을 확인하였다. 반면 제안 방식의 CDN Edge CPU 사용률은 동시 사용자 10명, 50명, 100명 조건에서 기존 A/B 워터마킹 방식보다 각각 2.53pp(percentage points), 13.32pp, 22.27pp 높게 나타났다. 이는 사용자 요청별 WMID 생성과 WEM 기반 XOR 연산을 CDN Edge에서 수행함에 따라 추가적인 연산 부하가 발생한 결과로 판단된다.

5. 결론

OTT 서비스의 이용이 증가함에 따라 콘텐츠의 불법 유통에 대응하고 유출 경로를 추적하기 위한 포렌식 워터마킹의 필요성이 증가하고 있다. 기존 A/B 워터마킹 방식은 CDN Edge에서 A/B 변형 세그먼트를 선택하여 사용자별 워터마크를 구성하면서도 캐시 재사용성을 유지할 수 있다. 그러나 각 콘텐츠의 해상도별 세그먼트마다 A와 B 두 개의 변형 세그먼트를 사전에 생성하여 저장해야 하므로 콘텐츠 수, 해상도 및 세그먼트 수가 증가할수록 저장 공간 및 운영 관리 부담이 증가하는 한계가 있다. 본 논문에서는 이러한 중복 저장 구조를 보완하기 위해 단일 공통 세그먼트와 WEM을 저장하고, 사용자 요청 시 CDN Edge에서 요청 토큰으로부터 생성한 WMID를 삽입하는 CDN Edge 기반 요청별 포렌식 워터마킹 아키텍처를 제안하였다. 입력 영상 10개를 360p, 540p, 720p, 1080p의 4개 해상도로 인코딩하여 실험한 결과, 기존 A/B 워터마킹 방식의 총 저장량은 약 7.48 GB, 제안 방식은 약 3.74 GB로 측정되어 저장 공간을 약 49.98% 절

감하였다. 또한 warm-cache 상태에서 동시 사용자 10명, 50명, 100명 조건 모두 기존 방식과 동일하게 100%의 캐시 적중률을 유지하였다. 반면, 제안 방식의 CDN Edge CPU 사용률은 기존 방식보다 각각 2.53pp, 13.32pp, 22.27pp 높게 나타났으며, 이는 요청별 WMID 생성 및 WEM 기반 XOR 연산에 따른 추가적인 연산 부하로 발생한 결과로 판단된다. 향후 연구에서는 CDN Edge의 워터마크 삽입 연산을 최적화하고, 실제 CDN 환경에서의 대규모 부하 검증과 함께 WEM 위치 선정 및 삽입된 WMID의 강인성에 대한 추가 검증을 수행하고자 한다.

본 연구는 문화체육관광부 및
문화기술기획평가원의 소프트웨어 저작권
연구개발(R&D) 사업으로 수행되었음(과제명
: OTT 콘텐츠 저작권 보호기술개발 및 적용을
위한 저작권기술(+법) 융합인재양성,
과제번호 : RS-2023-00225267, 기여율: 100%)

참 고 문 헌

- [1] Anirban Lekharu, Pranav Gupta, Arijit Sur, Moumita Patra, "Collaborative Video Caching in the Edge Network using Deep Reinforcement Learning", ACM Transactions on Internet Things, 5(3), Article 16, pp.1-26, 2024.8, DOI : 10.1145/3664613
- [2] A. D. Salman, A. T. Zeyad, A. A. S. Al-karkhi, S. M. Raafat, A. J. Humaidi, "Hybrid CDN Architecture Integrating Edge Caching, MEC Offloading, and Q-Learning-Based Adaptive Routing", Computers, 14(10), Article 433, 2025, DOI : 10.3390/computers14100433
- [3] R. Kalan, P. J. Canatalay, E. Karsli, "Security-Aware Adaptive Video Streaming via Watermarking: Tackling Time-to-First-Byte Delays and QoE

- Issues in Live Video Delivery Systems”, Computers, 14(10), 2025,
DOI : 10.3390/computers14100404
- [4] G. Elton, “Features and Benefits to Different Approaches to Watermarking,” 2023.6, BuyDRM,
<https://go.buydrm.com/thedrmblog/features-and-benefits-to-different-approaches-to-watermarking>
- [5] ETSI, “Publicly Available Specification (PAS); DASH-IF Forensic A/B Watermarking – An Interoperable Watermarking Integration Schema”, 2023.8, ETSI TS 104 002 V1.1.1,
<https://dashif.org/docs/IOP-Guidelines/DASH-IF-CTS-00XX-AB-Watermarking-0.9.1.pdf>
- [6] Unified Streaming, “Unified Streaming and Forensic A/B Watermarking,” in “Watermarking”, Unified Streaming Documentation,
<https://docs.unified-streaming.com/documentation/drm/watermarking.html>
- [7] Bitmovin, “Using Nagra NexGuard FileMarker A/B Watermarking”, Bitmovin Encoding Documentation,
<https://developer.bitmovin.com/encoding/docs/using-nagra-nexguard-filemarker-a-b-watermarking>
- [8] R. Fielding, M. Nottingham, J. Reschke, “HTTP Caching”, 2022.6, RFC 9111,
DOI : 10.17487/RFC9111
- [9] I. J. Cox, J. Kilian, F. T. Leighton, T. Shamoon, “Secure spread spectrum watermarking for multimedia”, IEEE Transactions on Image Processing, 6(12), pp.1673–1687, 1997.12,
DOI : <https://doi.org/10.1109/83.650120>
- [10] F. A. P. Petitcolas, R. J. Anderson, M. G. Kuhn, “Attacks on Copyright Marking Systems”, Information Hiding: Second International Workshop, IH98, pp.218–238, Springer, 1998,
DOI : 10.1007/3-540-49380-8_16

저 자 소 개



이재욱(Jewook Lee)

2022.2 평택대학교 정보통신학과 졸업
2026.3-현재: 숭실대학교 인공지능학과 석사과정
<주관심분야> 전자상거래 기술, 컴퓨터네트워크



이지은(Jieun Lee)

12023.2 경운대학교 항공정보통신학과 졸업
2023.3-현재 : 숭실대학교 컴퓨터학과 석박통합과정
<주관심분야> 컴퓨터네트워크, 인공지능, 정보 보안, 저작권 보호 및 이용 활성화 기술 개발



박병찬(Byeongchan Park)

2015.2 평생교육원 학점은행 졸업
2018.2 숭실대학교 컴퓨터학과 석사
2023.8 숭실대학교 컴퓨터학과 박사
2023.9-현재 : 숭실대학교 컴퓨터학부 초빙교수
<주관심분야> 저작권 보호 및 이용 활성화 기술 개발



신용태(Yongtae Shin)

1985.2 한양대학교 산업공학과 졸업
1990.12 Univ. of Iowa, Computer Science 석사
1994.2 Univ. of Iowa, Computer Science 박사
1995.3-현재 : 숭실대학교 컴퓨터학부 교수
<주관심분야> 컴퓨터네트워크, 분산 컴퓨팅, 인터넷프로토콜, 전자상거래 기술