프레임 드로핑을 이용한 MPEG-4 미디어 전송 시의 디지털 저작권 관리를 위한 암호화 기법 연구

양승미O 신동규,신동일,박세영
세종대학교
pluto8410@gce.sejong.ac.kr, sypark@gce.sejong.ac.kr,shindk@sejong.ac.kr,dshin@gce.sejong.ac.kr

Encryption Scheme for Digital Right Management in MPEG-4 Media Transmission Exploiting Frame Dropping

Seungmi YangO Seyoung Park,DongKyoo Shin, Dongil Shin
Sejong University

요 약

본 논문에서는 대역폭의 감소를 가능하게 하는 프레임 드로핑을 MPEG-4파일 포맷을 따르는 미디어에 적용 했을 때, 암호화를 통하여 저작권을 보호할 수 있는 방법을 제안하였다. 프레임 드로핑은 특정 비디오 프레임을 제거함으로써 서버의 과부하를 줄이는 방법이다. 프레임 간의 종속성이 가장 적은 B 프레임을 먼저 제거하고 종속성의 관계에 따라 I, P 프레임 순서대로 제거한다. 드로핑 된 파일을 서버에 저장후 전송하는 방법과 전송 시에 실시간 드로핑을 이용하여 전송하는 두 가지 방법을 설계하였다.

MPEG-4 데이터의 암호화에는 3가지 방법을 설계하였고, 블록암호화 알고리즘은 DES(Data Encryption Standartd)를 사용하였다. 이러한 방법으로 MPEG-4 스트리밍 서비스 시에 서버의 DRM 솔루션을 구현하고, 최적의 방법을 선택하기 위해 드로핑, 암호화, 복호화 및 영상의 품질을 비교하였다.[1)]

1. 서 론

최근 컴퓨터 기술의 급속한 발전과 인터넷 및 무선통신의 속도가 빨라짐에 따라서 다양한 멀티미디어 콘텐츠 배포를 위한 고품질의 서비스가 제공되고 있다. 이 서비스들 중 MPEG-4를 사용한 스트리밍 서비스(Streaming Service)는 유무선 모두에 적합한 방송 솔루션 중 하나이며 멀티미디어 표준을 제정한 MPEG(Moving Picture Experts Group)에서 개발하였고 종합 멀티미디어 부호화 규격을 목적으로 하고 있다 [1].

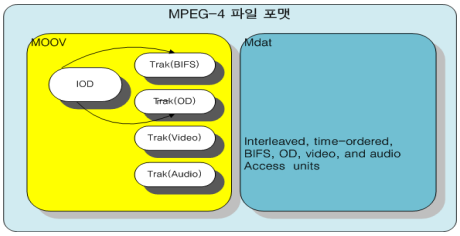
본 논문에서는 네트워크 상황에 따라 통신망의 과부하를 줄여주고 효과적인 동영상 스트리밍 서비스를 위해 프레임 드로핑(Frame Dropping)방법을 적용하고, 보안을 위해 프레임을 암호화 하여 보호 할 수 있는 방법을 제안하였다.

2. 배 경

2.1 MP4 파일 포맷

MP4 파일 포맷은 애플컴퓨터(Apple Computer)의 퀵타임 파일 포맷(QuickTime File Format)을 [2] 기반으로 개발되었고 ISO 미디어 파일 포맷(Part12)에 채용되었다. Part12로부터 MPEG-4용의 파일 포맷으로 파생한 것이 MP4 파일 포맷이다 [3].

MP4 파일은 데이터를 저장하고 있는 데이터 부분과 데이터의 저장 정보 및 기술 디스크립션 정보를 담고 있는 메타데이터 부분으로 나누어진다 [4]. (그림1)은 MP4 파일 포맷(File Format) 구조를 나타낸다.



(그림 1) MP4 파일 포맷

2.2 MPEG 비디오 암호화 기법

나이브 알고리즘은 I.Agi 와 L.Gong이 제안한 알고리즘으로 DES와 같은 표준 암호화 방법에 의한 전체 MPEG 비트스트림을 암호화 하는데 사용되는 일반적이고 간단한 알고리즘이다 [5].

T. B Mayples과 G. A Spanos가 제안한 선택적 알고리즘은 MPEG의 계층화된 구조의 특징을 이용하는 방법들을 선택적 알고리즘으로 분류한다 [6].

Tang은 지그재그 치환 알고리즘과 순수 치환 알고리즘을 제안하였다. 지그재그 치환 알고리즘은 암호화가 지그재그 치환 알고리즘에서 MPEG-4 압축 절차의 필수적인 부분으로 통합된다. 순수 치환 알고리즘은 간단하게 치환함으로서 바이트 스트림을 뒤섞는다 [7].

L. Qiao 와 Nahrstedt가 제안한 비디오 암호화 알고리즘은 MPEG의 압축된 비디오 프레임과 속성의 통계적 분석을

본 연구는 서울시 산학연 협력사업(과제번호 11098)의 지원에 의하여 수행되었음

이용하는 대칭적 암호화 시스템이다 [8].

효과적이고 전체적인 스케일러블 암호화(Efficient and Fully scalable Encryption)는 FGS(Fine Granularity Scalability)는 중간단계에서 암호문의 복호화가 없이 스트림 처리가 이루어지며 Chun Yuan과Yuzhuo Zhong가 제안하였다 [9].

합동 신호 처리와 암호표기법의 멀티미디어 암호화 접근은 Yinian Mao가 제안한 것으로 표준을 보존하고 대표적으로 친숙하게 처리하는 두 개의 암호화를 제안하였다 [10].

Shiguo Lian이 제안한 향상된 비디오 코딩 기반 선택적 암호화 기법은 내부예측 부분암호화 모드(PEM:내부예측을 위한 방법은 블록크기와 함께 바뀌서 인코딩 한다.), 모션벡터 부분암호화(모션벡터를 결정 한 것 에서 모션정보를 비디오 순서로 한다), Coefficients 부분 암호화(내부 매크로블록은 나머지 데이터와 그리고 MVDs 암호스트림과 함께 암호화), 그리고 키 생성을 제안하였다. 압축과정과 암호화 과정은 결합 할 수 있다 [11].

Amir Said가 제안한 부분적인 암호화는 일부 데이터 비트를 암호화시켜서 계산적 복잡성을 감소시킬 수 있다 [12].

2.2 MPEG 비디오 드로핑 기법

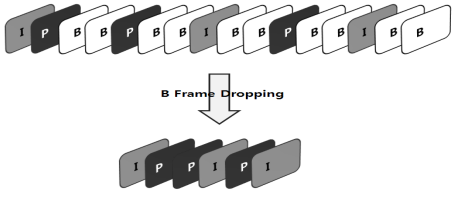
본 논문에서 제안한 프레임 드로핑은 프레임 간의 종속성이 가장 적은 프레임을 제거하여서 서버의 과부하를 줄여주는 방법이다.

블록 드로핑 기법은 W.Zeng, B. LIU가 제안한 방법으로 이미 압축 저장되어 있는 비디오를 전송할 때 I, P, B 별로 프레임별로 블록을 생략 전송하는 방법이다[13].

연속 적인 미디어를 위한 프레임 생략(ACME)는 J. Sandvoss 와 J. Winkler가 제안한 방법으로 시간제약을 어기는 경우 전송되는 프레임의 일부를 스키핑또는 퍼징하는 기법이다 [14].

3. 프레임 드로핑 설계 및 구현

본 논문에서는 프레임 드로핑(Frame Dropping)을 구현하였다. 프레임 드로핑은 I 프레임, P 프레임, B 프레임 중에 중요도와 다른 프레임에 비해 의존도가 떨어지는B 프레임을 제거하여 최상의 품질을 유지하고 비트율을 낮춘다. (그림 2)는 프레임 드로핑의 전반적인 구조를 나타낸다.



(그림 2) B 프레임 드로핑

본 논문에서는 서버에 드로핑 저장되어 있는 미디어 파일을 전송하는 방법과 서버에 드로핑 되어 있지 않는 파일을 저장하고 있다가 전송 시에 실시간으로 드로핑 해서 전송하는 두 가지 방법을 구현하였다.

1) 드로핑 되어 있는 파일을 전송하는 방법*(구현-1)*
2) 전송 시에 실시간으로 드로핑 해서 전송 하는 방법*(구현-2)*

3.1 드로핑 되어있는 파일을 전송 하는 방법*(구현-1)*

미리 서버에 드로핑 되어있는 파일을 만들어 놓고 클라이언트에게 기존의 파일을 변경 없이 전송 하는 방법이다. 각 프레임과 atom들은 서로 유기적으로 연결되어 있어 하나의 B프레임을 삭제하면 연결된atom의 정보를 함께 수정해주어야 한다. 본 연구에서는 각각의 atom의 정보를 수정하기 위해서 MPEG 미디어 스트림의 atom 데이터에 접근하였다. 드로핑 시킬 파일을 만드는 방법은 우선 B 프레임을 드로핑하고 난 후 헤더정보를 변경하고 필요 없어진 Mdat부분을 잘라내어 만든다.

이 방법에서 수정해야 하는 atom은 Stsz, Stco, Mdat, Stts atom 이다. Stsz는 샘플의 크기, Stco는 청크의 시작위치, Mdat 는 프레임의 데이터를 가지고 있고 Stts는 프레임의 재생시간 정보를 가지고 있다. 프레임 드로핑은 아래와 같은 순서로 B 프레임을 먼저 드로핑하고 헤더정보를 변경시켰다.

1) B 프레임 드로핑
2) Stsz atom 정보 수정
3) Stco atom 정보 수정
4) Mdat atom 정보 수정
5) Stts atom 정보 수정

3.1.1 B 프레임 드로핑 알고리즘

Mdat의 데이터 필드에는 실제 미디어 데이터가 포함되어있다. MPEG-4 비디오 표준[4]을 따라 비디오 데이터에서 각 프레임을 분류해낼 수 있다. 각 프레임을 추출하여 드로핑 하고자 하는B 프레임일 경우 프레임을 삭제시키고, I와 P 프레임일 경우에는 삭제시키지 않은 상태로 보존시킨다.

각 프레임에 접근하기 위해서는 프레임의 시작위치가 필요하고 프레임의 시작위치 정보는Stco와 Stsc atom을 가지고 알 수 있다. 각 프레임을 이동시키기 위해서는 프레임의 크기도 알아야 하는데 프레임의 크기는Stsz에 저장 되어있는 샘플의 크기를 통해 얻을 수 있다. B프레임을 드로핑 하기 위한 절차는 다음과 같다.

1) 비디오 미디어의 atom 정보 중 Mdat atom에 접근한다.

2) 첫 번째 프레임부터 순차적으로 각 프레임을 접근하여 프레임의 정보를 추출한다

3) B프레임인 경우 해당 프레임의 인덱스를 저장하고 다음 프레임에 접근한다

4) B프레임 경우에 다음 프레임으로 이동하며I 또는 P프레임일 경우에는 해당 저장되어 있는 인덱스의 위치로 이동한다

B 프레임을 드로핑 시키고I와 P프레임을 해당 위치로 이동시킨다고 하여도 프레임과 관련된atom정보를 수정하지 않으면 많은 문제점을 발생시키게 된다.(그림 3)의 (a)는 원래의 영상이고(그림 3)의 (b)는 드로핑 알고리즘을 적용시킨 결과를 보여준다

(a)드로핑전 원본파일 (b)드로핑 적용 후 파일

(C)stsz atom 정보 수정 (d) stco atom 정보 수정

(그림 3) 프레임 드로핑 알고리즘 적용 후 헤더정보 수정 결과

3.1.2 Stsz Atom 정보 수정

3.1.1의 프레임 드로핑 알고리즘만을 사용하면(그림 3)의 (b)에서 보이는 것과 같이 잘못 된 정보를 재생시킨다. 따라서 샘플 크기의 값을 나타내는 Stsz atom 정보를 수정해 주어야 한다. Stsz atom을 수정해주지 않으면 드로핑 된 프레임의 샘플크기는 이전의B프레임의 샘플 크기를 나타낸다. 이는 새롭게 복사되어 이동 된 I와 P프레임의 크기와 다르기 때문에 문제를 발생시키게 된다.

Stsz를 수정해 주지 않을 경우 프레임의 크기가 다를 수 있는 두 가지의 경우가 있을 수 있다. 첫 번째는 새로운 I와 P프레임의 크기가 기존B프레임의 크기보다 작을 경우이고, 두 번째는 이와 반대로 I와 P프레임의 크기가 기존 B프레임의 크기보다 큰 경우이다 첫 번째의 경우 덮어씌워진 새로운 프레임이외에 B프레임의 정보가 남아있게 되어 프레임을 재생시킬 때 기존의B프레임 중 일부의 데이터도 함께 재생된다 두 번째 경우에는 할당된 프레임의 크기보다 새로운 프레임의 크기가 커서 다음 프레임의 데이터영역을 침범하게 되고 이어지는 다음 프레임이 잘못 된 데이터를 재생 시키게 된다 따라서 Mdat의 프레임 정보와 Stsz atom 테이블 정보를 함께 수정해주어야 한다.

(그림 3)의 (c)는 프레임 드로핑 알고리즘을 적용시킨 후에 Stsz의 atom정보를 수정시켜준 결과이다 Stsz의 atom을 수정하여도 문제가 계속 발생함을 보여준다

3.1.3 Stco Atom 정보 수정

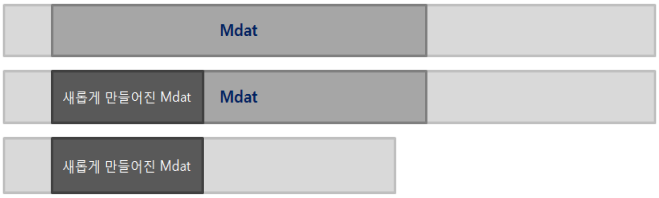
3.1.2절의 Stsz atom정보를 수정한 후에도 발생되는 문제는 기존의 청크의 크기보다 새롭게 구성된 청크의 크기가 작거나 크기 때문이다 이러한 문제를 해결하기 위해서는 청크의 위치를 나타내주는 Stco atom 정보를 수정 해 주어야 한다.

Stco는 청크안의 샘플의 크기가 변하게 되면 변하게 된 만큼 다음 청크의 시작위치 또한 변하게 되기 때문에 Stsz atom의 정보를 수정해주게 되면Stco atom 정보도 함께 수정해주어야만 한다. (그림 3)의 (d)는 Stco atom 정보를 수정한 결과를 보여준다

3.1.4 Mdat Drop

(그림 3)에서 보는 바와 같이 Stco 정보를 변경하고 난 후에 동영상을 재생시켜보면 원래의 미디어와 같은 깨끗한 화면이 재생됨을 알 수 있다

드로핑 알고리즘을 사용하면I 프레임과 P프레임을 이전에 존재하는 B프레임의 위치로 이동시켰기 때문에 화면의 속도가 굉장히 빠르게 재생된다 실제 실험에서 1시간 2분의 미디어를 사용하였지만 드로핑 알고리즘을 적용시킨 이후의 미디어는 12분 30초 만에 재생됨을 볼 수 있었다. 또한, 미디어의 마지막 장면이 재생된 이후에도 미디어가 계속 재생되는 문제점 또한 볼 수 있었다 이는 실제 화면에 보이는 프레임 데이터를 가지고 있는 Mdat atom에 (그림 4)의 두 번째 그림처럼 프레임 드로핑 알고리즘을 적용한 후 이동하기 전의I, B, P 프레임이 남아있기 때문이다 이 문제를 해결하기 위해서는 필요 없는 Mdat atom정보를 삭제 시켜야 한다 (그림 4)의 세 번째 그림은 Mdat이후의 atom을 새롭게 만들어진 Mdat의 이후로 이동시켜 놓음으로써 필요 없는 Mdat를 삭제 시키는 것을 보여준다

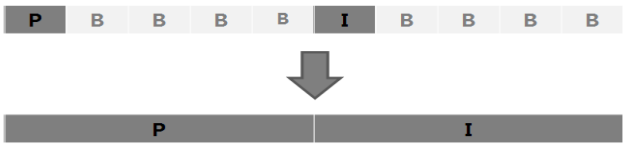


(그림 4) Mdat atom 정보를 수정하는 방법

3.1.5 Stts Atom 정보 변경

3.1.4에서 전술 한 바와 같이 B 프레임을 삭제시키고 나면 삭제 시킨 만큼의 재생시간이 줄어들게 된다. 클라이언트가 미디어에 접속해서 재생 시킬 경우에는 원래의 영상과 같은 속도로 재생이 되어야 하므로, 기존의 미디어와 동일한 재생시간을 갖기 위해서 프레임의 재생시간 정보를 가지고 있는 Stts atom의 정보를 수정해야 한다. Stts atom은 미디어 샘플(프레임)의 생존시간 정보를 저장한다. Stts atom을 수정함으로써 삭제된 만큼의 프레임 재생시간을 보상시키게 된다.

연속적으로 삭제되는 B프레임의 재생시간을 합쳐서 삭제되기 바로 이전의 I 또는 P프레임의 재생시간에 더해주어 I 또는 P프레임이 삭제된 B프레임의 시간동안에도 재생되게 만드는 것이다. (그림 5)은 B프레임을 삭제 후 I 와 P프레임의 재생시간 정보를 수정한 그림이다.Stts atom 정보를 수정하게 되면 새로 만들어진 미디어는 기존의 미디어보다 하나의 프레임 재생시간이 길어지기 때문에 실제 미디어에서 보면 화면이 약간 느리게 보일 수 있지만, 이번 실험에서의 결과에서 실제 재생에는 큰 영향을 주지는 않았다.



(그림 5) Stts atom 정보 수정 결과

3.2 전송 시에 실시간으로 드로핑 하여 전송 하는 방법*(구현-2)*

데이터 파일을 전송 시에 B 프레임을 실시간으로 드로핑 시켜서 전송하는 방법이다. 이 방법은 위의 4.1의 방법과 달리 헤더정보 변경이 불필요하다. 실시간으로 B 프레임을 드로핑 하기 위한 알고리즘은 먼저 프레임 전송 시에 B프레임은 제외시켜 전송시키고, B 프레임이 재생되어야 할 시점에 클라이언트에게 전송되어 있는 I와 P프레임을 재생 시키는 방법이다.

4. 프레임 드로핑 시의 암호화 방안

드로핑이 완료된 MPEG-4의 미디어 파일을 암호화 하여 파일의 크기가 작고 암호화된 파일을 구현하였다.

(a)암호화 전 (b)I-VOP 암호화

(c) P-VOP 암호화 (d)I-VOP와 P-VOP 모두 암호화

(그림 6) 드로핑 된 미디어 파일을 암호화한 결과

MPEG-4 포맷(format)과 헤더구조를 분석하고, I-VOP, P-VOP를 추출 하여 암호화 하고자 하는 프레임을 선택적 암호화 한다. 매크로 블록과(MB) 모션벡터(MV)를 DES를 이용하여 암호화 한다.

암호화는 아래와 같은 3가지 방법을 적용하였다.

1)I-VOP 내의 매크로 블록(MB) 암호화*(적용-1)*

2)P-VOP 내의 매크로 블록과 모션벡터 암호화*(적용-2)*

3)I-VOP의 내의 매크로블록과 P-VOP내의 모션벡터 동시 암호화*(적용-3)*

4.1 I-VOP내의 매크로 블록 암호화*(적용-1)*

I-VOP를 데이터에서 추출하고 난후, DES알고리즘을 적용해서 비디오를 암호화 할 수 있다. DES 알고리즘은 입력으로 64비트 데이터를 가지고, 암호화된 64비트 데이터를 출력하므로 입출력 데이터의 크기에는 변화를 주지 않는다 [15]. 다른 암호화를 사용하게 되면 오프셋이 바뀌고, 크기와 구조에 변하게 생기기 때문에 스트림 암호화를 위해서는 DES와 같은 대칭형 암호화 알고리즘을 사용해야하고, 파일 크기가 변하지 않도록 하기 위해서 DES 입력 값으로 64배수를 적용해야하며, DES 함수의 덧붙임이 일어나는 것을 방지한다. 위의 (그림 6)의 (a)는 암호화하기 전의 원본 영상이고(b)는 매크로 블록을 암호화 한 결과이다.

4.2 P-VOP내의 매크로 블록과 모션벡터 암호화*(적용-2)*

MPEG-4 파일 구조와 헤더구조를 분석해서 P-VOP를

추출하고 난후 P-VOP내의 매크로 블록(MB)과 모션벡터(MV)들을 I-VOP매크로블록을 암호화 한 것처럼DES를 사용하여 암호화 할 수 있다. (그림 6)의 (c)는 P-VOP내의 매크로 블록과 모션벡터를 암호화 한 결과를 보여준다.

4.3 I-VOP내의 매크로블록과 P-VOP내의 모션벡터 암호화(*적용-3*)

매크로 블록들이 많은 코딩 정보들을 가지고 있기 때문에 I-VOP 매크로 블록 암호화와 P-VOP내의 매크로블록 암호화 와 모션벡터 암호화는 모두 완전히 만족할 만한 결과를 보여주지 않는다. 이 문제를 해결하기 위해서 I-VOP 내의 매크로 블록들과, P-VOP내의 모션벡터를 모두 다 암호화하는 방법을 사용할 수 있다 (그림 6)의 (d)는 그 결과를 나타내준다. 이것이 3가지 방법 중 가장 좋은 암호화 방법이라는 것을 보여준다. 하지만 암호화를 위해 처리해야하는 데이터의 양이 위해2가지에 비해서 많다는 단점이 있다.

5. 드로핑과 암호화의 각 구현 방안에 대한 수행시간 속도 실험.

본 연구에서는 MPEG-4 데이터를 위한 두 가지의 드로핑 방법과 세 가지의 암호화 방법을 제안하였다. 실험에 사용된 데이터 파일은 PrettyWomen.mp4 파일을 사용하였으며, 이 데이터 파일의 특성은 <표1>에 기술하였다.

<표1> PrettyWoman.mp4 파일의 특성

Data File Name	PrettyWoman.mp4	
Size	172MB (180,792,022 Bytes)	
Number Of Samples	Total	89916
	I-VOP	10403
	B-VOP	71932
	P-VOP	7581
Bitrate	400kps	
Duration	1:02:34 seconds	
Framerate	89916 fps	
Width x Height	736 x 384	

5.1 드로핑에 대한 실험결과

본 연구에서 제안한 2가지의 드로핑 방법 중 *구현-1*의 장점은 파일의 크기를 줄여서 서버에 보관하기 때문에 서버를 최적화 시킬 수 있다는 것이고, *구현-2*는 파일의 헤더구조 및 원본 데이터 손상 없이 서버에 보관 할 수 있다는 것이 장점이다.

*구현-1*은 드로핑 시킨 파일의 크기가 크게 작아짐을 볼 수 있다. <표2>는 실험 후 생성되는 영상의 크기 변화를 보여준다. 기존의 크기보다 약 58%의 크기의 영상 크기를 얻을 수 있다.

<표 2> Mdat atom 수정 이후 파일 크기의 변화

구분	파일 크기
Mdat 수정 이전의 영상	172MB (180,792,022 바이트)
Mdat 수정 이후의 영상	100MB (105,749,629 바이트)

B 프레임을 드로핑 시키고 미디어 스트림atom의 정보를 변경 하여 얻은 영상은 기존의 영상과 차이가 없음을 보여주었으며, 각 atom을 변경하는데 걸리는 시간 또한 무시 할 수 있을 정도로 작음을 볼 수 있다. <표 3>은 atom 수정에 걸리는 시간을 표로 나타내고 있다.

<표 3> atom 수정에 걸리는 평균 및 전체 시간
(시간 단위 : 밀리 초, millisecond)

Data File	ATOM	샘플수	샘플 당 평균 수정시간	전체 VOP 수정시간
PrettyWoman.mp4	Stco	89916	0.012	211.936
	Stsz		0.010	173.359
	Stts		0.010	182.170

*구현-2*는 atom정보를 수정하지 않고 이미 전송된I와 P 프레임을 B 프레임 대신 재생 시키는 방법이므로I와 P 프레임을 다시 재생 시키는 시간만이 소요된다.<표 4>는 I와 P프레임을 다시 재생 시키는데 걸리는 시간을 나타내고 있다.

<표 4> 실시간 프레임 드로핑(구현-2) 시간 측정 결과
(시간 단위 : 밀리 초, millisecond)

Data File	VOP	샘플 수	샘플 당 평균 재 재생시간	전체 VOP 재 재생시간
PrettyWoman.mp4	I-VOP	10403	0.009	98.95
	P-VOP	7581	0.008	65.75

두 가지 방법 모두 파일의 크기를 줄여서 전송하기 때문에 통신망의 과부하를 줄일 수 있었고, 성능상의 별다른 차이를 보이지 않았다.

5.2 암호화에 대한 실험결과

본 연구에서는 MPEG-4 데이터를 위한 3가지 암호화 방법을 제안 하였는데 그 중 방법이 가장 뛰어난 암호화는 *적용-3*이 가장 뛰어난 암호화 결과를 보여준다. 그러나 암호화 과정에서 처리해야 하는 데이터의 양이 크다는 단점을 보여주게 된다. 이러한 단점을 극복하고 속도

를 향상시키기 위해서I-VOP의 빈도수를 조정하여 암호화에 사용되는 데이터의 양을 조절할 수도 있지만 일반적으로 I-VOP의 매크로 블록들만 암호화 하는 작업에 필요한 데이터의 양은 크지가 않다. VOP를 암호화 하는 시간은 아래 수식과 같다.

$$E(t) = DES(t) + M(t)$$

*E(t)*는 VOP의 암호화 작업을 처리하는 시간이고 *DES(t)*가 암호화 작업을 처리하는 시간이며 *M(t)*는 전처리 시간이다.

비록 *적용-3*이 *적용-1* 보다 암호화 및 복호화 작업에 배 정도의 시간소요를 요구하지만 일반적인 MPEG 클라이언트는 복호화, 디코딩, 렌더링 작업 모두를 메모리나 스왑영역의 전처리 버퍼에서 처리한 후 재생하기 때문에 실제 재생 시간 자체에는 큰 영향을 주지 않는다. <표 5>는 암호화의 각 방법에 따라PrettyWoman.mp4 파일을 암호화 시킨 결과이다.

<표 5> 3 가지 암호화 방법의 속도 측정 결과
(시간 단위 : 초, second)

Data File	ATOM	샘플수	샘플 당 평균 암호화 시간	전체 VOP 암호화 시간
PrettyWoman.mp4	적용-1	10403	0.0012	12.1149
	적용-2	79513	0.0002	19.1956
	적용-3	89916	0.0004	39.1413

6. 결론

본 논문에서는 서버의 과부하를 줄이기 위해서2가지의 드로핑 방법과 보안을 위해서3가지 암호화 방법을 제안하였다. 본 연구는 고화질의 영상을 재상하면서 서버의 과부하를 줄여서 전송속도를 빠르게 하고, 3가지 암호화 방법으로 인해서 다양한 콘텐츠를 안전하게 이용할 수 있게 하였다.

앞으로는 효과적인 스트리밍 서비스를 위해서 드로핑 방법이 중점 연구 될 것이고, 미디어 데이터의 암호화를 위해 DRM에 관해서 더욱 많은 연구가 진행되어야 할 필요성이 있다.

참고문헌

[1] 고성제, 김종옥 공역 "MPEG-4 의 세계" 브레인 코리아

[2] "QuickTime File Format", Apple Computer, June,2000

[3] http://ko.wikipedia.org/wiki/MPEG-4

[4] "Information technology-Coding of audio-visual objects - part1:System ISO/IEC14496-1:2001",ISO/IEC/SC29/WG11, 2001.

[5] I. Agi and L. Gong, "An Empirical Study of Mpeg Video Transmissions," In Proc. of the Internet Society Symposium on Network and Distributed System Security, San Diego, CA, Feb. 1996, pp. 137-144.

[6] T. B. Maples and G. A. Spanos, "Performance Study of a Selective Encryption Scheme for the Security of Networked, Real-time Video,"in Proc. of 4th International Conf. on Computer Communications and Networks, Las Vegas, Nevada, Sep. 1995.

[7] L. Tang, "Methods for Encrypting and Decrypting MPEG Video Data Efficiently," in Proc. of 4th ACM International Multimedia Conference, Boston MA, Nov. 1996, pp. 219-230.

[8] L. Qiao and K. Nahrstedt, "A New Algorithm for MPEG Video Encryption," in Proc. of The First International Conference on Imaging Science, Systems, and Technology (CISST'97), Las Vegas, Nevada, July 1997, pp. 21-29.

[9] C. Yuan, B. B. Zhu, Y. Wang, S. Li, Y. Zhong, "Efficient and Fully Scalable Encryption for MPEG-4 FGS," IEEE Int. Symp. Circuits and Systems, May, 2003

[10] Y. Mao and M. Wu: "A Joint Signal Processing and Cryptographic Approach to Multimedia Encryption", IEEE Trans. on Image Processing, vol. 15, no. 7, pp. 2061-2075, July 2006

[11] S. Lian, Z. Liu, Z. Ren and Z. Wang, "Selective Video Encryption Based on Advanced Video Coding," PCM 2005, Part II, Springer LNCS, vol. 3768, pp. 281-290, 2005

[12] A. Said, "Measuring the strength of partial encryption schemes," In proceedings of 2005 IEEE International Conference on Image Processing (ICIP 2005), 11-14 Sept., vol.2, pp. 1126-1129

[13] W.Zeng, B. LIU, "Rate Shaping by Block Dropping for Transmission of MPEG Precoded Video over Channels of Dynamic Bandwidth" Multimedia 96 Processing, The Fourth ACM Internatnional Multimedia Conference, Boston Ma.pp129-140 1996

[14] L.Delgrossi,C.Halstrick,D.hehmann, R.G.Herrtwich, O.Krone, J. Sandvoss, and C.Vogt, "Media Scaling for Audiovisual Communication with the Heidelberg Transport System", Proceedings ACM Multimedia, 1993

[15] Data Encryption Standard (DES), FIPS PUB 46-3, Oct. 25. 1999