DOI: 10.3745/KIPSTB.2008.15-B.6.575

대역폭 감소를 적용한 MPEG-4 미디어 전송시의 암호화 기법 연구

신 동 규[†] · 신 동 일[††] · 박 세 영[†††]

요 약

미디어의 전송시 네트워크의 상황에 따라 통신망의 과부하가 발생할 수 있으며, 이를 줄이기 위해서 필터링, 부하 분산기법, 막힘 제어 기법, 프레임 드로핑 등의 많은 연구가 진행되었다. 이들 중 효과적인 방법은 동영상의 비트율을 조절하기 위해 특정 비디오 프레임을 제거함으로써 대역폭의 감소를 가능하게 하는 프레임 드로핑(Frame dropping)이다. 프레임 드로핑은 프레임 간의 종속성이 가장 적은 B 프레임을 먼저 제거하고 종속성의 관계에 따라 I, P 프레임 순서대로 제거한다.

본 논문에서는 MPEG-4 미디어의 전송에 프레임 드로핑을 적용 하였으며 이때 암호화를 통하여 저작권을 보호할 수 있는 방법을 제안한다. 이를 위하여 서버에 저장되어있는 프레임 드로핑 이 이미 적용된 파일을 클라이언트에게 전송하는 방법과 서버에 저장되어있는 미디어 파일을 클라이언트에게 전송시 실시간으로 프레임 드로핑하는 두 가지 방법을 설계 구현 하였다. MPEG-4 데이터의 암호화에는 3가지 방법을 제안하였다: I-VOP내의 매크로 블록(Macro block) 암호화, P-VOP내의 매크로 블록과 모션벡터 암호화(Motion Vector), I-VOP내의 매크로블록과 P-VOP내의 모션벡터 암호화. MPEG-4 미디어의 전송시, 최적의 방법을 선택하기 위해 드로핑, 암호화, 복호화 및 영상의 품질을 비교하였으며 드로핑 후에도 원래의 영상과 큰 차이가 없었다. 암호화와 복호화에서는 I-VOP와 P-VOP를 모두 암호화 하였을 때가 가장 성능이 좋았다.

키워드 : MPEG(Moving Picture Experts Group)-4, 프레임 드로핑(Frame dropping), 암호화(Encryption)

Encryption Scheme for MPEG-4 Media Transmission Exploiting Frame Dropping

Dongkyoo Shin[†] · Dongil Shin[††] · Seyoung Park[†††]

ABSTRACT

According to the network condition, the communication network overload could be occurred when media transmitting. Many researches are being carried out to lessen the network overload, such as the filtering, load distributing, frame dropping and many other methods. Among these methods, one of effective method is frame dropping that reduces specified video frames for bandwidth diminution. B frames are dropped and then I, P frames are dropped according to dependency among the frames in frame dropping.

This paper proposes a scheme for protecting copyrights by encryption, when we apply frame dropping to reduce bandwidth of media following MPEG-4 file format. We designed two kinds of frame dropping: first one stores and then sends the dropped files and the other drops frames in real-time when transmitting. We designed three kinds of encryption methods in which DES algorithm is used to encrypt MPEG-4 data: macro block encryption in I-VOP, macro block and motion vector encryption in P-VOP, and macro block and motion vector encryption in I, P-VOP. Based on these three methods, we implemented a digital right management solution for MPEG-4 data streaming. We compared the results of dropping, encryption, decryption and quality of video sequences to select an optimal method, and there is no noticeable difference between the video sequences recovered after frame dropping and the ones recovered without frame dropping. The best performance in encryption and decryption of frames was obtained when we apply the macro block and motion vector encryption in I, P-VOP.

Keywords : MPEG(Moving Picture Experts Group)-4, Frame Dropping, Encryption

1. 서 론

오늘날 유비쿼터스 환경에서 인터넷 및 무선 통신이 발달함에 따라 사용자가 서비스에 대한 요구가 다양해지고 서비스의 품질에 대한 요구 또한 높아지고 있다. 이러한 클라이언트의 요구를 만족시키기 위한 디지털 콘텐츠 배포를 위해 오늘날 고품질의 서비스가 제공되고 있다. 특히, 멀티미디어 서비스의 등장으로 인해 화상 전송과 같은 넓은 대역폭을 요구하는 서비스와 방송 시스템과 같은 실시간 서비스가 제공되고 있다. 이러한 서비스들 중 MPEG-4를 이용한 스트

※ 본 연구는 서울시 산학연 협력사업(과제번호 11098)의 지원에 의하여 수행되었음.
† 종신회원 : 세종대학교 컴퓨터공학과 교수
†† 종신회원 : 세종대학교 컴퓨터공학과 부교수
††† 준 회 원 : 세종대학교 컴퓨터공학부
논문접수 : 2008년 8월 26일
수 정 일 : 1차 2008년 10월 1일
심사완료 : 2008년 10월 2일

리밍 서비스(Streaming Service)는 유무선 모두에 적합한 솔루션 중 하나이다. MPEG(Moving Picture Experts Group)에서 개발한 MPEG-4는 기존의 영상, 오디오 신호의 압축 부호화와 더불어 정지영상, 컴퓨터 그래픽스, 분석 합성 계의 음성 부호화, MIDI 등에 의한 합성 오디오 및 텍스트를 포함하는 종합 멀티미디어 부호화 규격으로 구성되어 있다[1]. 품질이 높은 스트리밍 서비스의 제공을 위해서는, 네트워크 상황에 따라 발생할 수 있는 통신망의 과부하를 줄여서 끊김 없는 서비스가 제공 되어야 한다. 통신망의 과부하를 줄이기 위해서 필터링, 부하 분산기법, 혼잡제어 알고리즘 등의 많은 연구가 있었다. 하지만, 미디어를 하나의 서버에 저장하는 중앙집중식 환경이 아닌 여러 개의 서버에 분산 저장하여 통신망의 과부하를 막는 부하 분산기법은 대규모의 저장 공간이 필요하게 된다[2]. 혼잡제어 알고리즘 중 TCP 혼잡 제어 방식은 끊김 없는 멀티미디어 스트리밍 서비스에 적합하지 않으므로 UDP 상에서 작동하는 TFRC (TCP Friendly Rate Control)기법을 사용한다[3]. 하지만, TFRC는 전송률이 일정할 때 적합하기 때문에, 스트리밍서비스와 같이 트래픽의 예측을 하기 힘든 경우에는 많은 패킷 손실이 발생 할 수 있다[4].

본 논문에서는 이전에 연구들에서의 단점을 고려하여 프레임간의 종속성이 가장 낮은 B프레임을 네트워크 상황에 따라 드로핑 시켜서 미디어의 대역폭을 조절할 수 있는 프레임 드로핑(Frame Dropping)방법을 설계 적용하였다. 이러한 프레임 드로핑을 이용한 스트리밍 서비스의 제공을 위해서는 미디어를 보호할 수 있는 방법이 필요하다. 미디어의 지적 재산권을 보호하기 위해 디지털 저작권 관리(DRM: Digital Right Management)시스템은 디지털 콘텐츠의 사용, 재생시간, 재생횟수, 회람, 저장 등을 제한한다. 이런 DRM 방법들을 표준화 하고 관리하는 ISO의 MPEG에서는 이러한 저작권 관리 서비스를 위해서 지적 재산권 관리보호(IPMP: Intellectual Property Management and Protection) 인터페이스 규격을 MPEG-4 표준화 부분에 추가 제정하였다[5, 6]. 본 논문에서는 DRM 시스템을 구현하기 위해서 암호화를 사용하였고, 암호화 알고리즘은 입력으로 64bit 데이터를 이용하여, 암호화된 64bit 데이터를 출력하는 DES (Data Encryption Standard) 블록 암호화 알고리즘을 사용하여 헤더구조의 변화가 없는 프레임의 암호화 방법을 제안하였으며, 프레임 드로핑을 이용하여 파일사이즈를 축소하여 헤더구조를 변화시키고 암호화 하는 방법도 제안하였다.

2. 관련연구

통신망에서의 콘텐츠 전송을 위해서는 높은 대역폭이 필요하지만, 통신망의 환경에 따라서 대역폭의 변화가 크다. 이러한 대역폭의 변화에 따른 동영상 품질의 변화를 막기 위해서는 안정적으로 지원할 수 있을 정도의 낮은 대역폭을 선택하여 사용하여야 한다. MPEG-4는 H.261, MPEG-1, MPEG-2에 비해 낮은 비트율의 압축효율이 현저하게 뛰어나기 때문에 낮은 비트율로 동영상을 제공할 때 주로 사용될 수 있다. 따라서 MPEG-4 기술은 이동통신이나 인터넷 방송 등에서 동영상을 전송할 때 상당히 유용하게 활용되고 있다[7,8]. 기존에 연구되어온 비디오 프레임 드로핑과 암호화 방법들에 관하여 아래와 같이 기술한다.

2.1 전통적인 MPEG 비디오 드로핑 기법

본 논문에서 제안한 프레임 드로핑은 특정 비디오 프레임을 제거함으로써 서버의 과부하를 줄이는 방법이다. 이러한 드로핑 방법 외에도 기존에 사용된 통신망의 부하를 줄이기 위한 여러 가지 방법이 존재한다.

하이델베르크(Heidelberg) 전송 시스템(HeiTS)은 미디어의 전송 단계에서 실행되며 네트워크의 현재 상황을 파악한 다음 유용한 대역폭으로 트래픽을 적용시키는 네트워크의 응답성에 중점을 둔다. 네트워크의 부하 정도에 따라서 시스템은 스케일링 업(Scaling up)과 스케일링 다운(Scaling down)을 수행하게 된다[9,10].

블록 드로핑에 의한 드로핑 전송 알고리즘은 이미 압축 저장되어있는 비디오를 전송할 때 네트워크 대역폭이 감소되는 경우 서버에서 압축된 비디오의 비트율을 감소시켜 전송해야 한다. 이때 I, P, B 프레임 별로 블록을 생략 전송하는 방법이 블록 드로핑 알고리즘이다. 여기서 블록은 매크로 블록(MB)을 말한다[11].

연속적인 미디어를 위한 프레임 생략(Abstraction for Continuous MEdia)은 시간제약을 어기는 경우에 전송될 프레임의 일부를 스키핑 또는 퍼징 하는 기법을 말한다[10,12].

계수 드로핑(Coefficient Dropping)은 중요하지 않은 고주파 DCT 계수의 VLC 코드를 제거하는 방법으로 프레임 드로핑에 비해서 적은 양의 데이터를 세밀하게 제어할 수 있다. EOB(End of Block)코드에서 순차적으로 제거하는 방법과 단일 블록 내에서 위치와 상관없이 중요도가 낮은 계수를 없애는 방법이 있다[13].

계층적 비디오 코딩(Scalable Coding)은 낮은 품질부터 높은 품질까지 다양한 화질의 비디오를 수신 측에서 수신할 수 있도록 계층적인 비트스트림을 만들어서 전송하는 방법이다. 계층적 비디오 코딩은 기본품질의 영상을 전송하는 기본계층과 향상된 영상을 전송하는 강화계층으로 구성된다. 수신 측에서는 수신된 기본계층과 강화 계층의 영상 중에 자신에게 맞는 영상을 복호화 할 수 있게 하여, 다양한 사용자를 만족시킬 수 있는 방법이다[14,15].

2.2 MPEG 비디오 암호화 기법

기존에 사용되어왔던 비디오 암호화 기법들 중 나이브 알고리즘은 I. Agi 와 L. Gong이 제안한 알고리즘으로 DES와 같은 표준 암호화 방법에 의한 전체 MPEG 비트스트림을 암호화 하는데 사용되는 일반적이고 간단한 알고리즘이지만 평문으로 된 MPEG 스트림을 처리하고, MPEG 비트스트림의 구조적 특징을 살리지 못한다는 단점이 있다[16].

T. B Mayples과 G. A Spanos가 제안한 선택적 알고리즘은 MPEG의 계층화된 구조의 특징을 이용하는 방법들을

선택적 알고리즘으로 분류한다[17].

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 데이터 주파수, 다이어그램 주파수 등을 사용하기 위한 암호화 분석의 어려움과 비효율성 때문에, 순수 치환 알고리즘은 간단하게 치환함으로써 바이트 스트림을 뒤섞는다[18].

L. Qiao와 Nahrstedt가 제안한 비디오 암호화 알고리즘은 MPEG의 압축된 비디오 프레임과 속성의 통계적 분석을 이용하는 대칭적 암호화 시스템이다. 이 알고리즘도 DES를 사용하였고 MPEG의 픽쳐층(Picture Layer)에 해당되는 모든 데이터를 암호화 한다[19].

효과적이고 전체적인 스케일러블 암호화(Efficient and Fully scalable Encryption)는 FGS(Fine Granularity Scalability) 중간단계에서 암호문의 복호화가 없이 스트림 처리가 이루어지며 Chun Yuan과 Yuzhuo Zhong가 제안하였다[20].

합동 신호 처리와 암호표기법의 멀티미디어 암호화 접근은 Yinian Mao가 제안한 것으로 표준을 보존하고 대표적으로 친숙하게 처리하는 두 개의 암호화를 제안하였다[21].

Shiguo Lian이 제안한 향상된 비디오 코딩 기반 선택적 암호화 기법은 내부예측 부분암호화 모드(PEM: 내부예측을 위한 방법은 블록크기와 함께 바꿔서 인코딩 한다.), 모션벡터 부분암호화(모션벡터를 결정 한 것에서 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[22].

Amir Said가 제안한 부분적인 암호화는 일부 데이터 비트를 암호화시켜서 계산적 복잡성을 감소시킬 수 있다. 다른 사용자들의 승인을 조건으로 하여 버전 설정을 바꾼다[23].

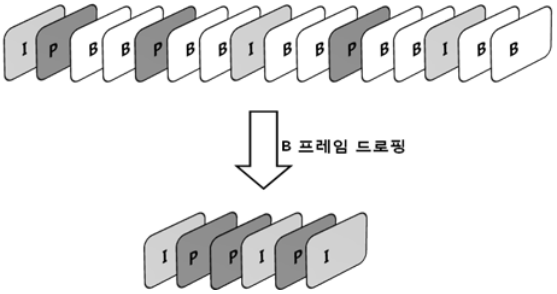
3. 프레임 드로핑 설계 및 구현

본 논문에서는 대역폭의 감소상황을 지원하는 MPEG-4 프레임 드로핑(Frame Dropping)을 구현하고 여기에 암호화 방안을 적용하였다. 프레임 드로핑은 I-VOP (Intra-coded), P-VOP (Predictive-coded), B-VOP (Bidirectional Predictive - coded) 프레임 중에 중요도가 떨어지고 다른 프레임에 비해 의존도가 떨어지는 B-프레임을 제거하여 최상의 품질을 유지하고 비트율을 낮춘다. (그림 1)은 프레임 드로핑의 전반적인 구조를 나타낸다.

본 논문에서는 MPEG-4파일을 전송 전에 드로핑 하여 서버에 저장해 놓고 클라이언트 요청 시 드로핑 되어있는 파일을 전송하는 방법과 서버에 드로핑 되지 않은 기존의 파일을 저장하고 있다가 클라이언트에게 파일을 전송 시 실시간으로 드로핑 하여 전송하는 두 가지 방법을 구현하였다.

(1) 서버의 미리 드로핑 되어있는 파일을 전송 하는 방법 (드로핑-1)

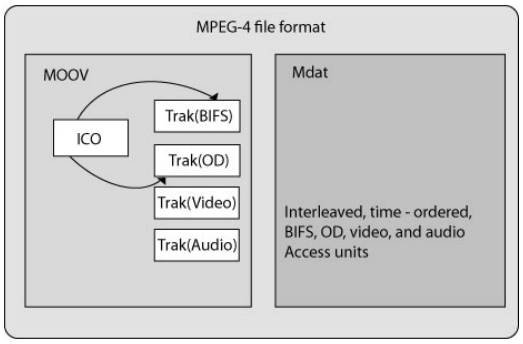
(2) 서버의 원본 파일을 전송 시에 실시간으로 드로핑 해서 전송 하는 방법 (드로핑-2)



(그림 1) B 프레임 드로핑

*(드로핑-1)*의 방법은 파일 포맷의 수정이 불가피한 방법이다. MP4 파일 포맷은 애플컴퓨터(apple computer)의 퀵타임 파일 포맷(QuickTime file format)을[24] 기반으로 개발되었고 ISO 미디어 파일 포맷(part12)에 채용되었다. Part12로부터 MPEG-4용의 파일 포맷으로서 파생한 것이 MP4 파일 포맷이다[1].

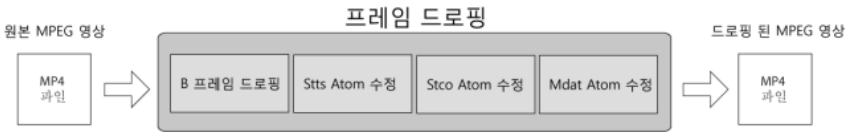
MP4 파일은 데이터를 저장하고 있는 데이터 부분과 데이터의 저장 정보 및 기술 디스크립션 정보를 담고 있는 메타데이터 부분으로 나누어진다[25]. (그림 2)는 MP4 파일 포맷(File format) 구조를 나타낸다. 가장 상위에는 실제 데이터들의 집합인 청크들을 포함하는 Mdat Atom이 존재하고, 그 밑으로 다수의 여러 종류 Atom을 포함하는 MOOV Atom이 존재한다. MOOV Atom 하위의 Atom 들은 실제 데이터에 대한 세부 기술정보를 가지고 있어서 서로 유기적으로 연결되어있다.



(그림 2) MP4 파일 포맷의 예

3.1 서버의 미리 드로핑 되어있는 파일의 전송*(드로핑-1)*

이 방법은 서버에 MP4 파일을 미리 드로핑 시켜 놓고 클라이언트 요청 시 저장되어있는 파일을 변경 없이 전송하는 방법이다. 서버에 드로핑 된 파일을 저장해 놓기 위해 프레임 드로핑을 수행할 때 단순하게 B 프레임만을 삭제하는 것으로 접근하기는 어렵다. 각 atom들은 프레임에 대한 정보를 저장하고 있기 때문에 각 프레임과 atom들은 서로 유기적으로 연결되어있다. 하나의 프레임을 수정하면 연결



(그림 3) 프레임 드로핑 과정

된 atom의 정보들 또한 모두 수정되어야 한다. 이를 위해 본 논문에서는 처음에 B 프레임을 드로핑 하고 난 후 각 atom의 정보를 수정하고 필요 없어진 프레임 데이터를 삭제 하였다. atom의 정보를 수정하기 위해 MPEG 미디어 스트림의 atom 데이터에 접근하였다. 이 때에 수정되어야 하는 atom은 Stsz, Stco, Mdat, Stts atom 이다. Stsz는 샘플의 크기, Stco는 청크의 시작 위치, Mdat 는 프레임의 데이터를 가지고 있고, Stts는 프레임의 재생시간 정보를 가지고 있다. 따라서, 프레임 드로핑은 아래와 (그림 3)과 같이 B 프레임 드로핑, Stsz atom 정보 수정, Stco atom 정보 수정, Mdat atom 정보수정, 그리고 Stts atom 정보 수정 순으로 진행 되었다.

(그림 4)와 (그림 5)는 프레임 드로핑 알고리즘을 적용시킨 결과를 나타내며, 순서대로 원본영상, B프레임을 적용시켰을 때의 영상, Stsz atom 정보 수정 후의 영상, Stco atom 정보 수정후의 영상을 나타낸다.

(그림 4) PrettyWoman.mp4의 드로핑 결과

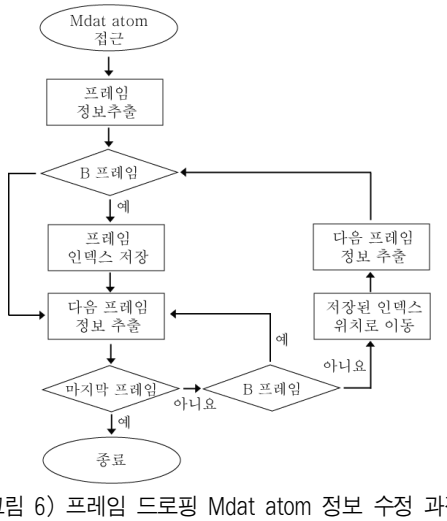
(그림 5) Akiyo.mp4의 드로핑 결과

3.1.1 B 프레임 드로핑

MPEG-4 비디오 표준[24]을 따라 각 VOP는 서로 다른 Video_start_code를 가지기 때문에 비디오 데이터에서 각 프레임을 분류해낼 수 있다. 실제 데이터가 저장되어있는 Mdat의 데이터 필드에서 각 프레임을 분류해내고 프레임이 드로핑 하고자 하는 B 프레임일 경우 프레임을 삭제시키고, I와 P 프레임인 경우에는 삭제시키지 않은 상태로 보존시킨다. 각 프레임에 접근하기 위해서는 프레임의 시작위치가 필요하고 프레임의 시작위치 정보는 Stco와 Stsc atom을 가지고 알 수 있다. 각 프레임을 이동시키기 위해서는 프레임의 크기도 알아야 하는데 프레임의 크기는 Stsz에 저장 되어있는 샘플의 크기를 통해 얻을 수 있다. Mdat atom정보에서 B프레임을 드로핑 하기 위한 절차는 (그림 6)과 같다.

처음 실제 데이터가 저장되어있는 Mdat atom에 접근하여, 프레임의 정보를 추출한다. 추출된 프레임 정보로 프레임이 B프레임인지 확인 한 후에, B프레임이라면 해당 프레임의 인덱스를 저장하고 다음 프레임의 정보를 추출한다. 프레임이 마지막 프레임이 아니라면 다시 어떤 프레임인지 확인하여 B프레임이라면, 바로 다음 프레임으로 이동하고 I 또는 P프레임이라면 저장된 인덱스의 위치로 해당 프레임을 이동시켜서 B프레임을 삭제 시킨다. 그리고 처음으로 돌아가서 마지막 프레임까지 똑같은 작업을 반복한다.

B 프레임을 드로핑 시키고 I와 P프레임을 해당 위치로 이동시킨다고 하여도 (그림 4)와 (그림 5)의 두 번째 그림과 같은 문제점을 발생시키게 된다. 따라서 프레임과 관련된 atom정보를 수정해주어 문제점을 해결해야 한다.



(그림 6) 프레임 드로핑 Mdat atom 정보 수정 과정

3.1.2 Stsz Atom 정보 수정

3.1.1절에서와 같이 단순히 B프레임만을 삭제하고, I와 P 프레임만을 남겨 놓을 경우 잘못된 정보를 재생시키는 이유는 새롭게 복사되어 이동된 I와 P프레임의 크기는 이전의 드로핑 된 B프레임의 크기와 같을 수 없지만 Stsz atom에

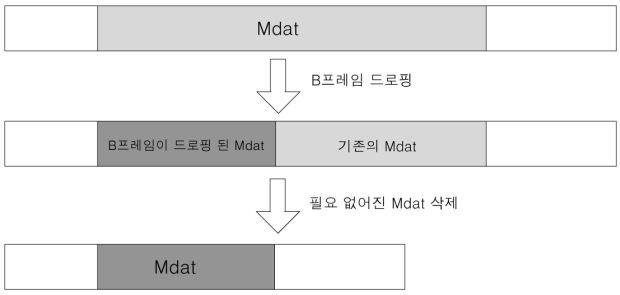
서는 이전의 B프레임 샘플 크기를 저장하고 있기 때문이다. 이러한 문제를 해결하기 위해서, Stsz atom의 정보를 수정하여 새로운 프레임의 크기를 정확하게 나타내도록 하여야 한다. (그림 4)와 (그림 5)의 세 번째 그림은 프레임 드로핑 알고리즘을 적용시킨 후에 Stsz의 atom정보를 수정시켜준 결과이다. 하지만, Stsz atom을 수정하여도 문제가 계속 발생함을 보여준다.

3.1.3 Stco Atom 정보 수정

Stsz atom정보를 수정한 후에도 문제가 발생하는 이유는 미디어 데이터 스트림 안의 청크의 위치 정보를 가지고 있는 Stco atom을 수정해 주지 않아서 발생하는 것이다. 샘플의 크기가 변하여 청크의 크기가 변하게 되면 다음 청크의 시작위치도 변하게 된다. 따라서, Stsz atom의 정보가 수정됨에 따라 Stco atom의 정보도 반드시 수정되어야 한다. (그림 4)와 (그림 5)의 마지막 그림은 Stco atom 정보를 수정한 결과 영상이다.

3.1.4 Mdat Drop

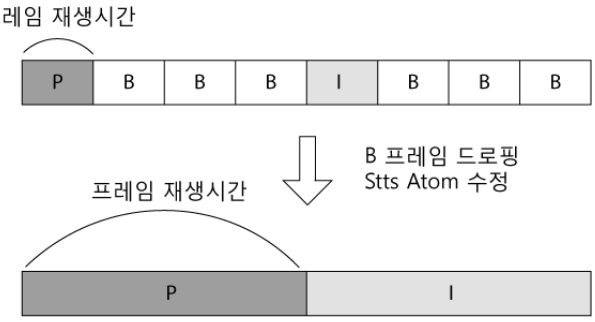
(그림 4)와 (그림 5)에서 보는 바와 같이 Stco 정보를 변경하고 난 후에 동영상을 재생시켜보면 원래의 미디어와 같은 깨끗한 화면이 재생됨을 알 수 있다. 하지만, 미디어의 마지막 장면이 재생된 이후에도 미디어가 계속 재생되는 문제점이 발생한다. 이는 프레임 드로핑을 적용하기 전의 I, B, P프레임이 남아 있기 때문이다. 이 문제의 해결을 위해서 Mdat에서 마지막 프레임 데이터 이후의 데이터는 모두 삭제해야 한다. (그림 7)은 Mdat에서 마지막 프레임 데이터를 삭제하여 Mdat 정보를 수정하는 방법을 나타낸다.



(그림 7) Mdat atom 정보를 수정하는 방법

3.1.5 Stts Atom 정보 변경

B프레임을 드로핑 시키고 나면 삭제 시킨 만큼의 재생시간이 줄어들게 된다. 실제 실험에서 1시간 2분의 재생시간을 가진 미디어를 사용하였지만, 드로핑 알고리즘을 적용시킨 이후의 미디어는 12분 30초 만에 재생이 되는 것을 볼 수 있었다. 하지만, 클라이언트가 미디어에 접속해서 재생시킬 경우에는 원래의 영상과 같은 속도로 재생이 되어야 하므로, 기존의 미디어와 동일한 재생시간을 갖기 위해서 프레임의 재생시간 정보를 가지고 있는 Stts atom의 정보를 수정함으로써 삭제된 만큼의 프레임 재생시간을 보상 시켰다. 본 논문의 실험에서는 연속적으로 삭제되는 B프레임의 재생시간을 합쳐서 삭제되기 바로 이전의 I 또는 P프레임의 재생시간에 더해주어 I 또는 P프레임이 삭제된 B프레임의 시간 동안에도 재생되게 구현 하였다. (그림 8)은 B프레임을 삭제 후 I 와 P프레임의 재생시간 정보를 수정하는 과정을 나타낸다.



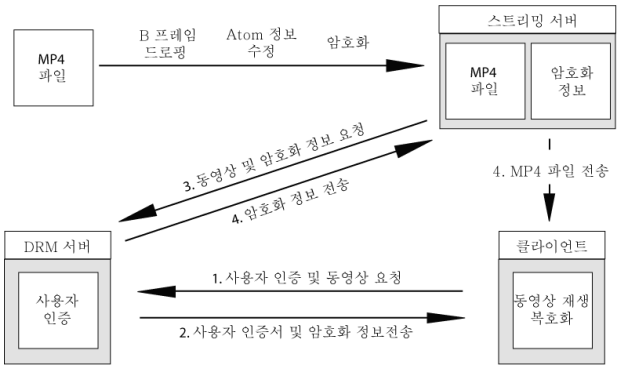
(그림 8) Stts atom 정보 수정 결과

3.2 전송 시에 실시간으로 드로핑 하여 전송 *(드로핑-2)*

서버에 드로핑 되지 않은 기존의 파일을 저장하고 있다가 클라이언트의 요청 시 B 프레임을 실시간으로 드로핑 시켜서 전송하는 방법이다. 이 방법은 위의 서버에 드로핑 된 파일을 저장 시켜 놓은 후 전송하는 방법과 달리 헤더정보 변경이 불필요하다. Mdat의 데이터 필드에서 각 프레임을 분류해내고 프레임이 드로핑 하고자 하는 B프레임일 경우에 제외 시켜서 전송을 하게 된다. 클라이언트 측에서는 B프레임이 재생되어야 할 시점에서 클라이언트에게 미리 전송된 I와 P프레임을 재생시키는 방법이다.

4. 프레임 드로핑 시의 암호화 방안

드로핑이 완료된 MPEG-4의 미디어 파일을 암호화 하여 파일의 크기가 작고 암호화된 파일을 가지고 빠르고 안전한 DRM 솔루션을 설계하고 구현하였다. (그림 9)는 DRM 구조를 나타낸다.



(그림 9) 스트리밍 전송을 위한 DRM의 전반적 구조

MPEG-4 포맷(format)과 헤더구조를 분석하고, I-VOP, P-VOP를 추출 하여 암호화 하고자 하는 프레임의 매크로

블록과(MB) 모션벡터(MV)를 DES를 이용하여 암호화 한다.

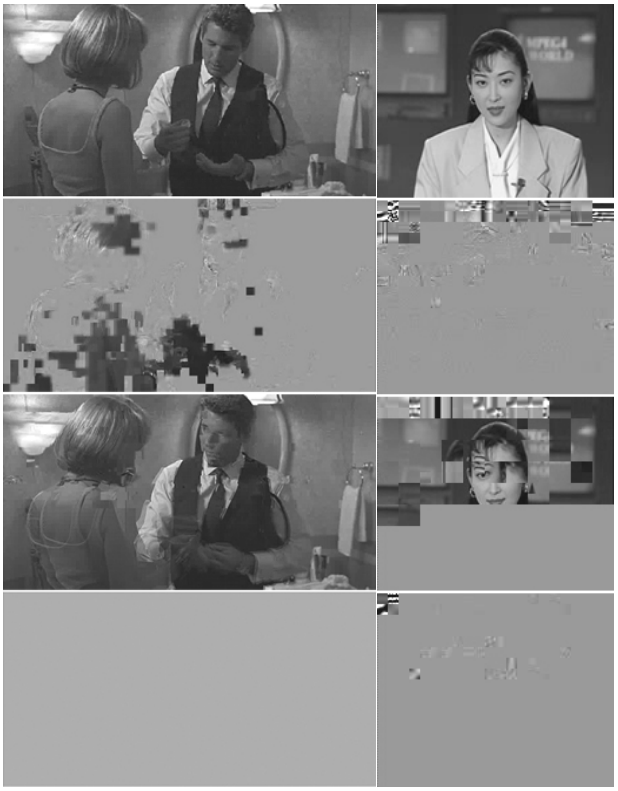
각 VOP는 하나의 Video_start_code를 가지고 있다[26]. Video_start_code의 값은 16진수로 00, 01, B6 이고, vop_coding_type은 I-VOP, P-VOP, B-VOP 타입(type)을 식별한다[26,27].

암호화는 아래와 같은 3가지 방법을 적용하였다.

1) I-VOP 내의 매크로 블록(MB) 암호화*(암호화-1)*
2) P-VOP 내의 매크로 블록과 모션벡터 암호화*(암호화-2)*
3) I-VOP의 내의 매크로블록과 P-VOP내의 모션 벡터 동시 암호화*(암호화-3)*

4.1 I-VOP내의 매크로 블록 암호화 (암호화-1)

I-VOP를 데이터에서 추출하고 난 후, DES알고리즘을 적용해서 비디오를 암호화 할 수 있다. DES 알고리즘은 위에서 언급하였듯이 입력으로 64비트 데이터를 가지고, 암호화된 64비트 데이터를 출력하므로 입출력 데이터의 크기에는 변화를 주지 않는다. 다른 암호화를 사용하게 되면 오프셋이 바뀌고, 크기와 구조에 변화가 생기기 때문에 스트림 암호화를 위해서는 DES와 같은 대칭형 암호화 알고리즘을 사용해야 하고, 파일 크기가 변하지 않도록 하기 위해서 DES 입력 값으로 64배수를 적용해야 하며, DES 함수의 덧붙임이 일어나는 것을 방지한다. (그림 10)과 (그림 11)의 첫 번째 영상은 암호화하기 전의 원본 영상이고 두 번째 영상은 I-VOP의 매크로 블록을 암호화 한 결과이다.



(그림 10) PrettyWoman.mp4 드로핑 후 암호화 결과

(그림 11) Akiyo.mp4 드로핑 후 암호화 결과

4.2 P-VOP내의 매크로 블록과 모션벡터 암호화 (암호화-2)

MPEG-4 파일 구조와 헤더구조를 분석해서 P-VOP를 추출하고 난 후 P-VOP내의 매크로 블록(MB)과 모션벡터(MV)들을 I-VOP매크로블록을 암호화 한 것처럼 DES를 사용하여 암호화 할 수 있다. (그림 10)과 (그림 11)의 세 번째 영상은 P-VOP내의 매크로 블록과 모션벡터를 암호화 한 결과를 보여준다.

4.3 I-VOP내의 매크로블록과 P-VOP내의 모션벡터 암호화 (암호화-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 10)와 (그림 11)의 마지막 영상은 그 결과를 나타내준다. 이것이 3가지 방법 중 가장 좋은 암호화 방법이라는 것을 보여준다. 하지만 암호화를 위해 처리해야 하는 데이터의 양이 위의 2가지에 비해서 많다는 단점이 있다.

5. 드로핑과 암호화의 각 구현 방안에 대한 수행시간속도 실험

본 연구에서는 MPEG-4 데이터를 위한 두 가지의 드로핑 방법과 세 가지의 암호화 방법을 제안하였다. 실험에 사용된 데이터 파일은 영화 2편(PrettyWomen.mp4, Gone with the wind.mp4)과 MPEG4 샘플 영상 7편(Akiyo.mp4, Foreman.mp4, Hall_monitor.mp4, Coastguard.mp4, Container.mp4, New, Stefan.mp4)을 사용을 사용하였으며, 이 데이터 파일의 특성은 <표 1>과 <표 2>에 기술하였다. 암호화와 드로핑 실험을 위한 프로그램은 Windows XP환경에서 모두 C 언어로 구현하였으며, 동영상의 재생 실험은 MPEG4IP 라는 재생 프로그램을 수정하여 사용하였다. 실험에 사용된 데이터 파일은 YUV Sequence File을 FFMPEG를 이용하여 MPEG4 표준 영상으로 변환시켜 사용하였다.

〈표 1〉 실험에 사용된 영화 파일의 특성

데이터 파일 명		PrettyWoman.mp4	Gone with the wind.mp4
파일 크기		172MB)	240MB
샘플의 수	전체	89916	124738
	I-VOP	10403	14432
	B-VOP	71932	99790
	P-VOP	7581	10516
비트율		384Kbps	3475Kbps
길이		1:02:34 초	50:38 초
프레임율		23975 fps	23000 fps
넓이 x 높이		736 x 384	640 x 464

〈표 2〉 실험에 사용된 MPEG4 샘플 영상 특성

데이터 파일 명		Akiyo.mp4	Foreman.mp4	Hall_monitor.mp4	Coastguard.mp4	Container.mp4	News.mp4	Stefan.mp4
파일 크기(KB)		111	4145	3426	528	2336	207	798
샘플의 수	전체	100	400	300	200	500	100	200
	I-VOP	9	35	26	18	44	9	18
	B-VOP	66	265	199	132	331	66	132
	P-VOP	25	100	75	50	125	25	50
비트율(Kbps)		222	2826	2336	1075	1908	418	1629
길이(초)		4	12	12	4	10	4	4
프레임율(fps)		25	25	25	25	29.970	25	25
넓이 x 높이		176 x 144	352 x 288	352 x 288	176 x 144	352 x 288	176 x 144	176 x 144

5.1 드로핑에 대한 실험결과

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

(드로핑-1)은 드로핑 시킨 파일의 크기가 크게 작아짐을 볼 수 있다. <표 3>은 실험 후 생성되는 영상의 크기 변화를 보여준다. 기존의 크기보다 약 60%의 크기의 영상 크기를 얻을 수 있다.

B 프레임을 드로핑 시키고 미디어 스트림 atom의 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 4>는 atom 수정에 걸리는 시간을 나타낸다.

(드로핑-2)는 atom정보를 수정하지 않고 이미 전송된 I와 P 프레임을 B 프레임 대신 재생 시키는 방법이므로 I와 P 프레임을 다시 재생 시키는 시간만이 소요된다. <표 5>는 I와 P프레임을 다시 재생 시키는데 걸리는 시간을 나타내고 있다.

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

〈표 3〉 드로핑 이후 파일 크기의 변화

파일명	파일크기		
	프레임 드로핑 이전의 영상	프레임 드로핑 이후의 영상	비율
PrettyWoman.mp4	172MB	100MB	58 %
Gone with the wind.mp4	240 GB	121 MB	50 %
Akiyo.mp4	111 KB	86 KB	77 %
Foreman.mp4	4145 KB	2485 KB	59 %
Hall_monitor.mp4	3426 KB	1791 KB	52 %
Coastguard.mp4	528 KB	318 KB	60 %
Container.mp4	2336 KB	1280 KB	54 %
News.mp4	207 KB	124 KB	59 %
Stefan.mp4	798 KB	478 KB	59 %

〈표 4〉 atom 수정에 걸리는 평균 및 전체 시간
(시간 단위 : 밀리 초, millisecond)

Data File	ATOM	샘플수	샘플 당 평균 수정시간
PrettyWoman.mp4	Stco	89916	0.012
	Stsz		0.010
	Stts		0.010
Gone with the wind.mp4	Stco	124738	0.014
	Stsz		0.011
	Stts		0.012
Akiyo.mp4	Stco	100	0.009
	Stsz		0.011
	Stts		0.009
Foreman.mp4	Stco	400	0.010
	Stsz		0.009
	Stts		0.007
Hall_monitor.mp4	Stco	300	0.009
	Stsz		0.010
	Stts		0.009
Coastguard.mp4	Stco	200	0.009
	Stsz		0.011
	Stts		0.010
Container.mp4	Stco	500	0.006
	Stsz		0.009
	Stts		0.007
News.mp4	Stco	100	0.009
	Stsz		0.004
	Stts		0.006
Stefan.mp4	Stco	200	0.007
	Stsz		0.008
	Stts		0.006

〈표 5〉 실시간 프레임 드로핑 (드로핑-2) 시간 측정 결과
(시간 단위 : 밀리 초, millisecond)

Data File	VOP	샘플수	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
Gone with the wind.mp4	I-VOP	14432	0.011	158.72
	P-VOP	10516	0.010	105.16
Akiyo.mp4	I-VOP	9	0.005	0.045
	P-VOP	25	0.003	0.075
Foreman.mp4	I-VOP	35	0.004	0.140
	P-VOP	100	0.004	0.400
Hall_monitor.mp4	I-VOP	26	0.004	0.104
	P-VOP	75	0.005	0.375
Coastguard.mp4	I-VOP	18	0.007	0.126
	P-VOP	50	0.006	0.300
Container.mp4	I-VOP	44	0.003	0.132
	P-VOP	125	0.004	0.500
News.mp4	I-VOP	9	0.005	0.045
	P-VOP	25	0.007	0.175
Stefan.mp4	I-VOP	18	0.006	0.108
	P-VOP	50	0.003	0.150

5.2 암호화에 대한 실험결과

본 연구에서는 MPEG-4 데이터를 위한 3가지 암호화 방법을 제안 하였는데 그 중 방법이 가장 뛰어난 암호화는 I-VOP 내의 매크로 블록과 P-VOP 내의 모션벡터를 모두 암호화 한 것이었다. 그러나 암호화 과정에서 처리해야 하는 데이터의 양이 크다는 단점을 가지고 있다. 이러한 단점을 극복하고 속도를 향상시키기 위해서 I-VOP의 빈도수를 조정하여 암호화에 사용되는 데이터의 양을 조절할 수도 있다. 하지만, 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 I-VOP 내의 매크로블록 암호화와 P-VOP내의 모션벡터를 함께 암호화 한 것이 I-VOP 암호화 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 6>는 암호화의 각 방법에 따라 드로핑 된 데이터 파일을 암호화 시킨 결과이다.

〈표 6〉 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	암호화-1	10403	0.0012	12.1149
	암호화-2	79513	0.0002	19.1956
	암호화-3	89916	0.0004	39.1413
Gone with the wind.mp4	암호화-1	14432	0.0013	18.7616
	암호화-2	110306	0.0005	5.2580
	암호화-3	124738	0.0004	9.9792
Akiyo.mp4	암호화-1	9	0.0008	0.0072
	암호화-2	25	0.0001	0.0025
	암호화-3	34	0.0002	0.0068
Foreman.mp4	암호화-1	35	0.0009	0.0315
	암호화-2	100	0.0001	0.0100
	암호화-3	135	0.0003	0.0405
Hall_monitor.mp4	암호화-1	26	0.0007	0.0182
	암호화-2	75	0.0002	0.0150
	암호화-3	101	0.0004	0.0404
Coastguard.mp4	암호화-1	18	0.0006	0.0108
	암호화-2	50	0.0001	0.0050
	암호화-3	68	0.0002	0.0136
Container.mp4	암호화-1	40	0.0005	0.0200
	암호화-2	125	0.0003	0.0375
	암호화-3	165	0.0004	0.0660
News.mp4	암호화-1	9	0.0009	0.0081
	암호화-2	25	0.0003	0.0075
	암호화-3	34	0.0004	0.0136
Stefan.mp4	암호화-1	18	0.0007	0.0126
	암호화-2	50	0.0002	0.0100
	암호화-3	68	0.0005	0.0134

6. 결 론

본 연구에서는 2가지의 드로핑 방법을 구현하였다. (드로핑-1)은 MPEG-4 파일 포맷에서 B 프레임을 드로핑하고 B 프레임이 드로핑 된 자리에 I, P 프레임을 이동시킨다. 그 후 Atom 정보들 중 Mdat와 Stco, Stsz, Stts부분을 수정하여 드로핑 시킨 후 전송 하는 방법이고, (드로핑-2)는 MPEG-4 파일을 서버에서 클라이언트로 전송 할 때 B 프레임을 실시간으로 드로핑 시켜서 전송하는 방법이다. 2가지 방법 모두 서버의 과부하를 줄여주어 전송속도를 빠르게 하고, 고화질의 영상을 재생시켜 준다는 장점이 있다. 그리고 (드로핑-1)의 방법은 실험결과 기존의 파일 크기의 약 60%정도의 크기를 서버에 저장함으로써 서버를 최적화 시킬 수 있지만, 헤더정보의 변경이 필요하여 원본 파일을 손상 시킬 수 있는 단점이 있다. (드로핑-2)의 방법은 헤더와 데이터를 변경시키지 않아도 된다는 장점이 있다. (드로핑-1)과 (드로핑-2)의 방법을 비교하였을 때 헤더정보가 변경되어야 한다는 차이점이 있을 뿐 전송하였을 때 성능의 차이점은 별로 없었다.

암호화에서는 DES를 사용해서 VOP에서 매크로 블록과 모션벡터를 추출하는 암호화 방법 3가지를 사용하였다. 암호화에서는 3가지 방법 중 I-VOP와 P-VOP의 내의 매크로 블록과 모션벡터를 동시에 암호화 하는 것이 성능은 좋았으나 데이터의 양이 많다는 단점이 있었다. 본 논문에서 적용한 암호화 방법은 영상을 먼저 디코딩 한 다음 특정한 암호화 기법을 적용시키거나 드로핑 시켜서 다시 MPEG방식으로 인코딩 하거나 두 작업을 동시에 진행하는 전통적인 비디오 영상 암호화의 일반적인 기법들에 비해 부가적인 오버헤드가 들어가지 않는다는 장점을 가지고 있다. 이런 특징은 멀티미디어 스트리밍 서비스에서 매우 큰 강점을 가지게 된다.

앞으로는 효과적인 스트리밍 서비스를 위해 서버의 과부하를 줄여줄 수 있는 드로핑 방법이 중점 연구 될 것이고, 이 때 전송할 미디어 데이터의 암호화를 위해 DRM에 관해서 더욱 많은 연구가 진행되어야 할 것이다.

참 고 문 헌

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

[2] C. C. Bisdikian and B. V. Patel, “Issues on Movie Allocation in Distributed Video-on-Demand Systems,” Proc. IEEE International Conference on Communications, IEEE Communications Society, New York, pp.250-255, 1995

[3] L. Xu and J. Helzer, “Media Streaming via TFRC: An Analytical Study of the Impact of TFRC on User-Perceived Media Quality,” IEEE INFOCOM, 2006.

[4] S. Floyd, M. Handley, J. Padhye and J. Widmer, “Equation-

Based Congestion Control for UnicastApplications," ACM SIGCOMM, 2000.

[5] "Proposed new text of IPMP FAQ," ISO/IEC JTC/SC29/WG11 M8141 Jeju 2002.

[6] "MPEG-4 Intellectual Property Management & Protection (IPMP) Overview & Application Document," ISO/IEC/SC29/WG11/N2614 MPEG 98, December, 1998.

[7] Ming-Ting Sun and Amy R. Reibman, Compressed Video over Networks, Marcel Dekker, Inc., 2001.

[8] King N. Ngan, Chi W. Yap and Keng T. Tan, Video Coding for Wireless Communication Systems, Marcel Dekker, Inc., 2001.

[9] 정홍섭, 박규석 "네트워크 부하 기반 프레임 생략 전송 알고리즘", 멀티미디어학회 논문지, 제6권 제7호, December, 2003.

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

[11] W. Zeng and 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. pp.129-140, 1996.

[12] J. Sandvoss, J. Winkler and H. Witting, "Network Layer Scaling : Congestion Control in Multimedia Communication with Heterogeneous Networks and Recivers," IBM European Networking Center, Heidelberg, 1994.

[13] Jae-Gon Kim, Yong Wang, Shih-Fu Chang, Kyeongok Kang and Jinwoong Kim "Description of utility function based optimum transcoding," ISO/IEC JTC1/SC29/WG11 M8319, Fairfax, May, 2002.

[14] W. Li, J. Ohm, M. V. Schaar, H. Jiang and S. Li, MPEG-4 Video verification model ver. 18.0, ISO/ IEC/ JTC1/ SC29/ WG11/ N3908, 2001.

[15] H. Radha, Y. Chen, K. Parthasarathy and R. Cohen, "Scalable internet video using MPEG-4," Signal Processing: Image Communication, Vol.15, pp.95-126, 1999.

[16] 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, pp.137-144, Feb., 1996.

[17] 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.

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

[19] 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, pp.21-29, July, 1997.

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

[21] 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.

[22] 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.

[23] 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.

[24] "QuickTime File Format," Apple Computer, June, 2000.

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

[26] 김건희, 신동규, 신동일 "MPEG-4 비디오 스트림의 디지털 저작권 관리를 위한 암호화 기법의 연구", 정보처리학회논문지, April, 2005.

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

신 동 규

e-mail : shindk@sejong.ac.kr

1986년 2월 서울대학교 계산통계학과(이학사)

1992년 8월 Illinois Institute of Technology 전산학과(공학석사)

1997년 8월 Texas A&M University 전산학과(공학박사)

1986년 2월~1991년 1월 한국국방연구원 연구원

1997년 8월~1998년 2월 현대전자 멀티미디어연구소 책임연구원

1998년 3월~현 재 세종대학교 컴퓨터공학과 교수

관심분야 : 상황인식 미들웨어, 웹기반 멀티미디어, 멀티미디어 DRM

신 동 일

e-mail : dshin@sejong.ac.kr

1988년 연세대학교 전산과학과(이학사)

1993년 M.S. in Computer Science, Washington State University

1997년 Ph.D in Computer Science, University of North Texas

1997년 9월~1998년 2월 시스템공학연구소 선임연구원

1998년 3월~현 재 세종대학교 컴퓨터공학과 부교수

관심분야 : 상황인식 미들웨어, 무선인터넷, 게임, 지능형 에이전트, HCI

박 세 영

e-mail : sypark@gce.sejong.ac.kr

2003년 3월~현 재 세종대학교 컴퓨터공학과 재학중

관심분야 : 무선 네트워크, 유비쿼터스 컴퓨팅, 멀티미디어DRM