공 학 석 사 학 위 논 문

H.264 SE 코덱 기반의 비디오 워터인크립션 기법

2009년 2월

부 경 대 학 교 대 학 원

컴 퓨 터 공 학 과

김 원 제

공 학 석 사 학 위 논 문

H.264 SE 코덱 기반의 비디오 워터인크립션 기법

지도교수 권 기 룡

이 논문을 공학석사 학위논문으로 제출함.

2009년 2월

부 경 대 학 교 대 학 원

컴 퓨 터 공 학 과

김 원 제

김원제의 공학석사 학위논문을 인준함.

2009년 2월 25일

주 심 공학박사 문 광 석 인

위 원 공학박사 이 석 환 인

위 원 공학박사 권 기 룡 인

목 차

그 림 목 차

Video Water-Encryption Technique Based on H.264 SE Codec

Won-Jei Kim

Department of Computer Engineering, The Graduate School,
Pukyong National Unaiversity

Abstract

Multimedia production and distribution, now a days, is all digital. The advantages of digital distribution, like noise-free transmission, are well known. However despite being an economic opportunity, this also is a major concern for content owners. The unlimited copying of digital data without loss of fidelity is undesirable because it causes considerable financial losses. An effective digital rights management(DRM) system would allow the content providers to track, monitor and enforce usage rights of their contents in both digital and analog form.

Watermarking and encryption are the two often mentioned techniques proposed for protection of intellectual property rights. Although encryption plays an important role in DRM and video streaming however it can only protect data during transmission from content provider to authorized user. Thus encryption does not provide any protection once the content has been decrypted. A watermark however persists within the decrypted video stream and can be used to access control rights. A DRM-complaint device can read the embedded watermark and control and prevent video duplication or playback. Video watermarking may also be used to track and trace video content and broadcast monitoring.

In this paper, a video water-encryption scheme based on H.264 scalable extension codec is presented. In order to compress video sequence and embed watermark and encryption at the same time, The video is watermarked and encrypted during H.264 scalable extension compression process. Embedding amount of watermark and embedding positions are calculated by a frame of each layer. And then watermark is embedded in a 4×4 DCT block of video data. The watermarked video data are encrypted.

The scheme keeps secure against present attacks, is efficient in implementation, keeps imperceptible, and is robust against other attacks, such as encoding, common signal processing, and geometrical processing. These properties make the scheme a choice for secure video transmission or distribution.

Ⅰ. 서 론

1. 연구 배경 및 필요성

세계 상용 서비스 후 가입자가 증가하고 있는 위성 및 지상파 DMB (Digital Multimedia Broadcasting)와 시장 진입을 서두르고 있는 IPTV (Internet Protocol Television)는 통신과 방송이 융합되는 디지털 컨버전스 (digital convergence)의 대표적인 서비스로 자리 매김하고 있다[1]. 통방융합은 방송의 디지털화와 통신의 광대역화와 함께 통신 및 미디어 처리 기술의 발전에 따라 크게 망, 단말, 서비스 및 사업자 범주에서의 컨버전스로 진화하고 있다.

통신사업자 측은 IPTV 및 ICoD(Internet Contents on Demand) 등의 형태로 기존의 인터넷 및 음성 서비스와 함께 TV를 통한 거실로의 진출을 추진하고 있으며, 방송사업자 측은 양방향 디지털 방송, DMC(Digital Media Center) 및 NGNA(Next Generation Network Architecture) 기반의 디지털 케이블 방송 등의 형태로 방송 서비스를 탈피하여 인터넷 및 VoIP (Voice over Internet Protocol)를 통한 통신 시장으로의 진출을 적극 추진하고 있다. 특히, 이동통신사업자들 역시 이동통신 시장을 벗어나 이동방송 (DMB, DVB-H: Digital Video Broadcasting-Handheld , MediaFLO: Forward Link Only) 및 휴대인터넷(WiBro: Wireless Broadband), HSDPA(High Speed Downlink Packet Access) 등 방송 콘텐츠 서비스로의 확장을 추진하고 있다.

통방융합의 세계적인 흐름과 발맞추어 유선데이터 및 방송, 무선데이터, 이동통신 등 인프라의 종류를 막론하고 모든 사업자들은 방송/통신/데이터

및 음성 서비스의 통합을 추진하고 있으며, 이런 배경 속에서 방송 서비스 역시 전통적인 방송 매체를 통한 거실 TV 대상의 서비스로부터 탈피하여 각종 이종 망과의 연동을 통한 다양한 종류의 단말을 대상으로 언제 어디서나 방송 콘텐츠의 접근 및 소비를 가능하게 하는 유비쿼터스(Ubiquitous) 방송 서비스로 진화하고 있다.

이러한 서비스를 겨냥해서 기존의 코덱을 개선한 H.264 SE(scalable extension)를 비롯한 여러 가지 스케일러블 비디오 코덱(codec)들이 개발되었다[2-8]. 본 논문에서 대상으로 하는 H.264 SE는 ISO/IEC 산하 MPEG (Moving Picture Experts Group) 과 ITU-T 산하 VCEG (Video Coding Experts Group)가 Joint Video Team(JVT)을 이루어 MPEG-4 SVC 또는 H.264 SE 라는 이름으로 표준화가 완료된 코덱이다. 이 코덱을 통해 부호화된 비디오 자료에서 해상도와 프레임율, 그리고 화질별로 다양한 영상을 추출하여 각 단말들의 특성에 맞게 제공하게 된다[9-16].

하지만 과도한 서비스 경쟁으로 미처 해결 방안을 내놓기 이전에 방송 서비스를 시행함으로서 무분별한 지적 재산권 및 저작권 침해와 복제, 무단 배포 등의 불법적인 사례들로 골치를 앓고 있다. 이에 지적 재산권과 저작권 침해를 방지하고 복제와 무단 배포를 근절해야 할 필요성이 대두되고 있다.

디지털 콘텐츠는 일반적인 오프라인 콘텐츠와는 달리 쉽게 복사 및 배포가 가능하다는 특성이 있다. 따라서 합법적인 구매자가 판매자로부터 디지털 콘텐츠를 구입한 후, 이것의 불법적인 재분배(Redistribution)를 막을 수 있는 방법이 고려되어야 한다. 이러한 방법들로 최근 디지털 콘텐츠의 지적 재산권 보호를 위한 디지털 워터마킹 기술 및 핑거프린팅의 연구가 활발히 진행되고 있다. 이러한 원천 기술들을 이용하여 많은 DRM(Digital Rights Management)모델들이 제시되어 왔으며 현재 널리 활용되고 있다.

또한 디지털 콘텐츠를 안전하게 보호하기 위한 응용기술로는 디지털 콘텐츠 유통/서비스를 위한 저작권 보호기술, 디지털 창작물에 대한 저작권/소유권/사용권을 제어하는 기술 및 암호기술 그리고 디지털 워터마킹 기술 등이 있다.

본 논문에서 사용한 알고리즘으로 워터마킹(watermarking)과 암호화가 있다. 첫째, 워터마킹 기술은 이미지, 오디오, 비디오, 텍스트 등의 콘텐츠 내에 부가정보를 비인지적(혹은 인지적)으로 삽입하고 추출하는 기술이다. 부가정보는 저작권자와 소유권자에 대한 정보, 구매자 정보, 기타 제어 정보 등이 삽입 될 수 있다. 삽입되는 정보의 종류에 따라 여러 가지 용도로 사용 가능한데, 소유권자의 정보가 삽입되어 있을 경우 콘텐츠의 소유권 정보를 추출하여 불법 복사본을 판별할 수 있고, 삽입 정보가 구매자 정보인 경우에는 처음 콘텐츠를 유출한 구매자를 역추적 할 수 있게 해 준다. 또한 삽입되어 있는 정보에 따라 콘텐츠의 사용제한을 가하거나 위, 변조 여부를 탐지하는 등의 다른 부가적인 서비스도 가능하게 해준다.

그 다음으로 암호화 기술은 허가 받지 않은 사람들은 쉽게 이해할 수 없도록 데이터를 암호문이라고 불리는 형태로 변환하는 것이다. 암호해독은 암호화된 데이터를 원래의 형태로 되돌림으로서, 누구나 이해할 수 있게 만드는 과정이다.

암호화/해독의 사용은 통신 기술의 역사만큼이나 오래 되었다. 전시에는 전송내용을 적이 훔치는 것을 막기 위하여 암호 코드가 사용되었다. 단순한 암호 코드에는 숫자를 위해 문자들로 치환하는 것, 알파벳 내에서 문자를 교체하는 것, 그리고 측파대 주파수를 전도시킴으로써 목소리 신호의 파장을 바꾸는 것 등이 포함된다. 복잡한 암호 코드는 디지털 신호 내의 데이터 비트들을 재배열하는 매우 복잡한 컴퓨터 알고리즘에 따라 조작된다.

2. 기존 연구

이전에 발표되었던 비디오 워터마킹 알고리즘에 대해서 살펴보면 크게 3가지로 나눌 수가 있다. 원본 비디오와 비디오를 부호화하는 과정, 그리고 부호화된 비디오 자료에 워터마킹을 수행하는 방법이다.

첫째, 비디오를 부호화하기 이전에 원본 비디오에 워터마크를 삽입하게 되면 재 부호화나 일반적인 신호처리에 강인하지만, 부호화 이전에 원본 비디오를 연산하는 과정을 거쳐하는 높은 계산량이 필요한 단점이 있다.

둘째, 비디오를 부호화하는 과정이나 부호화된 비디오 자료에 워터마킹을 수행하게 되면 가장 큰 장점으로 시간 효과를 노릴 수 있다. 하지만 재 부호화와 일반적인 신호처리에 약하다는 단점이 있다.

대표적인 워터마킹 알고리즘 몇 가지를 살펴보면, 1998년 Hartung등은 MPEG-2 비트열에 워터마크를 삽입하고, 대역확산 방식을 적용하였다[17]. 그러나 이 방법은 비트열상의 워터마크 삽입방법이 강인하지 않고, 다양한 부호화기에 적용할 수 없는 단점이 있다.

1998년 Hsu등은 워터마크를 삽입하기 위해 인접한 블록의 DCT 계수 사이의 극성을 이용하여 워터마크를 삽입한다[18]. 이 방법은 DCT 계수 값이 극성을 유지하기 위해 변화되므로 워터마크를 삽입할수록 DCT 계수의 오차가 누적되어 화질의 열화가 발생한다.

1999년 Zhu등은 정지영상 워터마킹 방식을 그대로 비디오 워터마킹 방법에 적용하였다[19]. 이 방법은 정지영상 워터마킹 방식을 그대로 비디오 워터마킹 방법에 적용함으로써, 비디오 프레임 간의 상관성 및 움직임 변화를 고려하지 않았다. 그 외에 실시간 전송을 목적으로 하는 워터마킹 방법은 복잡성을 최소화해야 하기 때문에 상대적으로 강인성을 보장하지 않

는다. 또한 다양한 비트율이나 부호화 방식 등에 적용될 수 없다.

2007년 Zhang등은 H.264/AVC 내에 워터마크를 삽입을 수행한다[20]. 워터마크 전처리를 통해서 삽입할 워터마크 비트를 구하고 압축을 수행하는 과정 중에 워터마크를 삽입한다. 하지만 이 방법은 일반적인 신호처리에는 강인하지만, 기하학적인 공격이나 재 부호화 공격시 강인성을 잃게 되는 단점을 가지고 있다.

대표적인 암호화 알고리즘에 대해서 살펴보면, 1977년 미국 상무성의 국립표준국에서 채택한 미국 표준 암호 알고리즘 DES(Data Encryption Standard)가 있다[21]. DES는 개인키를 사용하여 데이터를 암호화하는 방법으로서 널리 사용되며, 72천조개 이상의 암호 키가 사용되는 것이 가능하다. 주어진 각 메시지를 위한 키는, 이렇게 막대한 량의 키 중에서 무작위로 선택된다. 다른 개인키 암호화 방법과 마찬가지로, 송신자와 수신자 둘 모두는 동일한 개인키를 알고, 사용해야만 한다. DES는 컴퓨터 성능의 발달에 따라 보안성이 약화되어 2중, 3중 DES를 사용하였으며, 매 5년마다 안전성을 검증하다가 97년에 NIST(National Institute of Standards and Technology)는 AES(Advanced Encryption System)를 제시했고 2000년에 Rijndeal을 AES로 선택했다[22-24].

Rijndael은 블록 암호알고리즘 방식이며 128비트 블럭 단위로 암호화를 하고, 사용되는 키의 사이즈는 128비트, 192비트, 256비트 등이 있으며, 라운드 수는 각각 10, 12, 14라운드를 사용하고 각 라운드마다 SubBytes, ShiftRows, MixColumns, AddRoudKey의 4단계를 거치게 된다. 또한 기존의 DES가 피에스탈 구조(Feistel network)이었던 것과는 달리 SPN (substitution permutation network) 구조를 가지고 있어서 하드웨어나 소프웨어적으로 구현했을 때 구현이 쉽고 메모리를 적게 소모하는 것 등 모두 좋은 성능을 보이는 특성을 가지고 있다.

RSA(Ron Rivest, Adi Shamir, Leonard Adleman)는 1977년에 Ron Rivest, Adi Shamir와 Leonard Adleman에 의해 개발된 알고리즘을 사용하는 인터넷 암호화 및 인증 시스템이다[25]. RSA 알고리즘은 가장 보편적으로 사용되는 암호화 및 인증 알고리즘으로서, 넷스케이프와 마이크로소프트 웹브라우저 기능의 일부로 포함된다. 이 알고리즘은 두 개의 큰 소수들의 곱과 추가 연산을 통해 하나는 공개키를 구성하고, 또 하나는 개인키를 구성하는데 사용되는 두 세트의 수 체계를 유도하는 작업이 수반된다. 하지만 복잡한 알고리즘에 의해 연산량이 많아져서 오랜 수행시간이 필요하다는 단점이 있다.

3. 연구 목적 및 개요

현재 사용자 제작 콘텐츠(UCC: User Created Contents)와 같은 동영상 위주의 정보제공 콘텐츠가 급속히 증가하고 있으며, 공간적, 시간적, 그리고 화질적 변화같은 스케일러블 특성을 고려한 비디오 압축코덱 H.264 SE의 표준화가 완료되었다. 또한 각종 이종 망과의 연동을 통한 다양한 종류의 단말이 증가하였다. 이에 H.264 SE 코덱의 범용적인 사용에 대비한 SE 코덱 기반의 보다 효과적인 DRM 연구가 필요하기 때문에 본 논문에 관련된 연구를 시작하게 되었다. 그리하여 본 논문에서는 기존의 연구들에서 고려되지 않은 H.264 SE의 특성을 고려하면서 효율적인 암호화를 거침으로서 보다 높은 안정성과 보안성을 가진 강인한 비디오 워터마킹 및 암호화 알고리즘을 제안한다.

이 알고리즘은 H.264 SE에서 지원하는 여러 가지 해상도와 프레임율, 그

리고 화질의 변화에도 워터마크를 추출할 수 있으며, 최초 사용자에게는 암호화된 비디오를 제공하여 일차적인 보안을 유지하며, 이후 인증된 사용자를 통해서 비디오가 유포될 가능성을 고려하여 워터마크를 삽입함으로서 이차적인 보안까지 고려하였다. 또한 비디오 부호화 과정 중에 워터마킹을 수행함으로서 실시간 부호화를 적용할 수 있게 하였다.

2장에서 관련 연구로 H.264 SE의 특성과 AES(Advanced Encryption Standard)에 대해서 기술하고, 3장에서 제안한 비디오 워터인크립션 알고리즘에 대해 설명하고 4장에서 실험을 수행하는 방법에 대해서 설명하며, 실험을 통해서 나온 결과와 그 결과에 대해서 고찰해 본다. 마지막으로 본 논문의 결론을 맺으며, 향후 추가되어야 할 개선점이나 연구 방향 등을 제시한다.

Ⅱ. 관련 연구

1. H.264 SE

네트워크를 통한 멀티미디어 서비스에서 스케일러블 비디오 부호화의 목적은 특정 비트율로 화질을 우화하는 것이다. 이 때 그 비트열은 임의 비트율에서도 복호화가 되어야 한다. 단일 계층 압축 부호화기는 하나의 비트율/프레임율/영상크기만을 지원하는 하나의 비트열을 생성하는데 반해, 스케일러블 비디오 부호화기는 다양한 비트율/프레임율/영상크기에 대한 스케일러빌러티(scalability)를 지원한다. 스케일러블 비디오 코딩은 여러 개의 비디오 계층을 하나의 비트열로 부호화하며, 각 층은 각각의 비트율, 프레임율, 영상 크기 및 화질을 가지고 있다. 즉 비디오를 공간적, 시간적, 그리고 화질적 차원의 임의 값을 가지는 비트열로 부호화화며, 그 세 가지 차원의 조합에 따라 폭넓은 스케일러빌러티를 제공할 수 있다.

기술적인 관점에서, 하나의 스케일러블 비트열은 두 개 혹은 그 이상의 의존적인 계층으로 구성될 수 있다. 이 경우, 스케일러블 코덱은 하나의 기본 계층과 스케일러블 상위 계층들로 구성된다. 여기서 기본 계층 및 연속되는 상위 계층의 정보가 함께 이용되어 보다 개선된 비디오 비트열을 만든다. 일반적으로 기본 계층은 기본적인 비디오 화질을 제공하고, 연속된 상위 계층은 이전 계층들로 만들어진 비디오보다 높은 화질을 갖도록 부호화한다. 마찬가지로 시간 및 공간 해상도에서도 동일한 원리를 적용하여 스케일러빌러티를 지원한다. 이러한 스케일러빌러티를 제공하는 H.264 SE에서 2계층의 예시에 해당하는 블록도를 그림 1에서 나타내었다.

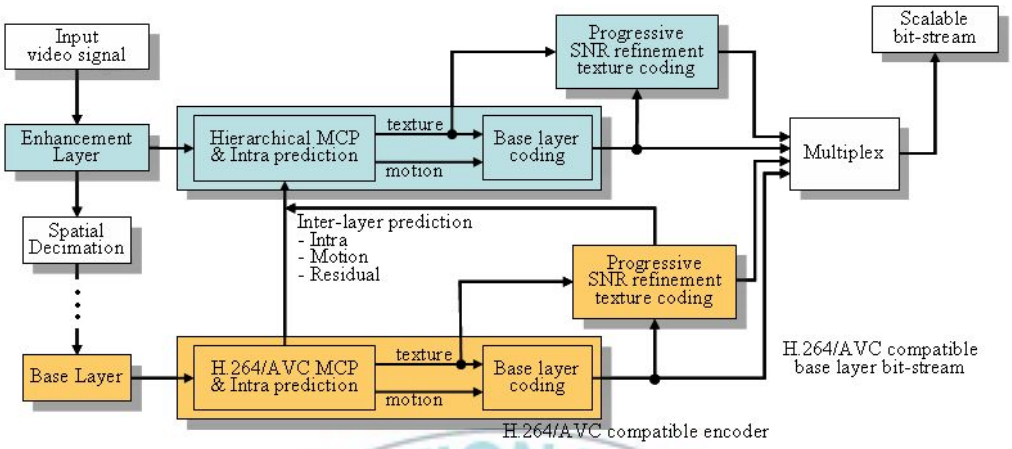


그림 1. 공간적 2계층의 스케일러빌러티를 제공하는 H.264 SE 부호화기

가. 공간 스케일러빌러티

공간 스케일러빌러티는 각 영상 크기의 계층을 쌓는 피라미드 개념을 지원한다. 이 때 각 해상도의 비디오는 하위 계층의 부호 결과물인 움직임, 텍스쳐 및 잔여 신호 정보를 이용함으로서 새로운 계층으로 부호화된다.

그림 2와 그림 3은 공간 스케일러빌러티를 위한 부호화기 및 복호화기를 블록도로 나타낸 것이다. 공간 스케일러빌러티의 부호화 과정에서는 입력 비디오에 대해 원하는 여러 가지 해상도로 변환 후 기본 계층(base layer)에 대해 기본적인 H.264/AVC 부호화를 수행하고 기본 계층을 제외한 나머지 상위 계층(enhancement layer)과의 차이 값을 부호화하여 압축된 비트열이 출력된다.

복호화 과정에서는 부호화 과정의 역으로서 기본 계층에 대해서 H.264/AVC 복호화를 수행하고 이에 복호된 상위 계층의 합으로서 최종 상위 계층에 해당하는 비디오가 출력으로 나오게 된다.

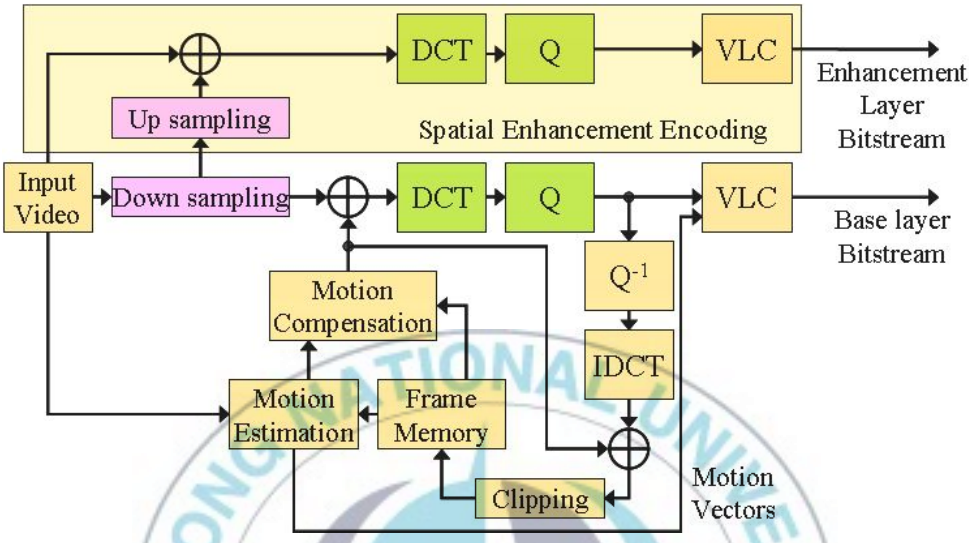


그림 2. 공간 스케일러빌러티를 위한 부호화기

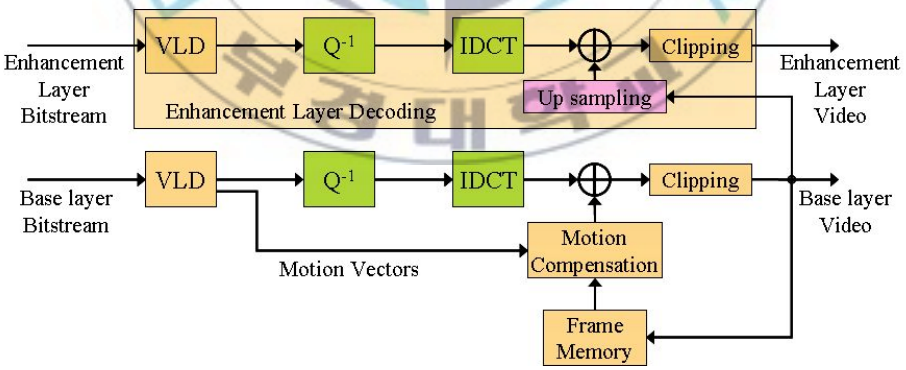


그림 3. 공간 스케일러빌러티를 위한 복호화기

나. 시간 스케일러빌러티

비디오의 첫 영상은 IDR(Instantaneous Decoder Refresh)영상으로서 화면내(intra) 부호화된다. 이전에 부호화된 모든 영상들이 현재 부호화되는 영상보다 화면 표시 순서 상 앞서 위치할 때, 이것을 키픽처(key picture)라고 부른다. 시간적으로 현재 키픽처와 이전 키픽처 사이에 위치하고 있는 모든 non-key picture들을 하나의 GOP(Group of Pictures)라고 한다. 키픽처는 임의 접근을 위해 화면내 부호화되거나 이전 키픽처를 참조 영상으로 움직임 보상 예측을 통해 P픽처로 화면간(inter) 부호화된다. GOP의 나머지 영상들은 계층적(hierarchical)으로 참조된다.

그림 4에서 보여지는 블록도와 같이 시간 스케일러빌러티는 시간축 상에서 다운샘플링(downsampling)과 업샘플링(upsampling)을 사용한다는 것을 제외하고 공간 스케일러빌러티와 유사하게 동작한다. 시간축상의 다운샘플링은 프레임 삭제, 업샘플링은 프레임 복사를 수행한다.

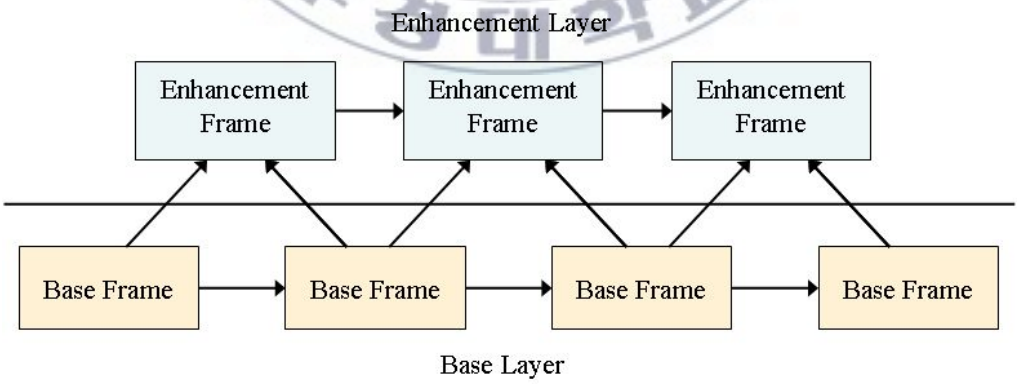


그림 4. 시간 스케일러빌러티 특성

다. FGS

FGS(Fine Granularity Scalability)는 계층 구조의 한계를 극복하고 각 비트율 변동에 적응적으로 복호 가능하도록 하는 SNR 스케일러블 부호화 기술이다. 이는 양자화에 의한 오차 신호를 이전 계층 보다 적은 양자화 계수 값으로 보정하여 부호화하는 개념이다. 즉, 첫 번째 계층은 큰 양자화 간격으로서 듬성듬성하게 표현한 후, 계층이 올라갈수록 양자화 간격의 폭을 세밀하게 조정하여 보다 좋은 화질의 계층을 쌓는 개념이다.

아래 그림 5와 6은 FGS 부호화기 및 복호화기의 블록도를 나타낸 것이다. 그림 5에 나타난 것처럼 FGS의 경우도 기본 계층과 상위 계층으로 나뉘며 기본 계층은 공간 스케일러빌러티와 유사하게 동작한다. 상위 계층은 원 영상과 기본 계층에서 복원된 영상 간의 차이 값을 전송하게 되는데 이러한 차이 값을 전송하는 방식이 계층 기반 스케일러블 방식과 다르다. 기본적으로 FGS는 DCT 계수 값을 비트평면(bit plane)단위로 전송하는 방식으로 엔트로피 부호화에서 차이가 발생한다. 부호화기에서 원영상과 기본 계층에서 복원된 영상 간의 차이 값은 다시 DCT 변환되어 각 화소가 DCT 계수 값으로 바뀌게 되고 이러한 DCT 계수 값을 스케일러블하게 전송하기 위하여 JPEG의 프로그레시브 전송처럼 비트평면으로 사상하여 최상위 비트(MSB: Most Significant)에서 최하위 비트(LSB: Least Significant) 순으로 각 평면 단위로 계수 값을 전송한다. 일반적으로 DCT 계수 값은 양자화 계수 값의 크기에 따라 통계적 특성이 크게 좌우되는데 비해 비트평면을 이용한 방법은 각 평면의 비트들이 양자화 계수 값에 비교적 독립적이기 때문에 부호화 효율 측면에서 유리하다.

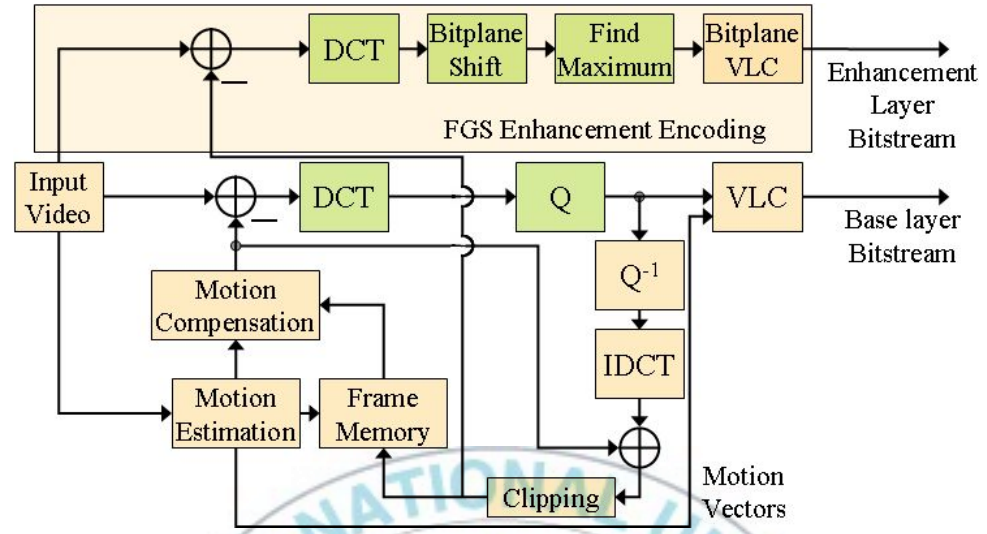


그림 5. FGS를 위한 부호화기

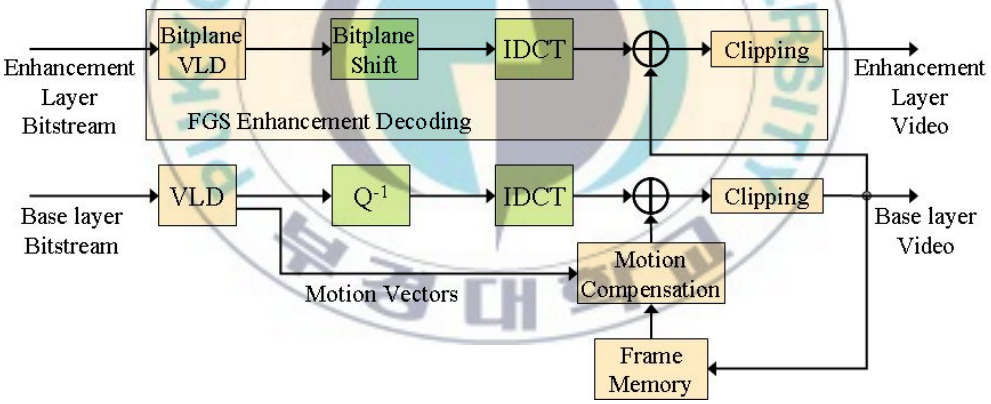


그림 6. FGS를 위한 복호화기

2. AES

2000년 10월 Rijndael알고리즘이 AES로 선정되었으며, DES(Data Encryption

Standard)의 페이스텔 구조와 달리 SPN(Substitution- Permutation Network)구조를 이용한다. 블록길이는 128비트이며, 키의 비트길이는 128, 192, 256비트 중 선택이 가능하다. 또한 큰 키를 쓰거나 라운드 수를 반복할수록 안정성이 커진다.

AES의 구조는 크게 SubBytes, ShiftRows, MixColumns, AddRoundkey의 네 단계로 나뉘며, 아래 그림 7은 AES의 동작 과정을 나타낸다.

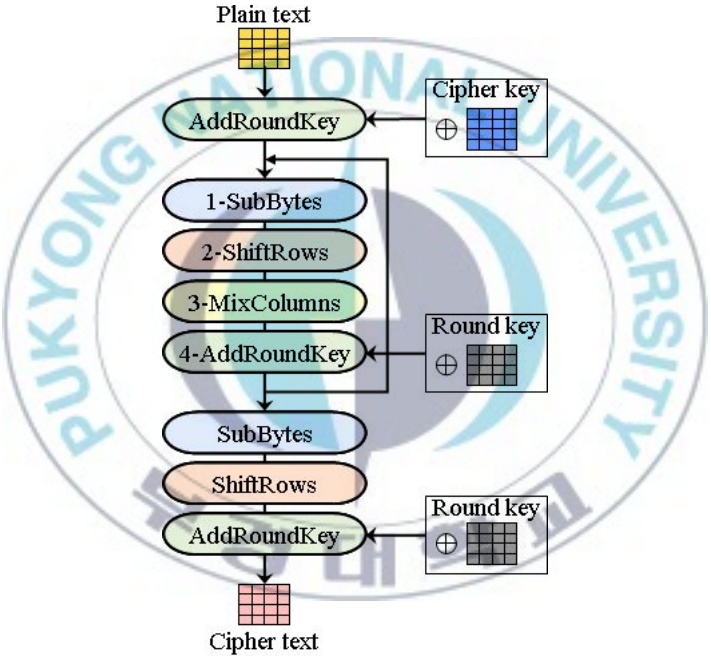


그림 7. AES의 동작 과정

가. SubBytes 단계

이 단계에서는 메시지를 정방행렬로 구성하였을 때, 그 배열에 있는 각각의 바이트들은 1바이트를 입력으로, 1바이트는 출력하는 S박스를 이용해

업데이트하며 S박스는 그림 8에 나타내었다.

하나당 1Byte라 하고, 한 바이트의 메시지를 a_n이라 하면 그림 9와 같이 16바이트(=128bit)가 나열되어 있을 것이다. 이 S박스는 단일치환암호(=일정 범위 안에서 치환되어지는 암호)와 같은 논리로 0~255의 범위 안에서 진행된다)

		y														
		0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
x	0	63	7c	77	7b	f2	6b	6f	c5	30	01	67	2b	fe	d7	ab	76
	1	ca	82	c9	7d	fa	59	47	f0	ad	d4	a2	af	9c	a4	72	c0
	2	b7	fd	93	26	36	3f	f7	cc	34	a5	e5	f1	71	d8	31	15
	3	04	c7	23	c3	18	96	05	9a	07	12	80	e2	eb	27	b2	75
	4	09	83	2c	1a	1b	6e	5a	a0	52	3b	d6	b3	29	e3	2f	84
	5	53	d1	00	ed	20	fc	b1	5b	6a	cb	be	39	4a	4c	58	cf
	6	d0	ef	aa	fb	43	4d	33	85	45	f9	02	7f	50	3c	9f	a8
	7	51	a3	40	8f	92	9d	38	f5	bc	b6	da	21	10	ff	f3	d2
	8	cd	0c	13	ec	5f	97	44	17	c4	a7	7e	3d	64	5d	19	73
	9	60	81	4f	dc	22	2a	90	88	46	ee	b8	14	de	5e	0b	db
	a	e0	32	3a	0a	49	06	24	5c	c2	d3	ac	62	91	95	e4	79
	b	e7	c8	37	6d	8d	d5	4e	a9	6c	56	f4	ea	65	7a	ae	08
	c	ba	78	25	2e	1c	a6	b4	c6	e8	dd	74	1f	4b	bd	8b	8a
	d	70	3e	b5	66	48	03	f6	0e	61	35	57	b9	86	c1	1d	9e
	e	e1	f8	98	11	69	d9	8e	94	9b	1e	87	e9	ce	55	28	df
	f	8c	a1	89	0d	bf	e6	42	68	41	99	2d	0f	b0	54	bb	16

그림 8. S박스의 구성

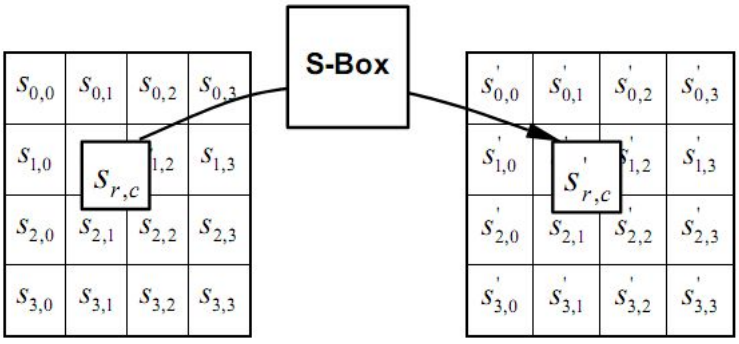


그림 9. SubBytes 수행 과정

나. ShiftRows 단계

이 단계에서 '유일한 비선형연산'이 이루어진다. 아래 그림 10에서 보듯이 각 행(Row)에 따라 행의 단위만큼 이동된다. 즉, Row-1만큼씩 이동됨을 볼 수 있을 것이다.

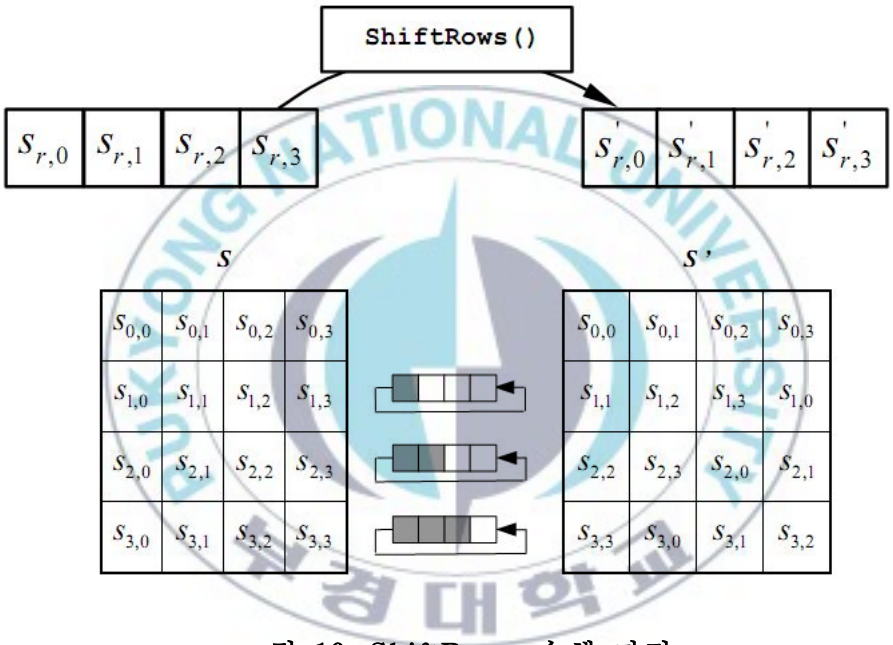


그림 10. ShiftRows 수행 과정

다. MixColumns 단계

그림 11에서 보여지는 이 단계는 각각의 COLUMN들이 고정된 다항식 $c(x) = 0.3x^3 + 0.1x^2 + 0.1x + 0.2$에 의해서 치환이 된다.

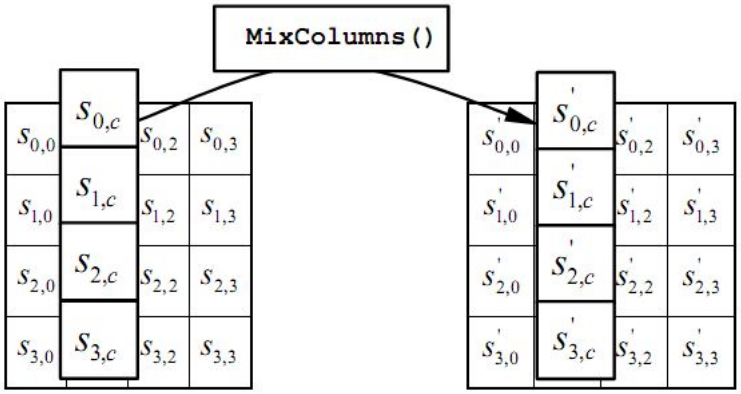


그림 11. MixColumns 수행 과정

라. AddRoundKey 단계

여기서는 MixColumns의 출력과 라운드 키의 XOR값을 구하는 단계이며, 아래 그림 12에서 보여준다.

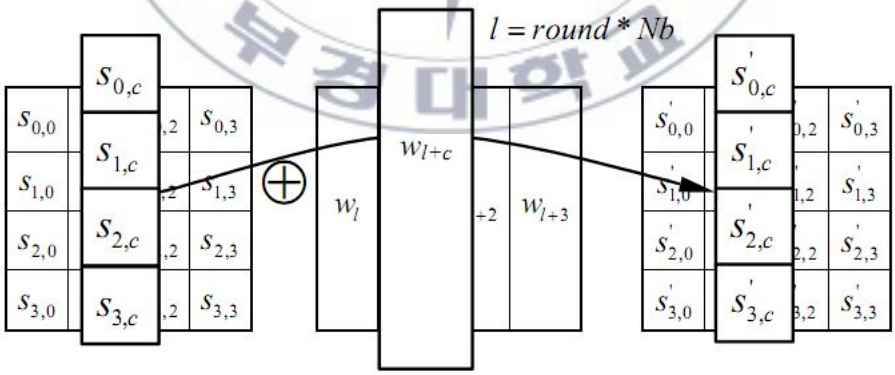


그림 12. AddRoundKey 수행 과정

이렇게 네 단계를 거쳐서 AES가 작동하게 된다. 이러한 라운드를 실제

Rijndael에서는 10~14회 반복한다.

3. 비교 논문 연구

Zhang등이 제안한 알고리즘은 워터마크를 생성하는 전처리 단계와 생성한 워터마크를 삽입하고 추출하는 워터마킹 단계로 나눌 수 있다.

우선 전처리 단계는 그림 13에서 보여지는 것과 같이 수행된다. 입력으로 사용되는 워터마크를 그 특성에 맞게 주파수 마스킹을 수행하여 입력하게 될 워터마크 값을 계산한다.

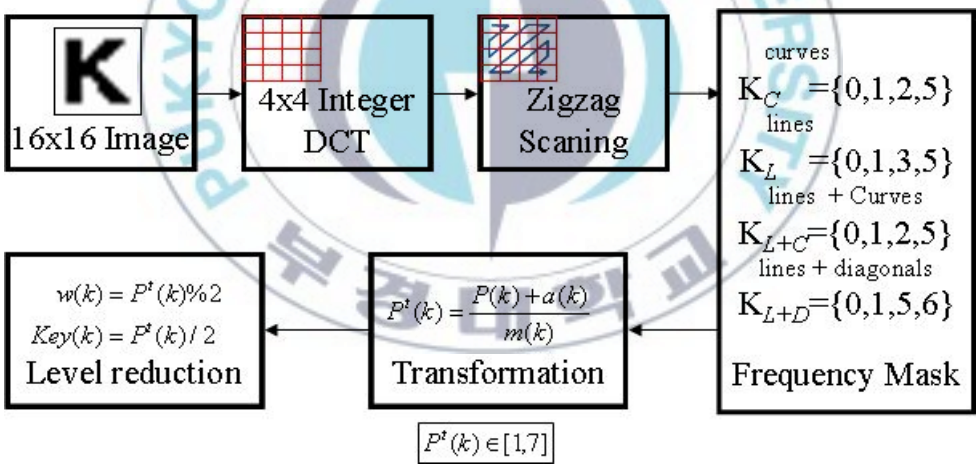


그림 13. Zhang의 알고리즘: 워터마크 전처리 단계

그림 14에서는 위에서 생성된 워터마크를 H.264/AVC 내에서 삽입하는 과정을 블록도로 나타낸 것이다.

워터마크 패턴이 가지는 크기가 $M \times M$일 때, 생성되는 워터마크 정보 $\mathbf{w}=\{w(n)|n=0,...,M^2/4-1\}$는 식 (1) 을 거쳐서 바이폴라 벡터로 사상한다.

$$w^b(n) = (-1)^{w(n)}, \qquad n = 0,1,...,M^2/4-1 \qquad (1)$$

4×4 DCT된 계수 값 중 중간주파수 위치의 계수 값을 워터마크비트 $w^b(m)$로 치환한다.

$$X_{u_o,v_o}(m) \leftarrow \widetilde{X}_{u_o,v_o}(m) = \alpha \bullet \beta \bullet w^b(m) \qquad (2)$$

$$\beta = \max[0, -X_{0,0}(m) + \mu \sum_{1 \le u,v \le 3} |X_{u,v}(m)|] \qquad (3)$$

u_o와 v_o는 계수의 위치를 가리키며, 이득계수 α는 실험적으로 결정되는 값이며, β는 DC계수 $X_{0,0}(m)$과 AC계수 $X_{u,v}(m)$에 의해 구해진다. μ는 가중치계수로서 워터마크의 가시성을 고려해서 결정되어지는 값이다.

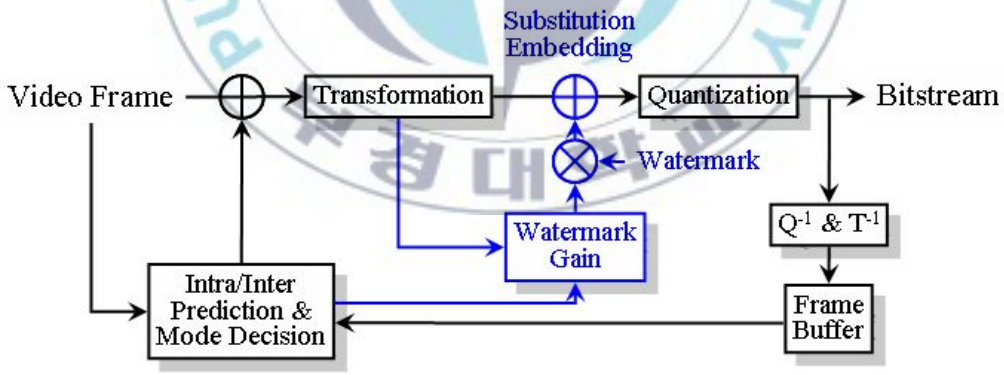


그림 14. Zhang의 알고리즘: H.264/AVC 내 워터마킹 시스템

워터마크를 삽입시에 식 (4)에 나오는 H.264/AVC 의 라그랑지안 최적화 알고리즘을 사용하여 선택되는 최고의 모드 $\tilde{o}$에 따라 삽입한다.

$$\tilde{o} = \arg\min\left(D(\tilde{B}, o) + \lambda R(\tilde{B}, o)\right) \qquad (4)$$

위에서 D와 R은 왜곡(distortion)과 현재 모드 o를 인코딩하기위해 소모되는 비트이며 λ는 모드선택을 위해 결정되어지는 라그랑지안 승수를 말한다.

Zhang의 알고리즘에서 주목할 부분은 주파수 마스킹과 변환 특성이며, 회전, 절단, 그리고 축소 등의 기하학적인 공격에 취약한 것으로 보여진다.

Ⅲ. 제안한 비디오 워터인크립션 알고리즘

본 논문에서 제안하는 워터인크립션 알고리즘은 앞서 언급한 각종 이종 망과의 연동을 통한 다양한 종류의 단말을 대상으로 개발된 H.264 SE 시스템을 기반으로 수행한다. 이 시스템에 적합한 DRM을 위한 영상에 대한 이중 보호를 수행한다. 불법배포에 따른 저작권 보호를 위해 수행하는 워터마킹은 영상 내 소유권 및 저작권 주장을 위한 워터마크를 삽입하며, 또한 여러 가지 공격에 강인한 알고리즘에 대해 연구하였다. 사용자 접근제어를 위한 암호화는 결과 영상의 열화를 일으켜 보안성을 고려하며, 인증된 표준 알고리즘을 사용 및 적용한다.

제안하는 알고리즘 적용시 고려해야 할 조건은 스케일러블 비디오 부호화 후 복호화되는 영상의 비가시성을 고려하며, H.264 SE 특성 및 구조를 만족하는 알고리즘을 연구하였다. 또한 코덱 부호화시 실시간성이 요구되므로 보다 낮은 계산량을 지니면서 빠른 연산 처리를 위해 H.264 SE 코덱 내 알고리즘을 적용하였다.

1. H.264 SE 코덱 내 워터인크립션 시스템

H.264 SE 코덱 내에서 수행하는 워터인크립션에 대해서 아래 그림 15와 같이 블록도로 나타내었다. 스케일러블 비디오 부호화의 특성을 나타내는 기본 시스템에서 각 레이어별 부호화 과정에서 워터마킹과 암호화를 수행하게 된다.

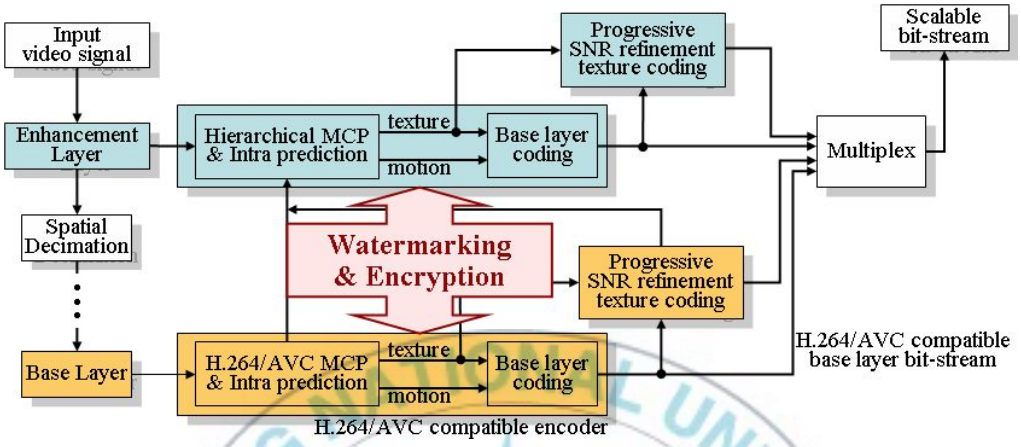


그림 15. 제안한 워터인크립션 알고리즘을 적용한 H.264 SE 전체 시스템

보다 자세한 설명을 하자면 아래 그림 16과 같이 나타낼 수 있다.

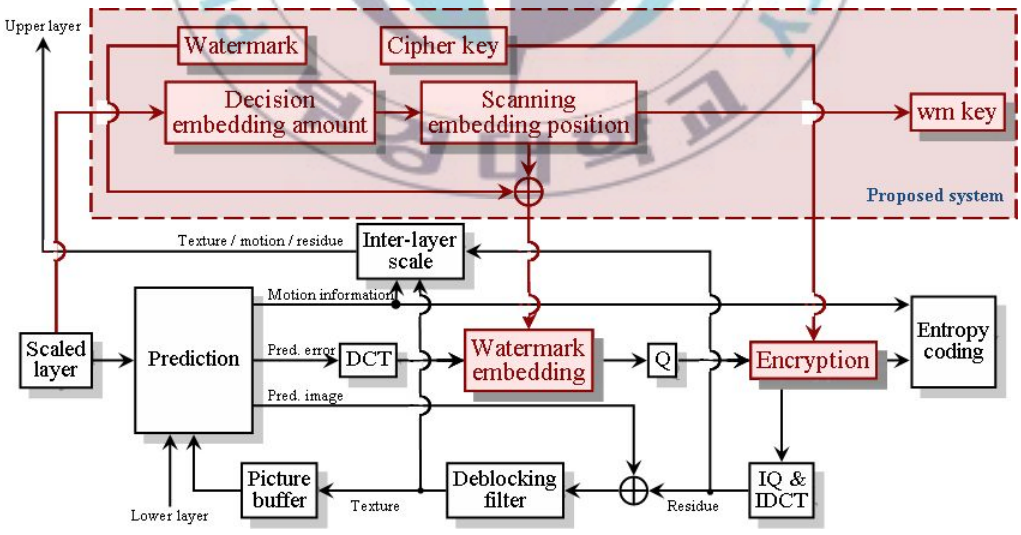


그림 16. 제안한 워터인크립션 알고리즘을 적용한 H.264 SE 세부 시스템

세부 시스템의 입력은 전체 시스템에서 공간적 축소를 수행한 레이어이다. 이 레이어에 해당하는 각 프레임에 대해서 예측과 변환을 수행을 하고, 양자화를 거쳐서 엔트로피 부호화를 하게 된다. 입력되는 레이어의 해당 프레임에 대해서 삽입할 워터마크 비트량을 결정하고 그 양만큼의 삽입 위치를 선별하여 키 값으로 저장을 한다. 이렇게 선별된 위치에 DCT 된 데이터를 사용하여 워터마크를 삽입하게 된다. 이렇게 워터마크가 삽입된 데이터를 양자화하고 엔트로피 부호화를 하기 전에 암호키에 의한 암호화를 거치게 된다.

2. 워터마크 삽입 알고리즘

본 절에서는 H.264 SE 내에서 워터마크를 삽입하는 알고리즘에 대해서 설명한다. 워터마크의 삽입은 크게 워터마크 삽입량 결정, 위치 선정 그리고 삽입으로 나눠진다.

우선, 워터마크 삽입량 결정은 입력된 비디오 프레임으로부터 *Block_n*을 구하고 이 값을 가지고 C_w와 *wm_amount*를 구하여, 마지막으로 워터마크를 삽입하게 될 위치 값 **wm_key**를 저장한다. 이렇게 저장한 **wm_key** 값은 워터마크가 삽입된 영상으로부터 워터마크를 추출할 때 키 값으로 사용되어진다.

입력되는 각 프레임별로 워터마크를 삽입할 양을 아래 식 (5)로부터 결정하게 된다.

$$Block\_n = \frac{w_f \times h_f}{16} \qquad (5)$$

여기서 w_f와 h_f는 각각 프레임의 수평 해상도와 수직 해상도를 뜻한다. 프레임 내 모든 DCT 블록의 개수 *Block_n*는 전체 프레임의 크기로부터 인트라 블록을 겹치지 않게 나누어진 개수를 의미한다.

$$C_W = floor\left[\frac{Block\_n}{wm\_length} \times m\right] \tag{6}$$

식 (6) 에서 *wm_length*는 삽입하고자 하는 워터마크 비트열의 길이를 뜻하는 것으로 인트라 블록 당 한 개의 비트씩 삽입된다. 여기서 한 개의 프레임 내에 존재하는 인트라 블록의 개수로부터 *wm_length*를 나누어 줌으로서 삽입 가능한 전체 워터마크의 개수를 알 수 있다. 본 논문에서는 워터마크가 삽입된 영상의 화질을 고려하여 전체에 삽입하지 않고, 한 개의 프레임의 m에 해당하는 영역에만 워터마크를 삽입한다. 이로써 워터마크가 반복되는 회수 C_W를 결정하게 된다.

$$wm\_amount = C_W \times wm\_length \tag{7}$$

식 (7)은 삽입하고자 하는 워터마크 비트열의 길이와 그 반복 회수의 곱으로서 한 개의 프레임 내에 삽입하게 될 총 워터마크 비트량 *wm_amount*를 구할 수 있게 된다.

다음으로 워터마크를 삽입하게 될 위치를 결정하게 되는데, 한 개의 프레임을 *Block_n*만큼 4×4정수 DCT를 수행한다. 각 DCT된 계수 값들을 식 (8)과 같이 계산하여 각각의 인트라 블록 내 화질의 복잡도 S를 구한다.

$$S_i = \sum_{1 \le u,v \le 3} |x_i(u,v)|, 0 \le i < Block\_n \qquad (8)$$

여기서, $x_i(u,v)$: i번째 인트라 DCT 블록 내의 계수값을 뜻하며, S는 인트라 DCT 블록 내의 AC 계수들의 절대 크기 합이다. 이렇게 구해진 S는 아래 그림 17과 같은 과정을 거치게 된다. S_order_i는 S_i의 크기에 따라 내림차순으로 정렬한 값이며, wm_amount만큼 추출한 값이 $S_order_{i\text{-}max}$이다. 마지막으로 $S_order_{i\text{-}max}$를 좌표 매핑을 통해서 워터마크를 삽입하고자 하는 위치값 wm_key를 구해낸다.

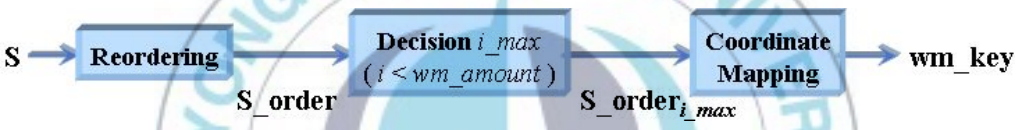


그림 17. 워터마크 삽입 위치 결정 과정

앞서 구한 wm_key값에 해당하는 위치에 워터마크를 삽입하기 위해 식 (9), 식 (10)을 사용한다.

$$wx_i = \frac{\sum_{1 \le u,v \le 3} |x_i(u,v)|}{9} \cdot (w_i + \alpha) \qquad (9)$$

$$x_i(ku_i, kv_i) = wx_i \qquad (10)$$

여기서, wx_i는 w_i및 AC계수합으로 생성된 새로운 AC계수이며, α는 워터마크 삽입강도, u, v는 AC계수의 좌표이며, ku, kv는 생성된 wx_i가 치환될 DCT 블록 내의 좌표이다.

3. 워터마크 추출 알고리즘

워터마크 추출은 복호된 영상으로부터 수행할 수 있다. 워터마크 삽입시에 사용했던 워터마크 키 값을 이용하여 워터마크가 삽입된 위치를 찾아 해당 위치의 계수 값으로부터 워터마크 비트를 결정할 수 있으며, 해당하는 수식은 아래 식 (11), 식 (12)와 같다.

$$\hat{w_i} = \begin{cases} 1, & \text{if } \tilde{x_i}(ku_i, kv_i) \geq th \\ 0, & otherwise \end{cases} \tag{11}$$

$$th = \frac{\sum_{1 \leq u,v \leq 3} \left| \tilde{x_i}(u,v) \right|}{9} \tag{12}$$

4. 암호화 알고리즘

워터마크가 삽입된 비디오에 대해서 사용자 접근제한을 설정하기 위해서 암호화를 수행한다. 사용한 암호화 알고리즘은 AES이며, 사용한 키의 길이는 128비트이다.

H.264 SE 내에 암호화를 수행하는 방법은 아래 그림 18과 같이 나타낼 수 있다.

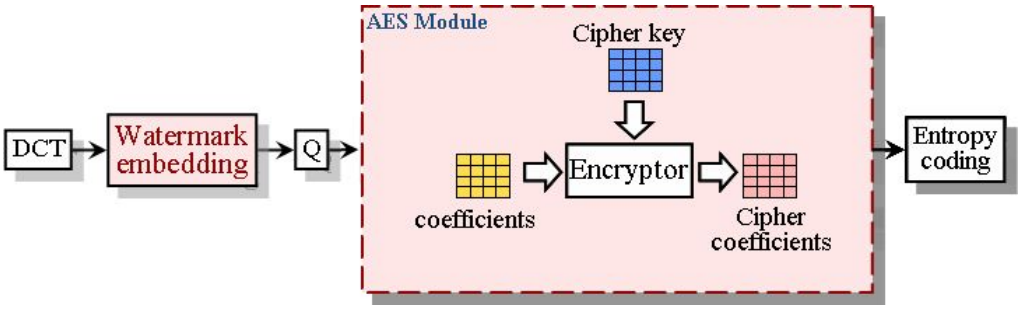


그림 18. H.264 SE 내 암호화 시스템

DCT 된 데이터에 워터마크가 삽입되고 양자화를 거친 데이터에 대해서 암호화를 수행한다. 해당 블록의 계수에 암호화 키를 사용하여 암호화된 블록의 계수를 생성해서 치환한 후 엔트로피 부호화로 내보내게 된다.

Ⅳ. 실험 결과 및 고찰

본 장에서는 비가시성과 강인성을 실험하기에 앞서 SE 특성에 대한 워터마크의 강인성을 확인해 보고 이후 본 논문에서 제안하는 알고리즘의 우수성을 Zhang등이 제안한 알고리즘과 비교하여 보고 그 결과에 대해 고찰해본다.

1. 실험 방법

본 논문에서 제안하는 비디오 워터마킹 알고리즘의 비가시성과 강인성을 실험은 H.264 SE JSVM 9.8 참조 소프트웨어를 사용하여 수행하였다[26]. 실험 영상은 Foreman, Container, Crew, 그리고 Mobile을 사용하였으며, 해상도는 QCIF(176×144), CIF(352×288) 그리고 4CIF(704×576), 프레임율은 초당 30프레임을 가지도록 하였다[27].

여기서 각 영상의 특성을 알아보면 Foreman 영상은 저주파 영상으로 국부적인 움직임이 있으며, Container 영상은 저주파 영상이며 움직임 또한 거의 없다고 할 수 있다. Crew 영상은 시간적 변화량이 점차적으로 증가하는 특징을 가지며, Mobile 영상은 고주파 영상으로 국부적인 빠른 움직임이 특징이다.

부호화 환경은 GOP 구조는 8의 크기를 가지도록 하였으며, 반복되는 인트라 픽처의 경우 32의 주기를 가지도록 하여 각 영상별로 전체 100프레임을 실험에 사용하였다.

워터마크가 삽입된 영상의 비가시성을 평가하기 위하여 부호화하지 않은 원본영상과 PSNR(Peak Signal to Noise Ratio)을 사용하여 나타내었으며, 강인성에 대한 평가는 삽입된 워터마크 비트와 공격 후 추출한 워터마크 간의 정규화된 상관도(normalized correlation)를 사용하여 수행하였다.

삽입된 워터마크의 강인성에 대한 공격 알고리즘으로는 양자화 값 30을 사용한 H.264 SE 부호화 및 1/3 트랜스 코딩과 같은 부호화 공격을 수행하였으며, 일반적인 신호처리 공격으로 대비 강화, 가우시안 필터링, 그리고 가우시안 노이즈 첨가 등을 수행하였고, 기하학적인 공격으로 절단, 회전, 그리고 변환 등을 가하였다.

실험을 수행한 하드웨어 사양은 인텔 펜티엄 4, CPU 3GHz, 램 2GB 이며, 소프트웨어 사양은 비주얼 스튜디오 2005를 사용하였다.

2. SE 특성에 대한 만족도 평가

본 논문에서 제안하는 방법에 대한 워터마크 삽입 후 영상의 화질 변화나 삽입된 워터마크의 강인성 등을 따져보기 이전에 제안하는 알고리즘이 코덱 내에서 수행하므로 SE 특성을 만족하는지를 먼저 확인해야한다. 따라서 각 변환특성에 따라 실험을 수행하였다. 공간적 스케일러빌러티는 QCIF, CIF, 그리고 4CIF 등 3가지의 크기를 사용하였으며, 시간적 스케일러빌러티는 30Hz와 15Hz를 사용하여 그 결과 값을 표 1과 같이 정리하였다.

각 특성별 강인성은 삽입 워터마크 비트 대 검출한 워터마크 비트의 비교를 비트 에러율로 나타내었다. 대부분의 수치가 0%를 나타내는데 반해 공간적 스케일러빌러티의 QCIF크기에서 나타나는 수치는 그 이상임을 알

수 있다. 본 논문에서 제안하는 알고리즘에 대한 연구가 CIF크기 위주임을 감안하여 다소 작은 영상인 QCIF크기에서 CIF크기에서 보다 적은 양의 워터마크가 삽입됨으로서 발생하는 오류도 판단된다. 하지만 그 수치는 전체 데이터의 10%미만으로 SE 특성에 충분히 만족하도록 알고리즘이 적용되었음을 알 수 있다.

이 실험으로 인해 이러한 각 변환 특성에 따라 생성된 영상에 대한 비가시성과 각 영상의 워터마크에 대한 공격 등은 CIF크기의 영상에 대한 결과값을 기준으로 판단이 가능하다는 것을 보여줌에 따라 이후 실험 영상의 크기는 CIF만을 사용하였다.

표 1. SE 특성에 따른 영상에 대한 워터마크의 강인성

Test Video	Spatial Scalability			Temporal Scalability	
	QCIF	CIF	4CIF	30Hz	15Hz
Foreman	4.69	0.0	0.0	0.0	0.0
Container	3.13	0.0	0.0	0.0	0.0
Crew	1.56	0.0	0.0	0.0	0.0
Mobile	0.0	0.0	0.0	0.0	0.0

3. 영상에 대한 비가시성 평가

본 논문에서 워터마크가 삽입되고 난 후에 영상의 열화를 알아보기 위하여 그 수치를 30프레임에 대한 평균 PSNR로 알아보았다. 워터마크를 삽입하지 않고 양자화 값 30을 사용하여 부호화한 영상(좌측 막대그래프)과 Zhang의 알고리즘(중앙 막대그래프)과 제안한 알고리즘(우측 막대그래프),

두 가지의 방법으로 워터마크를 삽입한 후의 영상의 열화를 그림 19와 같이 나타내었다. 그래프를 보게 되면 Mobile의 영상에서 큰 화질의 열화를 볼 수 있는데, 이는 영상 자체가 지니는 복잡한 영상에 의한 것으로 보여진다.

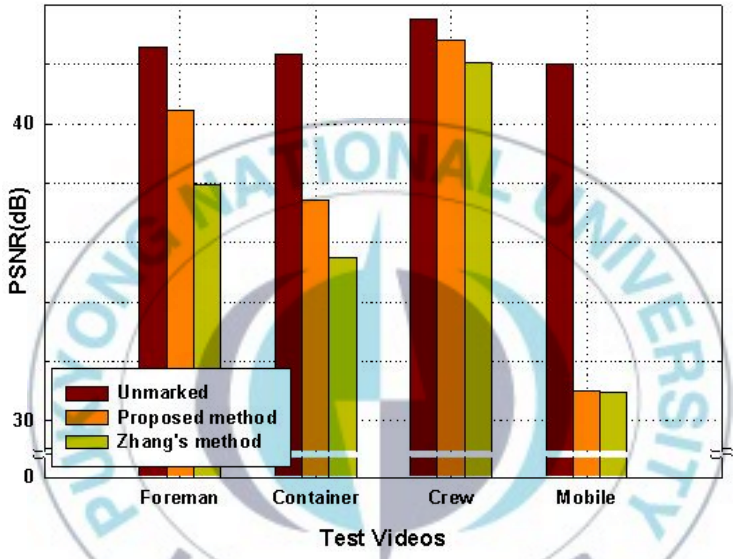


그림 19. 양자화 값 30으로 부호화한 후 워터마크가 삽입되지 않은 영상과 각 영상별 워터마크 삽입 후의 PSNR 비교

4. 워터마크에 대한 강인성 평가

본 논문에서 제안한 알고리즘의 강인성을 Zhang이 제안한 알고리즘과 실험으로 비교하여 보았다. 원본 영상에 대해서 두 가지의 알고리즘을 워터마크를 삽입한 후 부호화, 일반적인 신호처리, 기하학적인 공격 등을 수

행하여 추출한 워터마크를 원본 워터마크와의 정규상관도로 그 강인성을 평가하였다.

아래에서 보여지는 그림 20은 원본 영상에 대해서 가해지는 공격 후 영상의 변화를 알아보기 위한 그림으로 실험에 사용되는 4개의 영상에 대해서 공격이 가해진다.

일반적인 신호처리 공격에 비해 부호화 공격이나 기하학적인 공격에서 본 논문에서 제안한 알고리즘으로 삽입된 워터마크가 보다 강인함을 확인할 수 있었다.

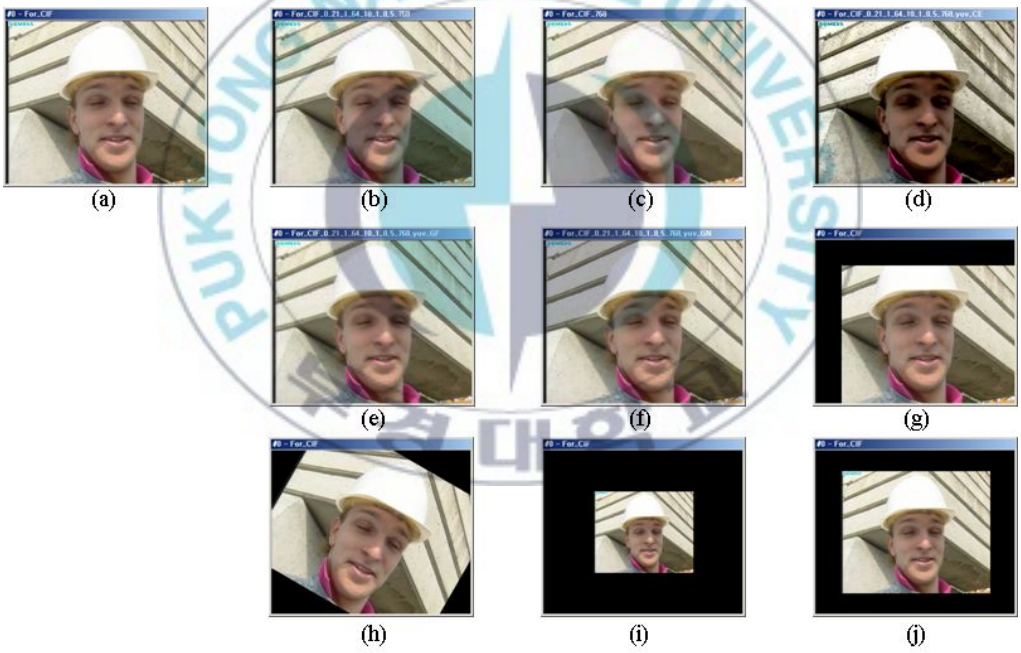


그림 20. 다양한 공격 후 영상 (a)Original (b)Encoding (c)Trans-coding (d)Contrast Enhancement (e)Gaussian Filtering (f)Gaussian Noise (g)Cropping (h)Rotation (i) 0.5 Scaling (j) 0.75 Scaling

또한, 각 영상별로 공격에 의한 실험값을 그림 21-24에 순서대로 Foreman, Container, Crew, 그리고 Mobile의 영상을 나타내었다.

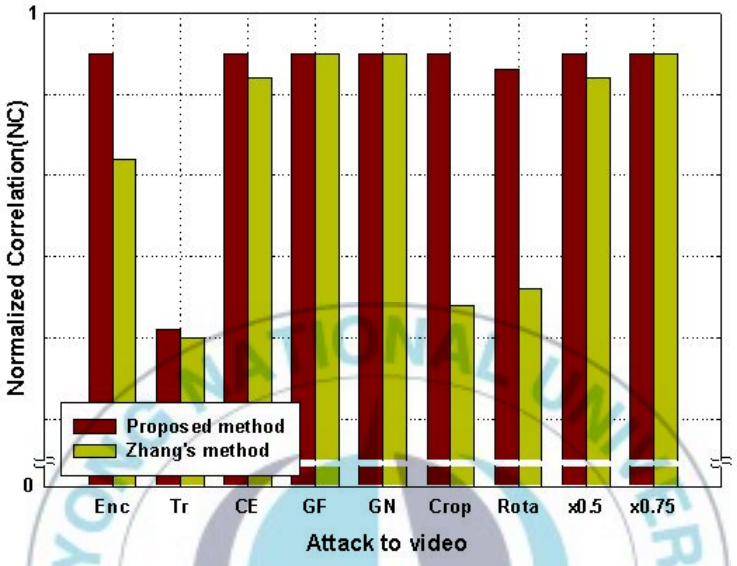


그림 21. Foreman 영상에 대한 공격 후 워터마크 강인성 실험

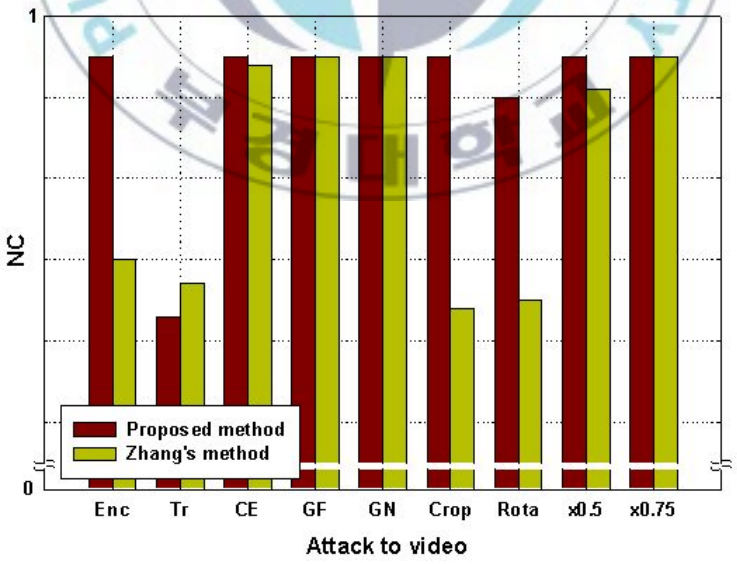


그림 22. Container 영상에 대한 공격 후 워터마크 강인성 실험

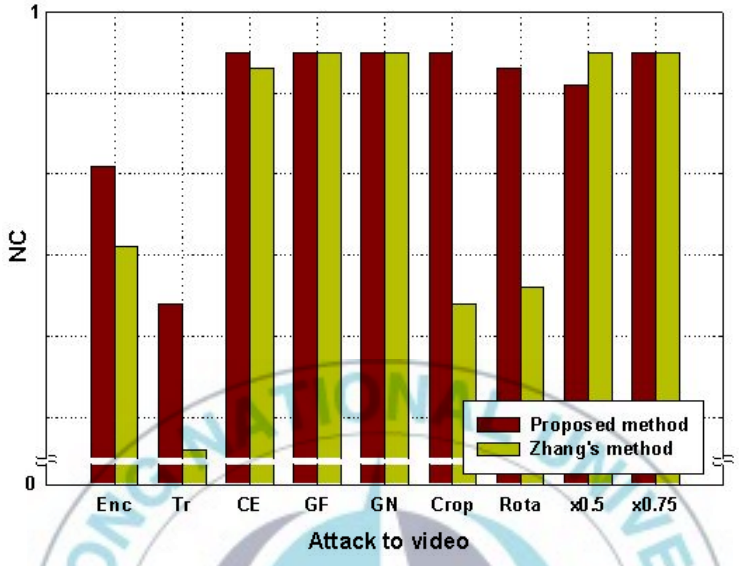


그림 23. Crew 영상에 대한 공격 후 워터마크 강인성 실험

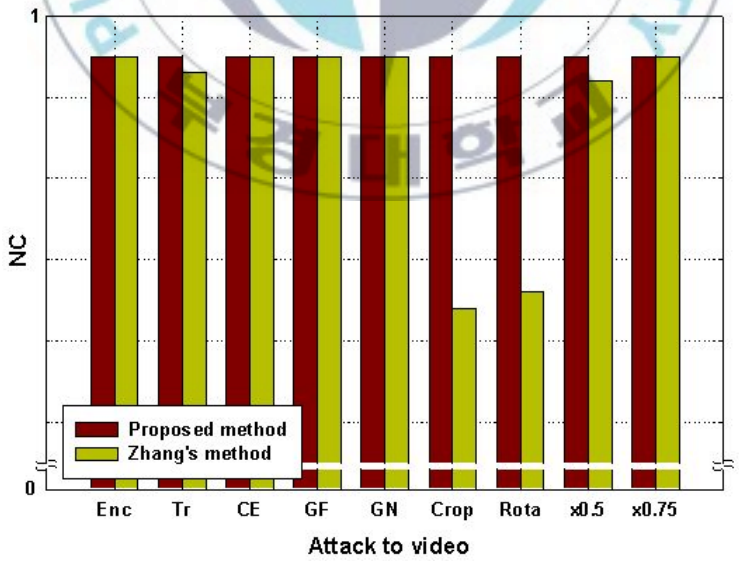
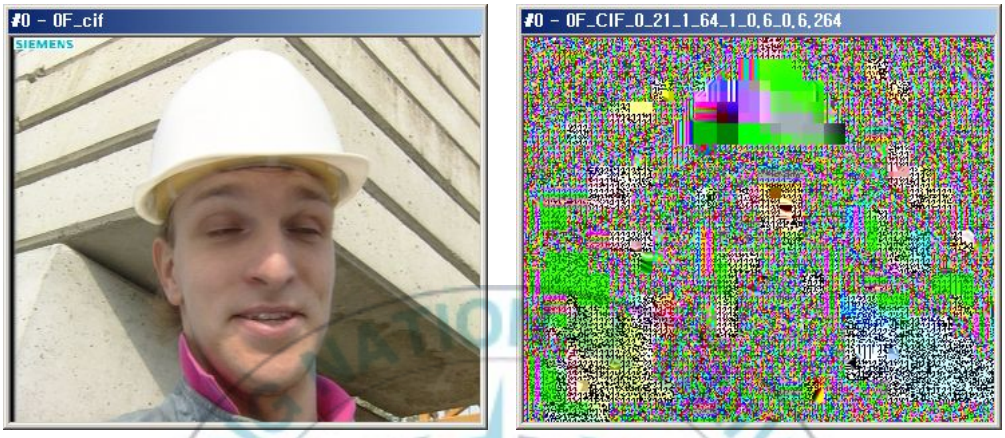


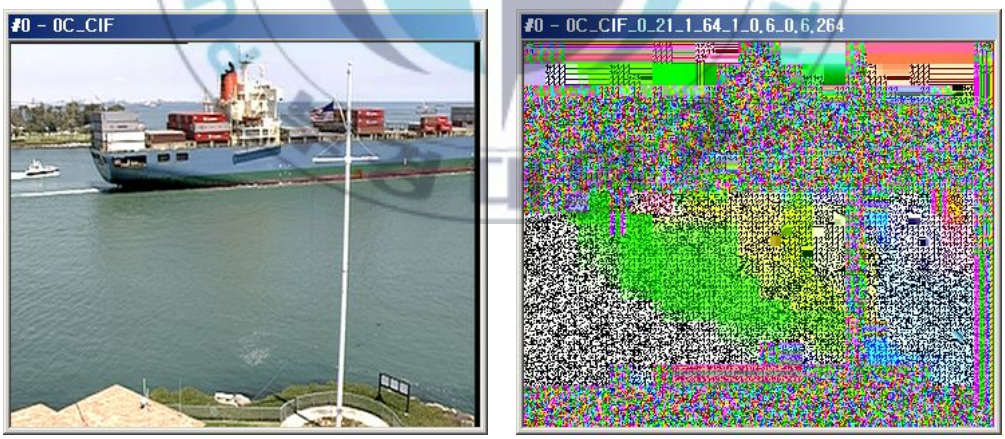
그림 24. Mobile 영상에 대한 공격 후 워터마크 강인성 실험

5. 영상에 대한 보안성 평가



(a) (b)

그림 25. Foreman의 (a)원본영상과 (b)암호화된 영상

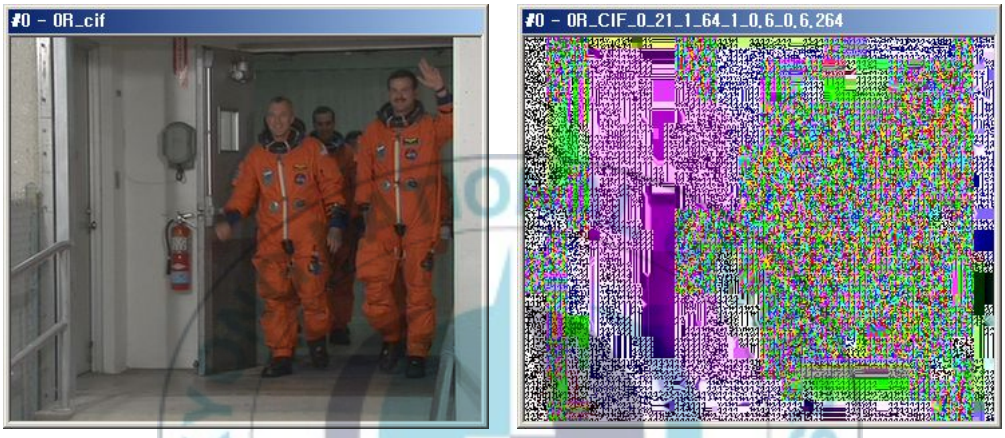


(a) (b)

그림 26. Container의 (a)원본영상과 (b)암호화된 영상

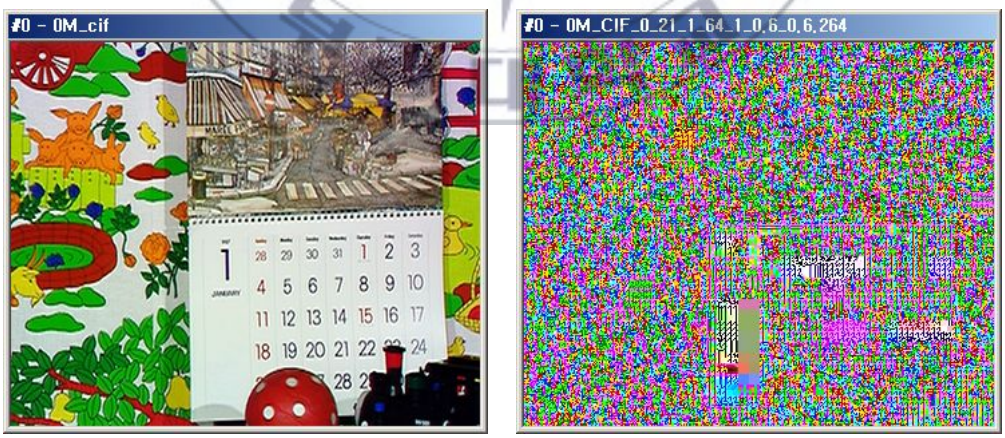
워터마크를 삽입하고 난 후 사용자의 접근 제어를 위해서 암호화를 수행

하고 그 결과 값을 원본 영상과 비교해 보았다. 좌측 (a)영상이 원본 영상이며 우측 (b)영상이 암호화된 영상으로 그림 25-28에 나타내었으며, 순서대로 Foreman, Container, Crew, 그리고 Mobile 의 영상을 나타내었다. 그림에서 보는 바와 같이 각 영상들은 암호화에 의해 지각적인 보안성을 확인할 수 있었다.



(a) (b)

그림 27. Crew의 (a)원본영상과 (b)암호화된 영상



(a) (b)

그림 28. Mobile의 (a)원본영상과 (b)암호화된 영상

6. 부호화에 대한 속도 평가

마지막으로 부호화시 실시간 처리를 만족하기 위한 실험을 수행하였다. 실험조건은 CIF사이즈의 각 영상들을 양자화 값 30을 사용하여 100프레임에 대해서 부호화하였다. 각 막대그래프는 왼쪽부터 순서대로 알고리즘을 적용하지 않고 수행되는 시간과 워터마킹 적용시, 암호화 적용시, 그리고 워터마킹과 암호화 둘 다 수행시의 시간을 측정해서 그림 29에 나타내었다.

측정 결과는 각 영상에 대해 부호화된 시간(초)로 나타내었으며, 1프레임의 평균 지연시간은 1/100초 미만으로 처리시간을 만족함을 확인하였다. 그림 29를 보게 되면 원본 부호화시 시간에 비해 암호화 수행시간이나 워터마킹 적용시 시간에 비해 두 가지, 모두 적용시 시간이 Foreman과 Crew 영상에서 짧게 나타난 것을 볼 수 있다. 이러한 부호화 시간의 감소는 원본 데이터가 암호화를 거치면서 데이터 변형이 일어나면서 엔트로피 부호화 과정에서 시간의 변화가 일어난 것으로 판단된다.

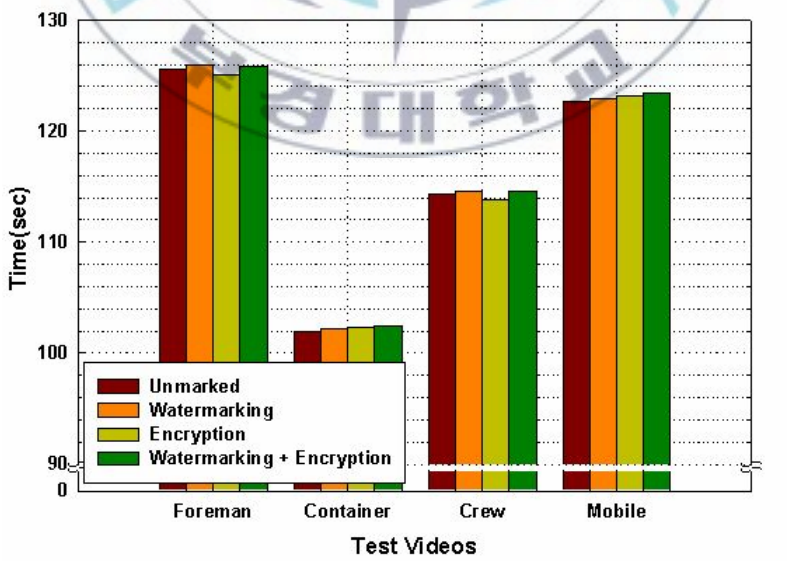


그림 29. 각 영상별 부호화 수행시간(QP: 30)

Ⅴ. 결 론

본 논문에서는 H.264 SE에 적합한 DRM을 위한 영상에 대한 이중보호를 목적으로 연구하였다.

불법 배포에 따른 저작권 보호를 위한 워터마킹에 대해서 다양한 공격을 거쳐 추출한 워터마크의 강인성이 우수함을 확인하였으며, 원본 비디오와 워터마크가 삽입된 비디오의 PSNR을 비교하여 비가시성이 우수함을 확인하였다.

사용자 접근 제어를 위해서 워터마크가 삽입된 데이터를 암호화할 때 사용한 알고리즘은 인증된 표준 AES 알고리즘을 사용하였으며, 암호화된 영상을 원본 영상과 비교함으로써 지각적인 보안성을 확인하였다.

마지막으로 코덱 부호화시 요구되는 실시간성은 기존의 부호화 시간과 제안한 알고리즘 적용 후 발생한 지연 시간을 비교함으로써 실시간 처리에 적합함을 확인하였다.

향후 H.264 SE 코덱이 보다 일반화되고 범용성을 띄게 될 것을 대비하여 보다 활발한 연구가 필요할 것으로 보여진다. 현재 워터마크 삽입시 생성하는 키를 사용하지 않고 추출하는 알고리즘이나 기하학적인 공격 후 원본 영상에 대한 블라인드 워터마킹 알고리즘 그리고 한 개의 프레임에 대한 워터마킹 보다 프레임 간의 영상 정보를 이용한 효과적인 워터마킹 알고리즘 연구 등이 필요할 것으로 보인다. 무엇보다도 암호화된 데이터에 대해서 인증받은 사용자를 위한 인증키를 워터마크로 사용함으로써 인증받은 사용자에 의한 불법배포 추적이 가능할 것으로 판단되며, 이것에 대한 심도 있는 연구가 행하여 졌으면 하는 바람이다.

Ⅳ. 참고문헌

[1] 강정원, 김재곤, 홍진우, “통방융합 유비쿼터스 콘텐츠 서비스 기술,” *전자통신동향분석* 제 21권 제 4호 2006년 8월.

[2] *Video Codec for Audiovisual Services at p*×64 kbit/s, ITU-T Rec. H.261, ITU-T, Version 1: Nov. 1990, Version 2: Mar. 1993.

[3] *Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to About 1.5 Mbit/s-Part 2: Video,* ISO/IEC 11172-2(MPEG-1 Video), ISO/IEC JTC 1, Mar. 1993.

[4] *Generic Coding of Moving Pictures and Associated Audio Information-Part 2: Video,* ITU-T Rec. H.262 and ISO/IEC 13818-2 (MPEG-2 Video), ITU-T and ISO/IEC JTC 1, Nov. 1994.

[5] *Video Coding for Low Bit Rate communication*, ITU-T Rec. H.263, ITU-T, Version 1: Nov. 1995, Version 2: Jan. 1998, Version 3: Nov. 2000.

[6] *Coding of audio-visual objects-Part 2: Visual*, ISO/IEC 14492-2 (MPEG-4 Visual), ISO/IEC JTC 1, Version 1: Apr. 1999, Version 2: Feb. 2000, Version 3: May 2004.

[7] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264 and ISO/IEC 14496-10 (MPEG-4 AVC), ITU-T and ISO/IEC JTC 1, Version 1: May 2003, Version 2: May 2004, Version 3: Mar. 2005, Version 4: Sept. 2005, Version 5

and Version 6: June 2006, Version 7: Apr. 2007, Version 8 (including SVC extension): Consented in July 2007.

[8] ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6, "Joint Draft ITU-T Rec. H.264 | ISO/IEC 14496-10 / Amd.3 Scalable video coding," 24th Meeting: Geneva, Switzerland, 29 June - 5 July, 2007.

[9] W. Li, "Overview of fine granularity in MPEG-4 video standard," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 11, pp. 301-317, Mar. 2001.

[10] A. Vetro, C. Christopoulos, and H. Sun "Video Transcoding Arghitectures and Techniques: An Overview," *IEEE Signal Processing Mag.*, vol. 20, pp. 18-29, Mar. 2003.

[11] M. Wien, H. Schwarz, and T. Oelbaum, "Performance analysis of SVC," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 17, no. 9, pp. 1194-1203, Sep. 2007.

[12] S. Pateux, Y.K. Wang, M. Hannuksela, and A. Eleftheriadis, "System and transport interface of the emerging SVC standard," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 17, no. 9, pp. 1149-1163, Sep. 2007.

[13] S. Wenger, and T. Schierl, "RTP payload for SVC," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 17, no. 9, pp. 1204-1217, Sep. 2007.

[14] D. Singer, T. Rathgen, and P. Amon, "File format for SVC," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 17, no. 9, pp. 1174-1185, Sep. 2007.

[15] J.R. Ohm, "Advances in scalable video coding," *Proc. IEEE*, vol. 93, no. 1, pp. 42-56, Jan. 2005.

[16] M. Winken, H. Schwarz, D. Marpe, and T. Wiegand, "Adaptive refinement of motion information for Fine-granular SNR scalable video coding," *presented at the EuMob, Alghero, Italy*, Sep. 2006.

[17] F. Hartung, and B. Girod, "Digital watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283-301, May 1998.

[18] C. T. Hsu, and J. L. Wu, "DCT-based watermarking for video," *IEEE Trans. Consumer Electronics,* vol. 44, no. 1, pp. 206-216, 1998.

[19] W. Zhu, Z. Xiong, and Y. Q. Zhang, "Multiresolution watermarking for images and video," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 9, no. 4, pp. 545-550, June 1999.

[20] Jing Zhang, Anthony T. S. Ho, Gang Qiu, and Pina Marziliano, "Robust Video Watermarking of H.264/AVC," *IEEE Trans. Circuits Sys. Video Tech.*, vol. 54, no. 2, pp. 205-209, February 2007.

[21] Shamir, A. "On the security of DES," *Advances in Cryptology, Proc. Crypto '85*, pp. 280-285, Aug. 1985.

[22] Miles E. Smid, "From DES to AES," 2000.

[23] NIST, "Announcing the Advanced Encryption Standard(AES)," *FIPS PUB ZZZ*, 2001.

[24] Daemen, J., and Rijmen, V. "AES Proposal: Rijndael, Version2.," *Submission to NIST*, March 1999.

[25] R. Rivest, A. Shamir and L. Adleman. "A Method for Obtaining Digital Signature and Public key Cryptosystem," *Communications of the ACM*, Vol. 21. No. 2. pp. 120-126. Feb. 1978.

[26] J. Reichel, H. Schwarz and M. Wien, "JSVM 9.8 Software," *Joint Video Team of ISO/IEC MPEG and ITU-T VCEG* N9212, June 2007, Geneva, Switzerland.

[27] ftp://ftp.tnt.uni-hannover.de/pub/svc/testsequences/

[28] Bong-Joo Jang, Won-Jei Kim, Seong-Geun Kwon, Suk-Hwan Lee, Kwang-Seok Moon, Jai-Jin Jung, and Ki-Ryong Kwon, "Real-time digital watermarking based on harr wavelet transform on mobile phone camera," *MITA2007*, pp. 5-8, August 2007.

[29] 김원제, 성택영, 이석환, 문광석, 서용수, 권기룡, "H.264/SVC의 블록 단위 비디오 워터마킹," *한국멀티미디어학회 춘계학술발표대회 논문집*, pp. 45-48, 2008년 5월.

[30] Won Jei Kim, Kwang-Seok Moon, MD Abul Bashar, Young-Ho Ahn, Jun-Hee Kim, Suk-Hwan Lee, Yong-Su Seo, and Ki-Ryong Kwon, "Random Block-Based Video Watermarking of H.264/SVC," *MITA2008*, pp. 290-293, July 2008.

[31] 김원제, 성택영, 문광석, 이석환, 권기룡, "H.264 SE 기반의 인트라 프레임을 이용한 비디오 워터마킹," *한국멀티미디어학회 추계학술발표대회 논문집*, pp. 214-217, 2008년 11월.

감사의 글

2009년! 새로운 한 걸음을 내딛을 때가 다가온 이 시점에서 지난 날을 돌이켜보면 아쉬움이 많이 남습니다. 하지만 매 순간 부족할 때가 있었겠지만 최선을 다했기에 지금의 제가 있다고 생각합니다.

학구열에 불타 학업에 대한 목마름으로 대학원을 가고자 하였던 그 시절, 불가피하게 아무 준비도 없이 산업전선에 뛰어 들었다가 쓰디쓴 사회의 참맛을 보고 아직은 부족한 제 자신을 키울 수 있게 저를 받아주신 아버지같은 **권기룡 교수님**께 먼저 감사의 마음을 전합니다. 좌충우돌 혼자만 생각하며 그렇게 살아왔던 제 생에서 어울림의 중요성과 필요성을 절실히 깨닫게 해주셨고, 나아가 타인을 배려함으로서 제 자신이 더욱 빛날 수 있다는 그런 진리를 깨닫게 해주셔서 정말 감사드립니다. 그리고 마지막까지 그렇게 많은 신경을 써주시고, 배려해주심에도 불구하고 기회를 져버리지 않고 실망을 안겨드려 죄송합니다. 한 번의 실수는 병가지상사란 말이 있듯이 사회에서 할 실수를 미리 다 경험하고 나가 열심히 살아가도록 노력하겠습니다.

그리 길지 않은 만남이었지만 잘못된 저의 행동 하나하나에 일침을 가함으로서 안일한 저의 대학원 생활을 깨어있도록 살펴주신 **문광석 교수님**께도 감사의 마음을 전합니다.

사회 초년생으로 나아가기 전에 보다 나은 사람으로 만들고자 기초와 원리를 바탕으로 한걸음 진보한 생각을 가질 수 있게 지도를 아끼지 않으셨던 **이석환 교수님**께 죄스런 마음과 함께 감사드립니다. 또한 일반대학원 석사과정동안 많은 도움을 주셨던 학과의 교수님들께 진심으로 감사드립니다.

연구실의 맏형으로써 잘할 때는 칭찬과 격려로, 때때로 방황할 땐 질타와 채찍질로 바로 잡아준 **택사마 택영이 형**, 나이가 같다는 이유로 어설

프게 기싸움을 벌였지만 뒤로는 누구보다 많은 배려를 해주었던 **저의 벗 봉주**, 나이는 어리지만 한 해 먼저 석사과정을 수료하고 많은 도움과 어설픈 충고를 해 주었던 **따당 혜정이**, 늘 피곤하다, 죽겠다를 입에 달고 살며 출퇴근의 성실로 무장한 **골초 지훈이**, 함께한 시간이 일년 남짓 밖에 되지 않아 선배로서 신경을 못써줘서 너무 미안한 일학년 후배들, 똘똘하면서도 멋지지만 맘보의 귀만큼 팔랑거리는 **귀얇은 성혜**와 비록 중간에 일이 있어서 그만 두었지만 고생이란 고생은 다하고간 **곰탱이 승환이**, 우직하면서도 발끈 잘하는 **이크! 진호**, 말도 많고 탈도 많았으면서도 연구실의 맏언니같이 많은 힘이 되어준 **녀석! 준희**, 우리 연구실과 떼려야 땔 수 없는 한국멀티미디어학회 과장을 하면서도 연구실 생활한다고 고생한 **영호형**, 따로 떨어져 교류가 모자라 아쉬웠던 외대 후배 **성찬이**와 **현덕**. 돌이켜보면 너무나 고마웠던 **우리 MCSP 연구실 선배님**들께도 이름이 일일이 나열하지 못하는 죄스런 마음과 함께 감사의 마음을 전합니다. 또한 부경대학교 대학원의 미처 언급하지 못한 선후배님, 그리고 동기들에게 감사드립니다.

철없는 아들이 하는 일이라면 아무 조건없이 믿어주시고, 또 뒷바라지해 주셨던 **아버지**와 **어머니**께 너무 감사드리고 사랑합니다.

마지막으로 골치 아프도록 고집 센 절 곁에서 응원해 준 친구들, 그 **친구들**에게 이 고마움을 전하며 감사의 글을 마무리하고자 합니다.

산을 오르면서 흘렸던 그 땀, 땀으로 맺은 정은 피보다 진하다란 말이 있듯이 그 정으로 더더욱 발전하는 멋진 MCSP 연구실이 되길 기원하며 다시 한번 감사드립니다.

2009년 기축년 새해의 희망을 담아

김원제 드림.