산학기술성공학회논문지
Vol. 3, No. 4, pp. 308-312, 2002

JPEG 암호화에 관한 연구

박용범* · 박종일

A Study on the Cipher JPEG Imag

Young B. Park* and Jong-Il Park

요 약 최근 인터넷 환경의 보급에 힘입어 멀티미디어 자료들의 비중이 점점 늘어나고 있다. 그 중에서 가장 범용적으로 사용되고 있는 멀티미디어 자료 표현 방법 중의 하나인 JPEG 파일의 경우에는 암호화나 특정 사용자에 대한 접근권한등에 특별한 대안이 없는 것이 사실이다. 이에 본 논문에서는 JPEG 파일에 대하여 자료의 보안과 허가된 사용자만이 접근이 가능하게 하기 위한 방법으로 암호화를 지원하며, 암호화 알고리즘은 간단한 비트교환부터 DES 등의 다양한 암호방법론이 적용 가능한 방법을 제시하였다. 데이터 암호화는 JPEG에서 복원을 할 때 가장 중요하게 사용되는 영역인 허프만 테이블과 비교를 위하여 이미지영역까지 확대하여 암호화를 수행하였다. 또한 이들 각각의 방법론들에 대한 비교분석을 통하여 임의의 환경 하에서의 가장 적합한 암호화 방법론의 선택기준을 살펴 보았다.

Abstract Recently, Internet is getting more popular and the usage of multimedia contents is getting more increased. Among the multimedia contents, the JPEG format is the most commonly used Image data format but the JPEG format doesn't supply cryptographic methods and access control. In this paper, a cryptographic method for cipher JPEG Image is proposed. This method can supply encryption and decryption using several algorithms such as DES to the JPEG Image data. Huffman table, which is the most important data area for JPEG data coding, is chosen as the coding data. Each method is compared to select the optimal cryptographic method in a certain environment.

Key Words : Multimedia data, JPEG, Cryptograph, Data Coding

1. 서 론

최근 많이 보편화된 인터넷 환경을 비롯한 수많은 컴퓨팅 환경에서 데이터 표현방법이 점점 다양화되고 시각화 되고 있으며 이에 따라 멀티미디어 자료들이 점점 늘어나고 있다. 하지만 멀티미디어 자료에 대한 손쉬운 접근이 오히려 데이터를 보호하고 지적 소유권을 주장하는데 장애물로서 나타나게 되었다[9].

특히 도면이나 서류를 다루는 기업체 환경에서는 JPEG와 같은 이미지 포맷을 이용하여 멀티미디어 자료화하는 경향을 가지고 있으며 이러한 자료에 대한 지적 소유권 문제와 데이터에 대한 보안이 중요한 이슈로 대두된다. 따라서 원치 않는 불특정 다수에게 데이터가 노출되지 않고 안전하게 허가된 사용자에게만 데이터에 대한 접근이 허락되는 방법이 필요하다.

위와 같은 문제를 해결하기 위한 방법으로 멀티미디어 자료 중 영상자료를 JPEG 파일을 기반으로 하고 파일 내부에 추가적인 코드를 사용하여 암호화 하는 방법을 제안하게 되었다. 우선 본 논문에서는 실제 테스트를 위하여 접근이 용이한 기본 JPEG파일 포맷을 사용하였으며, 향후 과제로써 MPEG에도 동일한 방법으로 테스트를 할 예정이다. 테스트 방법으로는 파일 내부에 특정한 코드를 사용함으로써 암호화를 하였다는 신호를 추가한 후 특정 영역 데이터에 대한 암호화를 수행하였다.

이 결과로 데이터 자체에 대한 보안성을 높이고, 설정된 암호를 알고있는 특정 사용자만이 파일에 대한 접근권한을 가지게 할 수 있었다. 위와 같은 방법으로 암호화가 되어 있다면 암호를 알지 못할 경우에는 자료가 노출되더라도 데이터 자체를 복원하지 못하므로 자료에 대한 보안성을 유지할 수 있다.

*단국대학교 전자계산학과 부교수
Tel: 016-438-8840

2. JPEG 파일구조

JPEG은 정지영상의 표준화를 위하여 국제전문가회의(Joint Photographic Expert Group)에서 규정한 표준이다. 기본적으로 JPEG은 인간 비전 시스템의 한계를

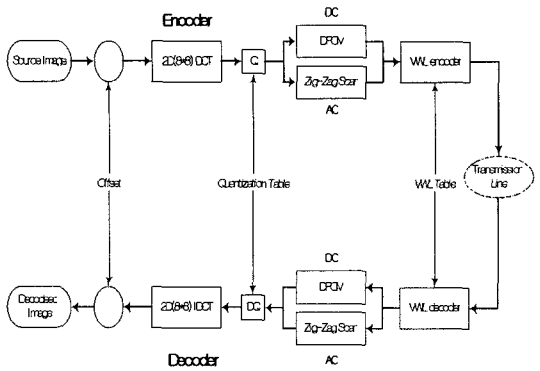


그림 1. JPEG의 압축 및 복원.

이용한 것으로 인간의 시각 시스템은 휘도 신호(luminance)에는 민감하지만, 색차 신호에는 둔감하다는 것과 일반적인 압축의 대상으로 삼고 있는 사진과 같은 자연의 영상이 인접한 픽셀간의 픽셀 값이 급격히 변하지 않는다는 속성을 이용한다[5, 7].

JPEG의 압축 과 복원되는 과정을 살펴보면 다음과 같다. 우선 영상을 휘도(Y)와 색차신호(R-Y, B-Y)로 변환하고 컬러 구성요소를 줄이기 위하여 색차 신호 다운샘플링하며 영상을 8X8 화소의 블록들로 분할하고 각 블록에서 DCT를 실행한다. DCT 결과 후에 DCT 계수를 양자화하고 각각의 계수를 지그재그 스캔을 이용하여 허프만 코딩과 같은 엔트로피 코딩을 함으로써 압축을 한다. 복원은 압축의 역순으로 진행하면 된다[2, 5, 6]. 아래 그림 1은 위의 내용을 나타낸다.

JPEG에서 파일구조를 살펴보면 계층형태로 구성되어 있으며, 전체영상 → 프레임 → 스캔 → 스캔과 연결된 실제 데이터로 구성되어 있고, 각각은 마커 코드에 의해 분리되어 진다.

위의 그림 2은 JPEG이미지 데이터의 전체적인 구조로써 첫번째 테이블은 JPEG이미지의 전체적인 모습을 나타내고 있으며 밑에 있는 테이블 일수록 상세하게 분류하여 보여준다.

SOI라고 하는 부분은 Start of Image라는 뜻으로 영상 전체의 시작점이다. 그 다음으로 TABLES라고 하는

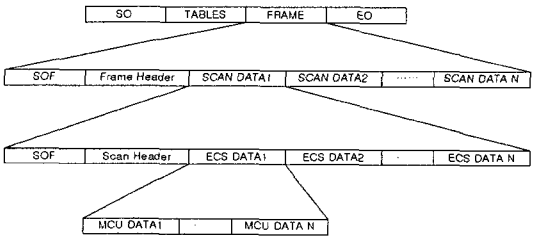


그림 2. JPEG 파일의 계층적 구조.

부분은 영상정보를 다시 복원하고 싶을 때 사용하는 테이블 정보이며, EOI(End of Image)는 영상전체의 끝부분을 나타낸다.

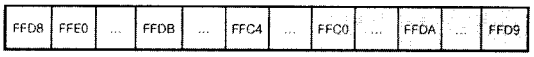
JPEG이미지를 출력할 경우에는 허프만 테이블과 양자화 테이블이 필요하다. 중간에 있는 FRAME은 JPEG 방식 중에서 Baseline DCT방법을 사용한다면 일반적으로 하나의 FRAME 만을 사용하고 Baseline DCT방법이 아닌 다른 방식의 경우에는 하나 이상의 FRAME이 사용되는 경우도 있다. FRAME안에는 영상 데이터 정보가 들어가게 되며 여러 개의 SCAN DATA라고 하는 부분이 있는데 이 부분은 스캔 정보와 여러 개의 MCU라고 하는 블록으로 구성된다.

JPEG의 파일 구조는 기본적으로는 위와 같은 순차적인 방법으로 구성이 되지만, 특별한 경우에는 별도의 처리를 위하여 순서대로 구성이 만들어지지 않을 수도 있다. 이 경우 JPEG에서는 특정한 마커코드를 이용하여 그 마커코드에 따른 처리가 이루어진다. JPEG에서

표 1. 마커코드의 구조와 종류

마커코드 (2Bytes)	마커코드길이 (2Bytes)	데이터(최대65533까지 저장가능)

Code	Symbol	Description
FFC0-FFCF	SOF_0-SOF_{15}	Start of Frame Marker
FFC4	DHT	Define Huffman Table(s)
FFCC	DAC	Define arithmetic coding
FFC3	SOF_3	Lossless (sequential)
FFD0-FFD7	RST_M	Restart with modulo 8, # 'm'
FFD8	SOI	Start of image
FFD9	EOI	End of image
FFDA	SOS	Start of scan
FFDB	DQT	Define quantization table
FFDC	DNL	Define number of lines
FFDD	DRI	Define restart interval
FFDE	DHP	Define hierarchical progression
FFDF	EXP	Expand reference component
FFE0-FFEF	APP_N	Reserved : application segments
FFF0-FFFD	JPG_N	Reserved : JPEG extensions
FFFE	COM	Comment
FF01	TEM	Temporary private used
FF02-FFBF	RES	Reserved



FFD8 : Start of Image (SOI)
FFE0 : JFIF Marker
FFDB : Quantization Table Marker
FFC4 : Huffman Table Marker
FFC0 : Frame Marker
FFDA : Scan Header Marker
FFDD : Restart Interval Marker
FFD9 : End of Image (EOI)

그림 3. JPEG 파일의 순차적 구조.

의 마커코드는 명령이 나타나는 것을 의미하는 것이며, 뒤에 오는 데이터는 이것에 대한 인수값을 나타낸다.

JPEG파일은 정보를 기록하기 전에 최상단에 마커코드를 위치시키며, 이 마커코드에 의해서 필요한 정보를 읽을 수 있는데, 이것은 2바이트 형태로서 첫째 코드는 'FF'로 시작하고, 그 이후에 'C0~FE'까지의 코드를 사용한다. '02~BF'까지의 코드와 '00' 및 'FF'는 둘째 코드로 사용되지 않고 만일 허프만 부호중에 'FF'가 생길 경우 'FF'뒤에 '00'을 삽입, 마커코드가 아님을 알려준다.

마커코드는 표 1과 같은 일반적인 형태를 가지며, 마커의 종류로는 다음과 같은 것들이 존재한다.

JPEG 파일을 순차적 데이터로 보면 그림 3.과 같은 형태로 되어 있다.

3. 암호화 방법론

위의 그림 3에서 처럼 JPEG 파일의 경우에는 각각의 계층은 마커코드로서 구분된다. 따라서 확장된 마커코드를 이용하면 암호화 정보를 제공할 수 있다. 또한 파일을 암호화시키기 위해서는 어떤 데이터 영역에 어떠한 암호화 알고리즘을 사용할 것인지를 결정하여야 하는데 이것이 실제 효율을 판가름하는 가장 중요한 요소가 된다.

3.1 암호화 영역

앞에서 논의한 바와 같이 파일에 대하여 암호화를 수행하게 될 때는 고려해야 할 사항으로 어떤 부분을 암호화 할 것인가 하는 것이 중요하다. 자료에 보안을 제공하는 방식으로 파일 전체를 대상으로 암호화 하는 방법과 중요하다고 판단되는 자료만을 암호화 하는 두 가지 방식이 고려사항이 될 수 있다. 그리고 암호화를 수행해야하는 영역의 크기에 따라 암호화 시간이 차이가 나기 때문에 비슷한 정도의 보안성을 가지고 있다면 적은 영역을 암호화 하는 것이 효율적이다.

따라서 본 논문에서는 전체를 암호화를 하는 것보다는 파일의 수정을 최소화하고 복호화를 수행하는데 걸리는 시간을 줄이기 위한 방법으로 JPEG 파일을 복원

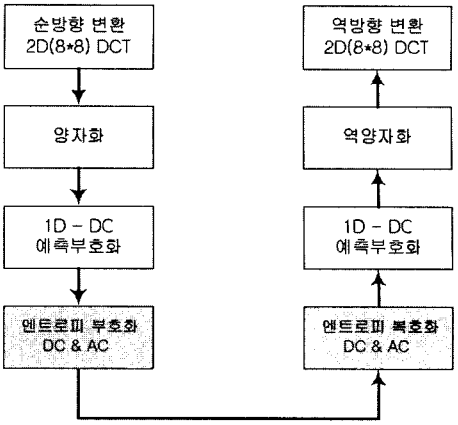


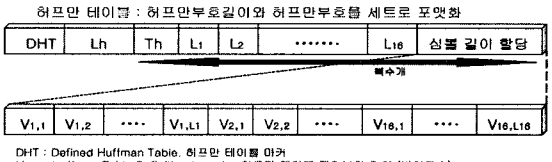
그림 4. JPEG 코드화 및 복호화 과정.

하는데 키값으로 사용되는 허프만 테이블[3]을 암호화하는 대상으로 사용했다.

위의 그림 4를 살펴보면, JPEG 코드화에서는 원본 이미지가 순방향 DCT를 수행한 후에 양자화를 거치고 예측부호화를 수행한 다음에 엔트로피 부호화를 수행한다. 즉, 허프만 테이블이 구성되려면 기존의 이미지로부터 그림 4와 같은 각각의 단계를 거치면서 이미지 자체가 캡슐화 되고 코드화 된다. 허프만 테이블이 코드화의 마지막 단계에 존재한다는 것은 각각의 모든 단계의 코드화가 반영이 되었다는 것을 의미한다. 게다가 복호화 과정에 있어서도 허프만 테이블은 엔트로피 복호화를 수행하는데 있어서 꼭 필요한 내용으로써 정상적으로 복원되지 않으면 예측 부호화 및 역 양자화를 비롯하여 IDCT 등을 수행할 수 없게 되므로 정상적인 이미지를 복원할 수 없게 된다[2].

따라서 허프만 테이블에 암호화를 적용하는 것은 JPEG의 기본압축을 통한 코드화 방법에 더불어 적용되어지는 것으로 허프만 테이블에 암호화를 적용함으로써 JPEG코드화 각 단계에 영향을 미쳐 전체 파일에 암호화 영향을 줄 수 있어 효율적이다.

위의 그림 5에서 보면 허프만 테이블을 지시하는 마커코드가 나타나고 마커의 길이가 2 바이트 나타나게 된다. 따라서 암호화 영역은 마커코드의 길이에서 총 4



DHT : Defined Huffman Table. 허프만 테이블 마커
Lh : Huffman Table Definition Length. 허프만 테이블 정의부의 길이 (바이트수)
Th : Huffman Table Identifier. 허프만 테이블 번호. Th에 가능할 수 있는 번호는 기본방식으로, DC 계수에는 00H, 01H, AC 계수에는 01H.11H의 네 종류이다.(Li는 16진수의 의미)
Li : Number of Huffman Codes of Length i. 부호길이 i 인 허프만 부호의 수
Vi,j : Value Associated with Each Huffman Code. 각 부호길이 i 에 할당된 허프만 부호

그림 5. 허프만 테이블의 구조.

바이트(마커코드와 길이)를 제외한 허프만 테이블 전체가 된다.

본 논문에서는 허프만 테이블의 마커코드와 길이를 제외한 나머지 부분을 암호화하는데 사용하였다.

3.2 암호화 알고리즘

본 논문에서는 현재 사용되고 있지 않은 마커코드의 예약된 범위 중에서 하나를 사용하여 암호화 하였다. 암호화 하였다는 신호와 함께 데이터 영역에는 암호화 방법론을 저장하여 다양한 암호화 방법을 적용할 수 있도록 구성하였다. 즉, 암호화를 지시하는 마커코드 뒤에 암호화 방법론을 인자로 지정하여 특정 값으로 비트열을 XOR 시키거나 블록암호인 DES[1, 4]까지 다양하게 사용할 수 있도록 했다.

본 논문에서는 가장 대표적으로 적용할 수 있는 두 가지 방법을 선택하여 실험해 보았다. 일반적으로 임의의 데이터를 암호화한다고 하면 크게 두 가지로 구분한다. 하나는 스트림 암호화이며 또 하나는 블록 암호화이다. 스트림암호에서 XOR를 사용하게 된 것은 스트림 암호뿐만 아니라 ECB(Electronic CodeBook Mode)나 CBC(Cipher Block Chainning Mode) 와 같은 블록암호화 내부에서 적용되는 거의 모든 방법은 비트연산인 XOR을 이용한다는 것에 기반하고 있다는것에 근거한다[8]. 또한 블록암호화 방법론에서는 블록암호 방식의 대표적인 DES를 사용하였다.

첫번째로 스트림암호에서 사용한 방법인 XOR는 비트교환 방법으로써 가장 적은 연산으로 원하는 영역의 데이터의 변조를 가져올 수 있으며, 가장 빠르게 사용할 수 있는 데이터 변조 방식이다. 적용된 키값으로는 0x00~0xFF 까지의 지정된 패턴을 사용하여 위에서 언급되었던 허프만 테이블 영역에 XOR 연산을 적용하는 방법을 사용하였다.

두번째로 사용한 방법은 DES이며, DES는 블록암호로써 블록암호는 n 비트의 평문블럭을 l 비트의 매개변수를 이용하여 n 비트의 암호문 블록으로 사상시키는 함수이며, DES는 64비트의 키를 적용하여 64비트의 평문을 64비트의 암호문으로 암호화 시키는 대칭형 블록암호이다. 따라서 DES 의 경우에는 64비트 단위로 적용이 되어야 하기 때문에 실제적으로 허프만 테이블의 경우에 64비트로 나누었을 경우에 나머지 부분에 대해서는 패딩을 시키는 방법보다는 처리하지 않는 방법을 선택하였다. 그 이유는 패딩을 시키게 된다면 허프만 테이블에서 마커코드뒤에 나오는 길이값에도 변경을 가해야 하므로 수정을 최소화하기 위하여 나머지 부분에 대해서는 처리하지 않는 방법을 선택하였다.

본 논문에서는 기존에 사용하고 있지 않았던 마커코드인 "FF02"를 암호화 방법을 위한 마커코드로 사용하였으며, 각각의 암호화 방법론에 있어서 XOR와 DES의 경우에는 복호화 키값을 외부로부터 입력받아 파일을 복호화하므로 파일내부에 키값을 가지고 있을 필요가 없으므로 암호화 방식만을 저장하는 방법을 택하였다.

4. 실험 및 결론

JPEG 파일의 암호화 테스트를 위하여 우선 JPEG 파일을 확인할 수 있고 저장할 수 있는 프로그램을 작성하였다. 파일을 읽어 들인 후에 암호화를 수행하게하는 함수를 선택하여 암호화의 종류와 키값을 입력하게하는 루틴을 첨부하였다. 이때 사용된 암호화 방법론과 키값은 위에서 정의한 마커코드 "FF02"를 이용하여 파일에 추가적으로 저장을 하였으며, 파일에서 암호화를 해야할 영역을 검색하여 정의된 암호화 방법으로 내용을 암호화한 후에 저장하는 방법을 적용하였다.

영역에 따른 암호화를 비교하기 위하여 다음과 같이 두가지 영역으로 나누어서 테스트를 진행하였다. 첫번째는 허프만 테이블만을 암호화 하였으며, 두번째는 허프만 테이블과 이미지 저장 부분인 스캔 데이터까지 암호화를 적용했을 때로 나누어서 실험해 보았다.

실제 실험에서는 JPEG 파일을 복호화 되는 시간의 차이가 미미하므로 시간을 확인하기 위하여 각각 내부적으로 10회 반복적으로 읽어 들였으며, 이것을 5차례 반복하여 평균값을 계산해서 암호화 방법론에 따른 시간을 확인하는 방법을 택하였다. 아래 표 2와 표 3에서

표 2. 암호화별 복호화 시간(허프만)

File #	FileSize	Normal	XOR	DES
01	115,461	1.050	1.263	1.353
02	130,414	0.381	0.613	0.737
03	289,759	0.978	1.524	1.585
04	313,725	0.619	1.250	1.216
05	1,230,685	1.307	3.572	3.675

표 3. 암호화별 복호화 시간(허프만+스캔)

File #	FileSize	Normal	XOR	DES
01	115,461	1.050	1.503	1.538
02	130,414	0.381	0.878	0.959
03	289,759	0.978	2.056	2.131
04	313,725	0.619	1.772	1.822
05	1,230,685	1.307	5.947	6.072

시간의 단위는 초이다. 실험환경은 펜티엄4(1.6GHz)를 사용하였으며, 운영체제는 MS Windows2000 Server이었고 비주얼씨++ 6.0을 이용하여 테스트하였다.

표 2와 표 3에서 Normal 은 암호화가 안된 JPEG 파일을 읽어 들이는 시간을 의미하는 것이며, XOR은 XOR 방식으로 저장된 파일을 읽어 들이는 시간이고 DES는 DES 방식을 적용한 파일을 읽어 들이는 시간이다.

표 2에서는 허프만 테이블만을 암호화한 경우이며, 표 3은 허프만 테이블과 스캔 데이터 영역을 함께 암호화한 경우이다. 위의 표를 비교해보면 표 3의 시간이 전체적으로 표 2보다 수치가 높음을 알 수 있는데 이것은 암호화를 수행해야 하는 영역이 증가하면 상대적으로 시간이 증가한다는 것을 의미한다.

여기에서 허프만 테이블만 암호화 한 것과 허프만 테이블과 스캔데이터 영역을 함께 암호화 한 것의 보안성은 "3.1 암호화 영역"에서의 허프만 테이블이 쓰인 이유에서 처럼 이미 두영역 모두 코드화를 거쳤기 때문에 거의 같다고 볼 수 있다. 따라서 본 논문에서는 암호화 방법으로 허프만 테이블만을 암호화 하는 것을 제시하였다.

프로그램을 살펴보면, 우선 정상적인 출력의 경우에는 파일의 내용을 우선 버퍼로 옮겨 바로 출력루틴으로 보내서 처리하였으며, XOR와 DES 의 경우에는 중간에 허프만 헤더를 추출하는 함수를 만들어서 처리하는 과정을 한번 더 거치게 하였다. 그러므로 메모리를 엑세스하는 과정이 발생하여 파일의 크기가 증가하였을 경우에 복호화 하는데 조금 더 많은 시간이 걸린 것으로 나타났다. 즉 복호화 하는 시간이 아니라 메모리처리에서 시간이 소요된 것으로 추측할 수 있으며, 실제로 적용하게 될때에는 버퍼에서 처리하는 과정이 없이 복호화 루틴에서 처리하게 된다면 시스템에 거의 영향을 주지않고 처리가 가능하다.

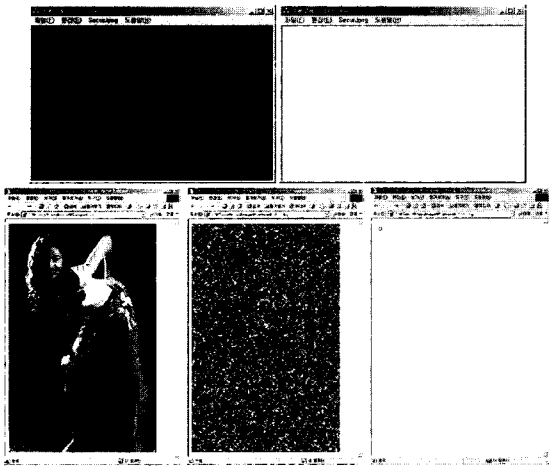


그림 6. 암호화된 JPEG 결과.

결과를 살펴보게 되면 아래의 그림들은 왼쪽은 정상적인 원래 그림이며, 오른쪽은 위와 같은 방법으로 암호화된 JPEG 파일을 일반적인 이미지뷰어 프로그램에서 봤을 경우에 나타나는 모습이다. 즉 일반적으로 아래의 그림처럼 표현을 못하거나 또는 그림이 깨진 형태로 표현하게 된다.

결론적으로 JPEG 파일의 암호화에 있어서 전체 데이터가 아닌 허프만 테이블이라는 최소한의 데이터를 수정함으로써 효과적으로 암호화를 수행할 수 있었으며, 서론에서 언급했던 도면이나 서류를 다루는 기업체 환경 또는 멀티미디어 서비스의 경우에는 접근이 허가된 사용자에게 별도의 이미지뷰어를 제공함으로써 암호화된 자료에 대한 접근을 보다 용이하게 제어할 수 있음을 확인하였다.

향후 고려해 볼만한 사항으로는 JPEG의 확장의 형태인 MPEG에서도 거의 유사한 방법으로 암호화를 수행할 수 있으며 접근이 허락된 사용자만이 볼 수 있는 VOD(Video on Demand)같은 서비스나 스트리밍 서비스를 제공하게 될 때 시스템에 영향을 최소화하면서 암호화를 할 수 있는 효과적인 방법이 될 수 있을 것이다.

참고문헌

[1] Bishop, M., "An Application of a Fast Data Encryption Standard Implementation", C~, Lputing Systems, 1, 3 (Summer 1988) 221-254.

[2] R. C. Gonzalez and R. E. Woods, "Digital Image Processing", Addison-Wesley Publishing Company, 1992.

[3] Huffman, "A Method for the Construction of Minimum Redundancy codes", Proc. IRE, Vol.40, pp. 1098-1101, 1952.

[4] U. S. National Bureau of Standards, Data Encryption Standard. Federal Information Processing Standard Publication 46, 1977.

[5] Netravali and Haskell, Digital Pictures: "Representation and Compression", Plenum Press, New York 1988.

[6] Pennebaker and Mitchell, JPEG: "Still Image Data Compression Standard", Van Nostrand Reinhold, New York 1993.

[7] Rabbani and Jones, "Digital Image Compression Techniques", Tutorial Texts in Optical Engineering, vol. TT7, SPIE Press, 1991.

[8] 이만영 외, "전자상거래 보안 기술", 생능출판사, 2001.

[9] 한국전자통신연구원, "암호학의 기초", 경문사, 1999.