디지탈 컨텐츠의 부분 암호화 기법
(Partial Encryption Technique of Digital Contents)

황 선 태 †
(Suntae Hwang)

요 약 급속도로 확대되고 있는 전자거래에 있어서 디지탈 정보의 보호가 점차 심각한 문제로 대두되고 있다. 따라서 효율적이고 편리하며 안전한 방법이 구축되어야 전자거래를 활성화시킬 수 있으리라 판단된다. 본 연구에서는 공개키 기반구조에서 스마트카드를 이용하여 디지탈 정보의 권한을 효율적으로 관리함으로서 정보를 보호하고자 한다. 이와 같은 목적을 달성하기 위해서 정보의 암복호화 시간을 단축하고, 서버의 부하를 감소시키는 부분 암호화 기법을 제시하며, 장차 디지탈 정보 서비스 업체에서 이 방법을 사용함으로서 상대적으로 강한 경쟁력을 갖출 수 있으리라 본다.
키워드 : 부분암호화, 디지탈 컨텐츠, 공개키

Abstract In the rapidly growing e-business area, the protection of information from hacking or tapping becomes very serious issue. Therefore, the more effective, convenient and secure methods are required to make the e-business more active. In this study, we develop the effective method of protecting digital contents on the public key infrastructure. To do this, we propose the partial encryption scheme to reduce the encryption time, and therefore, to release the server's workload. Our suggested scheme is believed to bring the strong competition to the portal service agents.
Key words : Partial Encryption, Digital Contents, Public Key

1. 서 론

가상의 세계를 대상으로 한 인터넷 거래는 그 편리성 때문에 이용률이 급격히 증가하고 있다. 공급자는 실제 상점을 운영하지 않기 때문에 상품 가격이 저렴하고, 고객은 편하게 주문 및 대금 지불을 할 수 있다는 이점이 있다. 이와 같은 상품에는 유형의 제품뿐만 아니라 무형의 정보도 포함된다. 상품은 배달 중에 손실될 경우 적법하게 즉시 배상이 될 수 있으나, 무형의 정보는 타인에 의한 불법 복사나 도용이 쉽게 발견되지 않는다. 또한 이와 같은 불법 행위는 그 정보의 제공자뿐만 아니라 이용자의 권한과 재산권을 침해하게 된다. 따라서 점차 확대되어 가는 전자상거래 상에서 오디오나 비디오 자료 같은 무형의 디지탈 컨텐츠(Digital Contents)를 적절히 보호하기 위한 효율적인 방안이 제시되어야 하리라 본다.

1974년 스마트카드의 개념이 등장한 이후 우수한 보안성과 편리성 때문에 스마트카드는 사회 전반에 걸쳐 그 사용 영역을 점차 확대해 나가고 있다. 최근에 국경을 넘어 인터넷이 발전하고 PC를 이용한 전자거래 및 각종 제품이 개발되면서 개인의 신분 증명과, 지불수단, 공개키 기반 구조를 위한 키의 보관용으로 스마트카드가 각광을 받고 있다[1]. 특히 암호 시스템에서 키의 중요성을 감안할 때, 키가 카드 외부로 노출이 안되므로, 고도의 안전성을 확보할 수 있다. 또한 인터넷 사용자 수의 증가와 함께 전자상거래(EC) 시장이 급속히 성장하면서 2003년에는 전자상거래 시장 규모가 2조 달러가 되리라 예측하고 있다. 이와 같은 전자상거래 시장에서는 전자제품이나 책 등과 같은 물건은 물론 관련 정보 및 오디오, 비디오 등과 같은 멀티미디어 컨텐츠도 함께 거래되고 있다. 향후 5년 내에 이와 같은 세계 멀티미디어 컨텐츠 시장 규모는 3조 달러에 육박할 것으로 보고 있다[2].

본 연구에서는 이와 같은 상황에 효과적으로 대처하기 위해 스마트카드를 이용한 공개키 방식의 접근과 더불어[3, 4, 5] 디지탈 컨텐츠 등의 전송 시 필수적인 부분 암호화 기법을 제안함으로서 효율성을 극대화 하고자 한다.

† 종신회원 : 대전대학교 컴퓨터정보통신공학부 교수
hwang@dju.ac.kr
논문접수 : 2001년 8월 23일
심사완료 : 2002년 1월 23일

2. 스마트카드 구조

1974년 Innovation S. A.의 Roland Moreno에 의해 처음으로 스마트카드에 대한 특허가 출원된 이후 최초의 스마트카드는 1977년 불(BULL)사와 모토롤라사의 합작으로 등장하게 되었다[6, 7]. 이 스마트카드는 메모리 부분과 컨트롤러 부분이 둘로 분리된 2칩(Chip) 구조였다. 그 후 3년 뒤인 1980년 모토롤라사에 의해서 SPOM01이라고 명명된 단일 칩 스마트카드가 최초로 개발되었다. 일반적으로 스마트카드는 8비트 마이크로 프로세서를 주로 사용하는데 가장 널리 사용되는 마이크로 프로세서는 모토롤라의 68HC05 및 인텔의 80C51 이다[8, 9].

일반적인 접촉식 스마트카드 형태는 얇은 플라스틱 카드에 8개의 접점을 가진 조그마한 반도체 칩을 장착하고 있으며, ISO 7816의 제안을 따르고 있다[8, 10]. CPU는 8비트부터 32비트 RISC 프로세서까지 개발되고 있고, 롬(ROM)은 주로 16/32K바이트 용량으로 불변의 데이터, COS 그리고 Look-up 테이블 등을 저장하는 반면, 램(RAM)은 1K바이트 미만으로 전압이 공급되는 동안만 CPU의 연산을 위해 이용된다. 또 EEPROM은 8/16K바이트로 전압이 공급되는 동안에는 데이터의 저장이나 변경 등이 가능하나 단전된 상태에서는 저장된 데이터는 그대로 유지된다. 이 EEPROM에 기억되는 내용으로는 카드 아이디 번호(Card Id Number), PIN, 잔액 그리고 크레딧 한계 등이다.

스마트카드를 운용하기 위해서는 내장된 IC 칩을 운영해주는 Chip Operating System(COS)이 필요하고, 그 외에 정보의 보안을 위한 인증 및 디지탈 서명과 사용하고자 하는 업무의 성격에 맞는 데이터 구조의 정의 및 프로그램 작성이 필요하다. COS는 마이크로 프로세서에 내장되어 있는 시스템 프로그램으로서 응용 프로그램의 H/W 접근을 가능하게 할 뿐만 아니라 스마트카드의 기본적인 기능을 결정한다. COS의 주된 기능은 카드와 단말기 사이의 데이터 송수신, 명령어 수행 제어, 데이터 관리 그리고 암호 알고리듬의 수행 등이다. 이와 같은 COS는 처음에는 개개의 응용 분야 별로 개발이 되었으나 점차 통합 환경에서 여러 종류의 응용 프로그램에 대해 수행이 가능하도록 설계되고 있으며, 또한 변경 시에도 최소한의 비용으로 단기간에 요건을 만족시키도록 구성되고 있다[6, 11]. 유럽의 이동 통신에 이용되고 있는 GSM 카드가 대표적인 한 예이다[12]. 따라서 우수한 스마트카드를 설계하기 위해서는 실리콘의 면적 및 공정기술, 비트 수, 동작 속도, 알고리즘, 신뢰성 있는 회로, 메모리 용량, 전력 소비율, 범용성 등 다양한 면을 검토하여야 한다[13, 14, 15, 16].

3. 효율적인 디지탈 정보 권한관리 방법

3.1 일반적인 암복호화 기법

디지탈 컨텐츠의 보호를 위한 일반적인 방법은 그 내용을 암호화하여 저장이나 전송한 후, 권한을 가진 자만이 해독하여 이용할 수 있도록 하는 것이다. 암호화 방법은 크게 대칭키 방식과 비대칭키 방식을 들 수 있는데, 대칭키의 대표적인 것으로 비밀키를 사용하는 DES 알고리즘 그리고 비대칭키 방식으로는 공개키와 개인키를 이용하는 RSA 알고리즘을 들 수 있다. 그러나 DES에 비해 상대적으로 속도가 느림에도 불구하고 키 관리 등의 편리성 때문에 공개키 방식을 선호하고 있다[4].

RSA 방식은 1976년 Diffie-Hellman에 의해 그 개념이 소개된 이후 1978년에 MIT의 Rivest, Shamir, Adleman에 의해 최초로 효율성과 보안성을 갖춘 공개키 암호 시스템으로 개발되었다. 이 방식은 평문과 암호문이 어떤 정수 n에 대하여 0 ~ (n-1) 사이의 정수인 블록 암호 구조이다. 즉 평문은 블록으로 암호화되고, 각 블록은 어떤 수 n보다 작은 이진 값을 갖는다. 암복호화는 평문 블록 M과 암호문 블록 C에 대해 $C = M^e \bmod n$, 그리고 $M = C^d \bmod n$ 의 형태를 따른다. 송수신자는 n의 값을 알아야하고, 또 송신자는 e의 값을 알고 d의 값은 수신자만이 알아야한다. 따라서 공개키는 {e, n}이고 개인키는 {d, n}이 된다[5].

이렇게 생성된 키들은 적절히 분배되고 관리되어야 한다. 이러한 목적의 인증기관은 안정성의 확보를 위하여 공개키를 공개적으로 사용 가능한 동적 디렉토리를 보유하게 되는데, 그 내용은 각 가입자에 대해 {이름, 공개키} 형식을 갖춘다. 뿐만 아니라 공개키 디렉토리는 어떤 형태로든 항상 공개가 가능해야 하며, 가입자는 전자적으로 디렉토리에 접근이 허용되어야 한다[17]. 이와 같은 공개키 암호화 방식은 전용 코프로세서를 탑재한 스마트카드 등을 이용하여 빠른 속도로 구현이 가능하며 안전성이 확보되어 있다. 최근 같은 공개키 방식인 ECC(Elliptic Curve Cryptosystem) 알고리듬이 점차 주목을 받고 있다. 특히 이 ECC 방식은 RSA에 비해 10% 내지 20% 정도 크기의 키로 RSA에 버금가는 강력한 보안성을 제공할 뿐만 아니라, 메모리 사용도 상당히 효율적이다. 또한 RSA 방식과는 달리 덧셈이 주된 연산이므로 수행 시간에 있어서 RSA 보다 10여 배 더 고속으로 수행이 가능하다[5].

이와 같이 단순한 전체 암복호화 시스템은 향후 수많은 디지탈 컨텐츠의 활용 시 서버나 클라이언트 시스템에 막대한 부하를 부담시킨다. 따라서 시스템과 사용자의 효율성 제고를 위해 보다 향상된 암복호화 기법이 요구되어 지며, 이는 현존하는 일부 암복호화 알고리듬을 이용하되 그 알고리듬을 효율적으로 사용할 수 있는 방법을 강구함으로서 그 목적을 이룰 수 있다.

3.2 부분 암복호화 기법

디지탈 정보에 대한 권한을 합법적으로 관리하기 위해서는 이 정보에 접근 권한을 가진 자만이 접근할 수 있도록 시스템을 구축하여야 한다. 최근 업계에서 개발한 시스템들은 대부분 사용자가 디지탈 파일의 복호화 및 재생 프로그램과 토큰(Token)을 전송 받아서 이를 처리할 수 있도록 하고 있다. 이에 우리는 공개키 기반 구조 상에서 스마트카드를 이용하여 키 관리를 함으로써 보안성을 높이고, 특히 정보의 암복호화 그리고 사용자의 인증 문제 등을 효율적으로 구현하는 방안을 제시하고자 한다. (그림 1)에서 전체 시스템의 개략을 파악할 수 있으며, 각 단계별 특성 및 구현 과정은 다음과 같다.

(1) 디지탈 컨텐츠(D. C.) 전송 신청 - 사용자가 원하는 디지탈 정보의 전송을 DCP 서버에 요구한다. 이때 사용자는 자기의 패스워드를 입력하여 MD5와 같이 안전성이 확보된 해쉬 함수를 이용해 해쉬 값으로 변경한 후 사용자 인증을 위해 전송한다. 인증 서버에 저장된 해당 패스워드의 해쉬값은 패스워드의 안전성을 보장한다.

(2) 키생성 및 사용자 인증 요구 - 디지탈 정보의 요청자가 새로운 사용자이면 인증 서버에 공개키와 개인키 생성을 요청하고, 등록된 고객이면 패스워드의 해쉬값을 이용하여 사용자 인증을 한다. 키 생성을 위해서는 임의의 네 숫자를 초기 치로 선정한 후 프로그램으로 변형하여 두개의 소수를 얻고, 이로부터 공개키와 개인키를 얻는다. 인증 서버에는 사용자의 키 생성 및 관련 알고리듬, 탐색 기능 등이 저장되어 있고, 인증 파일은 사용자 Id, 패스워드, 그리고 공개키가 각각 20, 16, 40 바이트로 구성되어 있다.

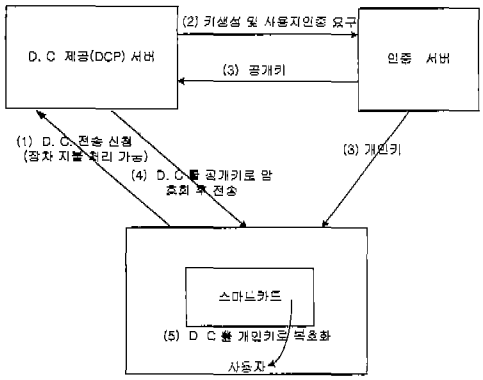

그림 1 Security Flow 개념도

(3) 공개키/개인키 - 인증기관은 요청자가 새로운 사용자로 판명되면 생성된 개인키를 사용자의 스마트카드에 전송하여 EEPROM에 저장하고 또한 생성된 공개키를 DCP 서버에 전송한다. 확인된 기 등록자이면 공개키를 DCP 서버에 전송한다.

(4) D. C.를 공개키로 암호화 후 전송 - DCP 서버는 요청된 디지탈 정보를 사용자의 공개키로 부분 암호화하여 사용자에게 전송한다. 이 부분 암호화 시스템은 안전성이 확보된 RSA 이론을 기본으로 하여 개발하며 다음과 같이 구현된다.

```
Var : Encr_rate, /* 3-byte 암호화율 % */
      Encr_blk_size; /* 4-byte 암호화블럭크기 Kbyte */
Begin /* 부분암호화 */
    Input Encr_rate, Encr_blk_size;
    Put Encr_rate + Encr_blk_size into Header;
    Write Header into Encr_file; /* 암호화 파일 */
    While (not eof)
        For Encr_rate
            Read Data_file using Encr_blk_size; /* D. C.
            파일 */
            Encrypt it; /* RSA 이용 */
            Append it to Encr_file;
        For (100 - Encr_rate)
            Read Data_file using Encr_blk_size & (100 -
            Encr_rate);
            Append it to Encr_file;
    End While
End. /* 부분암호화 */
```

(5) D. C.를 개인키로 복호화 - 사용자는 스마트카드에 저장된 개인키를 이용해 DCP 서버로부터 전송 되어온 부분 암호화된 D. C.를 복호화 하여 디스크에 저장하거나 재생한다. 이 복호화 시스템 역시 안전성이 확보된 RSA 이론을 기초로 하여 다음과 같이 개발한다.

```
Var : Encr_rate,
      Encr_blk_size;
Begin /* 복호화 */
    Read Header from Encr_file;
```

```
While (not eof)
    For Encr_rate
        Read Encr_file using Encr_blk_size;
        Decrypt it; /* RSA 이용 */
        Append it to Decr_file; /* 복호화 파일 */
    For (100 - Encr_rate)
        Read Encr_file using Encr_blk_size & (100 -
        Encr_rate);
        Append it to Decr_file;
End While
End. /* 복호화 */
```

이와 같은 일련의 과정에서 필요한 부분 암호화 정보는 부분 암호화된 파일의 헤더(Header) 부분에 7바이트 길이로 첨부된다. 이 7바이트는 3바이트의 암호화 율(Encryption_rate) 정보와 4바이트의 암호화 블럭 크기(Encryption_block_size)로 구성된다. 암호화 율은 0부터 100% 중에서 선택되어지며, 전체 정보 중 몇 퍼센트를 암호화 할 것인가를 결정하는 데 사용된다. 또 암호화 블럭 크기는 킬로바이트 단위로서 암호화될 단위 블럭 크기를 결정하게 된다. 이 단위는 추후에 변동이 가능하다. 예를 들어 20% 암호화 율에 10K바이트 암호화 블럭 크기를 선정하여 암호화 수행 시, 파일의 첫 10K바이트(20%)를 암호화하고 연속해서 80%인 40K바이트를 원문대로 복사하고, 다시 10K바이트를 암호화하고 이어서 40K바이트를 그대로 복사하는 과정을 파일 마지막 부분까지 계속한다. 이 7바이트의 효율적인 암호화 정보는 복호화 때 참조된다.

3.3 실험 및 평가

현재 스마트카드가 이용되고 있는 분야는 사회전반에 걸쳐서 증가하고 있는 추세이며 특히 전자거래 등에서 상호 교환되는 디지탈 정보의 보호에 커다란 역할을 할 수 있다. 디지탈 컨텐츠의 보호를 위한 일반적인 방법은 크게 대칭키 방식과 비대칭키 방식으로 나눌 수 있는데, 대칭키의 대표적인 것으로 비밀키를 사용하는 DES 알고리즘 그리고 비대칭키 방식으로는 공개키와 개인키를 이용하는 RSA 알고리즘을 들 수 있다. 그러나 DES에 비해 상대적으로 속도가 느림에도 불구하고 키 관리 등의 편리성 때문에 공개키 방식을 선호하고 있다. 특히 최근에 주목받는 ECC 방식은 RSA에 비해 10% 내지 20% 정도 크기의 키로 RSA에 버금가는 강력한 보안성을 제공한다. 그러나 이와 같은 알고리듬을 이용한 단순한 전체 암복호화 시스템은 향후 수많은 디지탈 컨텐츠의 활용 시 서버나 클라이언트 시스템에 막대한 부하를 부담시킨다. 따라서 시스템과 사용자의 효율성 제고를 위해 보다 향상된 암복호화 기법이 요구되어 지며, 이는 현존하는 일부 암복호화 알고리듬을 효율적으로 사용할 수 있는 방법을 강구함으로서 그 목적을 달성할 수 있다. 우리가 개발한 부분 암복호화 프로그램에서는 기존의 방식과는 달리 멀티미디어 컨텐츠의 일부분만을 암호화하기 때문에 서버 상에서 암호화 시간의 경감을 가져다준다. 특히 오디오나 비디오 관련 자료 등과 같은 대용량의 디지탈 컨텐츠는 이와 같은 부분 암호화 기법 적용 시 시스템 효율이 무척 높아진다. 스마트카드 관련 부분 암복호화 프로그램은 UNIX/Solaris 2.7 플랫폼 상에서 MS C 컴파일러 5.0을 사용해 구현하였으며, 인증프로그램은 백그라운드 작업으로 Daemon 상에서 항시 수행된다. 암복호화 시뮬레이션에서 필요한 입력 데이터는 3바이트의 암호화 율, 4바이트의 암호화 블럭 크기, 공개키와 개인키, 그리고 암호화 대상 파일이다. 또한 사용자 인증을 위해서는 패스워드가 인증파일 상의 ID 확인을 위해 필요하다.

이와 같은 환경에서 사용자 인증을 위한 목적으로 각 사용자의 ID, 패스워드의 해쉬코드, 공개키를 인증 서버에 저장해 놓는다. 사용자로부터 디지탈 정보의 전송 요청 시 DCP 서버는 인증 서버에 사용자 인증을 요청하고 확인된 사용자의 공개키는 DCP 서버에 우송된다. DCP 서버는 공개키를 이용하여 디지탈 정보를 부분 암호화한 후, 부분 암호화 정보와 함께 사용자에게 전송한다. 사용자는 스마트카드에 저장된 개인키와 전송된 암호화 정보를 이용하여 데이터를 복호화 한 후 이용한다. 이를 도식으로 나타내면 (그림 2)와 같다. 이 그림에서 나타나듯이, 사용자에 의한 D.C. 요청과 인증기관에 대한 인증요구에 의해 전체 시스템이 단계적으로 수행되어진다.

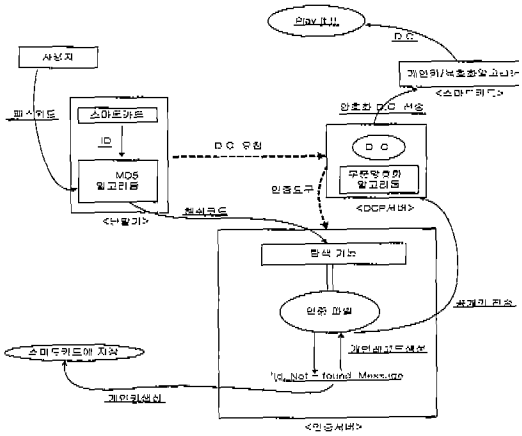

그림 2 사용자 인증 및 부분암호화 과정

이러한 일련의 부분 암복호화 과정에서 필요한 암호화 정보는 부분 암호화 파일의 헤더 부분에 3바이트의 암호화 율과 4바이트의 암호화 블럭 크기로 첨부되며, 이 정보는 복호화 때 참조된다. 다음 <표 1>, <표 2>, <표 3>에 부분 암복호화 시뮬레이션 결과의 일부가 나타나 있다. 이 도표는 파일 크기(File_size)가 1.725M바이트인 MP3 파일을 대상으로 각각 10%, 20%, 33%의 부분 암복호화 수행 결과를 보여주고 있으며, 사용된 공개키{e, n}와 개인키{d, n}의 크기는 각각 250비트{e/d}이다. 이 MP3 파일은 일반적으로 첫 부분에 싱크 헤더(Sync Header)인 12비트의 '1'을 포함한 4바이트의 제어정보를 가진다. 이 제어정보에는 버전, 레이어, 비트율, 카피권한 등이 포함되며, 그 다음에 오디오 데이터가 위치한다. 이 표에서 암호화/복호화 시간 값은 프로그램 구현상의 주관적인 문제로 인해 의미를 갖지 못하며, 단지 상대적인 비교 치로서 그 의미가 있다. 각 표는 암호화 블럭 크기, 암호화 율, 암호화 시간, 복호화 시간을 포함하며, 각각 상대적인 암호화 블럭 크기와 암호화 율 값을 가지고 결과를 비교할 수 있다. 여기서 암호화 블럭 크기가
너무 크거나 하면 각 표의 마지막 행들에서 나타나듯이 원하는 암호화 율을 얻을 수 없다. 이를 수식으로 설명하면 다음과 같다.

즉, {File_size Mod (Encryption_block_size ÷ Encryption_rate × 100)}의 값의 크기가 작거나 또는 (Encryption_block_size ÷ Encryption_rate × 100)의 값에 가까울수록 원하는 암호화 율이나 암호화 블럭 크기에 접근할 수 있다. 따라서 이 식은 원하는 값을 구하기 위한 상대적인 척도로 사용되어질 수 있다.

<표 1> ~ <표 3>에서 살펴보면 암호화 율이 증가할수록 암호화 및 복호화 시간이 함께 증가함을 알 수 있다. 또한 전체 암호화 기법과의 효율성 비교를 위해 파일 크기가 1.725M바이트인 MP3 파일 전체를 암호화 및 복호화 한 결과 각각 4시간 23분 정도의 시간이 걸리고, 따라서 암복호할 정보의 양과 소요 시간이 비례함을 알 수 있다. 이는 결국 암호화하는 정보의 양을 줄일수록 암복호화 시간을 절약할 수 있음을 나타내고 있다. 각 도표 내에서 암호화 블럭 크기가 증가할 때 암호화/복호화 시간이 약간씩 감소하는 경향은 암호화와 비 암호화 부분을 번갈아 처리하는 횟수가 감소하기 때문이며 전체 시간에는 별로 큰 영향을 끼치지 못한다. 또한 부분 암호화 기법은 정보의 중요도에 따라 암호화 율과 블럭 크기를 조절함으로서 적절한 정보보호와 암복호화 시간 단축의 이점을 동시에 제공한다. 특히 시뮬레이션 결과에서 보듯 동일한 암호화 율인 경우, 암호화 시에 가능한 한 암호화 블럭 크기가 작고 블럭 수가 많은 것이 블럭 크기가 크고 수가 적은 것보다 정보보호에 훨씬 유리하다. 그 이유는 블럭 크기가 큰 경우에 비 암호화 정보가 경우에 따라 쉽게 노출될 가능성이 있기 때문이다. MP3와 같은 음악인 경우 20% 내지 30%의 암호화 율로 정보의 권한 관리가 충분히 가능함을 증명할 수 있었으며, 데이터 양이 많은 비디오 자료와 같은 경우도 동일한 결과를 예측할 수 있었다.

표 1 10% 정도 암호화한 경우

암호화 블럭 크기	암호화 율	암호화 시간	복호화 시간
10 K바이트	10.44 %	27분 36초	28분 32초
18 K바이트	10.43 %	27분 34초	28분 29초
35 K바이트	10.15 %	26분 46초	27분 41초
58 K바이트	10.09 %	26분 37초	27분 32초
87 K바이트	10.09 %	26분 36초	27분 31초
173 K바이트	10.03 %	26분 26초	27분 23초
300 K바이트	17.39 %	45분 49초	47분 25초

표 2 20% 정도 암호화한 경우

암호화 블럭 크기	암호화 율	암호화 시간	복호화 시간
10 K바이트	20.29 %	53분 36초	55분 25초
35 K바이트	20.29 %	53분 34초	55분 13초
58 K바이트	20.18 %	53분 15초	55분 01초
87 K바이트	20.18 %	53분 10초	55분 00초
173 K바이트	20.06 %	52분 54초	54분 40초
345 K바이트	20.00 %	52분 42초	54분 32초
500 K바이트	28.99 %	76분 25초	79분 00초

표 3 33% 정도 암호화한 경우

암호화 블럭 크기	암호화 율	암호화 시간	복호화 시간
10 K바이트	33.63 %	88분 52초	92분 02초
29 K바이트	33.63 %	88분 46초	91분 59초
58 K바이트	33.63 %	88분 45초	91분 57초
115 K바이트	33.34 %	87분 57초	89분 58초
288 K바이트	33.40 %	88분 04초	91분 02초
575 K바이트	33.34 %	87분 51초	90분 52초
1000 K바이트	57.98 %	152분 59초	156분 07초

4. 결 론

현재 무한대로 확장되어 나가는 디지탈 멀티미디어 컨텐츠 산업의 보호를 위해 권한 관리 기능을 강화하여야 할 필요성이 제기되고 있다. 따라서 막대한 양의 정보를 효과적으로 관리하기 위하여 기존의 전체 암호화 기법에 비해 경제적이고 효율적인 부분 암호화 기법을 본 논문에 제시한다. 위의 실험에서 기존의 방법대로 파일 크기가 1.725M바이트인 MP3 파일 전체를 암호화 및 복호화 하는데 각각 4시간 23분 정도의 시간이 걸리며, 암복호할 정보의 양과 소요되는 시간이 비례함을 알 수 있다. 특히 시뮬레이션 결과에서 나타나듯이 암호화율이 감소할수록 암호화 및 복호화 시간이 함께 감소함을 알 수 있는데, 이는 결국 암호화하는 정보의 양을 줄일수록 암복호화 시간을 절약할 수 있음을 나타내고 있다. 또한 부분 암호화 기법은 정보의 중요도에 따라 암호화 율과 블럭 크기를 조절함으로서 적절한 정보보호와 암복호화 시간 단축의 이점을 동시에 제공한다. 특히 동일한 암호화 율인 경우, 암호화 시에 가능한 한 암호화 블럭 크기가 작고 블럭 수가 많은 것이 블럭 크기가 크고 수가 적은 것보다 정보보호에 훨씬 유리함을 알 수 있다. 데이터 양이 많은 비디오나 오디오의 경우 품질에 이용자가 매우 민감한 만큼, 20% 내지 30%의 암호화 율로 충분한 효과를 얻을 수 있었다. 따라서 공개키 기반의 스마트카드를 이용한 부분 암호화 기법은 암복호화 시간의 단축은 물론 불법 도청 등에 대한 권한 관리를 효율적으로 할 수 있으리라 기대된다.

참 고 문 헌

[1] R. Townend, 'Smart Card Technology International', 1st Ed., Chantry Hurst Books, 1998.

[2] 조정석, 최종욱, "전자상거래에서의 디지털 컨텐츠 저작권 보호를 위한 데이터 은닉과 디지털 워터마킹 기술", 정보처리학회지, 제6권 6호, pp. 93-104, 1999.

[3] 박희운, 이임영, "암호 기술", 정보처리학회지, 제7권 2호, pp. 7-19, 2000.

[4] 김철, '암호학의 이해', 초판, 영풍문고, 1996.

[5] 박창섭, '암호이론과 보안', 초판, 대영사, 1999.

[6] W. Rankl, W. Effing, 'Smart Card Handbook', 1st Ed., John Wiley & Sons, 1997.

[7] 황선태, 이형, "스마트카드 모델의 기준에 관한 연구", 한국전자 거래학회 연구논문, 제4권 제3호, pp. 197-212, 1999.

[8] C. H. Fancher, "In your pocket : smartcards" IEEE Spectrum, pp. 47-53, Feb. 1997.

[9] H. Dreifus, J. Monk, 'Smart Cards', 1st Ed., John Wiley & Sons, 1998.

[10] ISO, 'ISO/IEC 7816', 1997.

[11] M. Hendry, 'Smart Card Security and Applications', 1st Ed., Artech House, 1997.

[12] ETSI/PT12, 'GSM 11.11 : Specification of the SIM-ME Interface', 1991.

[13] Semiconductor and IC Div., 'H8/3102 Series Hardware Manual', 1st Ed., Hitachi Ltd., 1995.

[14] O. Kömmerling, M. Kuhn, "Design Pinciples for Tamper-Resistant Smartcard Processors", Proc. of USENIX Workshop on Smartcard Technology, pp. 1-12, May 1999.

[15] Fridrich J., "Applications of Data Hiding in Digital Images", Proc. of The 6th IEEE Int'l. Workshop on Intelligent Signal Processing and Communication Systems, pp. 24-30, Nov. 1998.

[16] 유병곤, 유종선, 이원재, 김보우, "강유전체 메모리 기술 현황 및 전망", 전자공학회지, 제25권 제7호, pp. 99-106, 1998.

[17] 김석우, 서창호, "전자상거래 인증서비스 기술", 정보처리학회지, 제7권 제2호, pp. 20-24, 2000.

황 선 태

1979년 서강대학교 수학과 이학사. 1979년 ~ 1982년 KIST 연구원. 1987년 Case Western Reserve University 전자계산학과 석사. 1988년 ~ 1989년 Cleveland State University 전자공학과 강사. 1993년 Case Western Reserve University 전자계산학과 박사. 1993년 ~ 1995년 현대전자 연구소 책임연구원. 1995년 ~ 현재 대전대학교 컴퓨터정보통신공학부 교수. 관심분야는 Smart Card, Security, Parallel Processing, VLSI Testing