DRM 보안 강화를 위한 키 교환 메커니즘 설계

임헌정*, 정성민*, 엄정호**, 정태명**
*성균관 대학교 전자전기컴퓨터 공학과
**성균관 대학교 정보통신공학부
e-mail : {hjlim99, smjung, jheom}@imtl.skku.ac.kr
tmchung@ece.skku.ac.kr

A Design of DRM System Key Exchange Mechanism

Hun-Jung Lim*, Sung-Min Jung*, Jeong-Ho Eom**, Tae-Myeong chung**
*Dept. of Computer Engineering, Sungkyunkwan University
**School of Information Communication Engineering, Sungkyunkwan Univ.

요 약

최근 전자 문서 및 콘텐츠 보호를 위하여 디지털 저작권 관리 시스템에 대한 연구 및 개발이 활발히 진행 되고 있다. 본 논문에서는 기존에 개발된 단순 메시지 암호화 단계의 디지털 저작권 관리 시스템 보안 기능상 문제점을 파악하고 보안 요구사항을 만족하기 위하여 마스터키 생성, 공개키 교환, 공유키 공유, 세션키 생성의 네 단계의 키 교환 단계를 설계하고 설계된 키 교환 메커니즘이 충족 시키는 보안 기능에 대하여 정리 하였다.

1. 서론

1950 년도부터 진행된 사무 자동화는 개인 PC 의 보급으로 급속히 자리잡게 되었다. 사무 자동화의 발전과 함께 최근 들어 전자문서결제시스템의 설치가 일반화 되고 있다. 전자문서결제시스템은 복잡한 문서 사무 업무를 간소화 시키고 업무 생산성을 증대시킬 수 있어 사회전반으로 활발히 사용되고 있다. 전자문서결제시스템의 특징 중 하나는 종이를 쓰지 않는 작업환경(paperless-office)이라는데 있다. 하지만, 모든 문서를 전자적 형태로 보관하기 때문에 자료의 복사 및 유출이 손쉽다는 취약점을 가지고 있다.

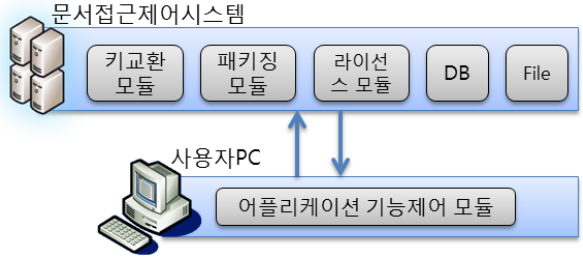
전자문서의 유출에 대한 해결책으로 디지털 저작권 관리(DRM:Digital Right Management)시스템에 대하여 많은 연구가 이루어지고 있다. DRM 은 디지털 콘텐츠의 불법 유통과 복제를 방지하고, 적법한 사용자만이 주어진 권한 내에서 콘텐츠를 사용케 하는 디지털 콘텐츠 저작권 관리 기술을 총칭하는 용어이다[1].

DRM 기술은 암호화 기술, 인증기술, 키관리/교환 기술, 패키징 기술, 권리표현 기술, 사용통제 기술, 탬퍼링 방지 기술 등을 이용하여 콘테츠를 보호 한다. 본 논문에서는 여러 DRM 보안 기술 중에서 키관리/교환 메커니즘에 대하여 디자인 하려 한다.

본 논문의 구성은 다음과 같다. 1 장의 서론에 이어 2 장에서는 전체 시스템의 구성 및 동작과정에 대하여 설명하였다. 3 장에서는 교환되는 키의 종류 및 특징에 대하여 설명하고 4 장에서는 각 키의 교환 메커니즘에 대하여 상세하게 기술 하였다. 5 장 결론에서는 설계된 방식으로 충족되는 보안 기능들에 대하여 정리 하였다.

2. 전체 시스템 구성도

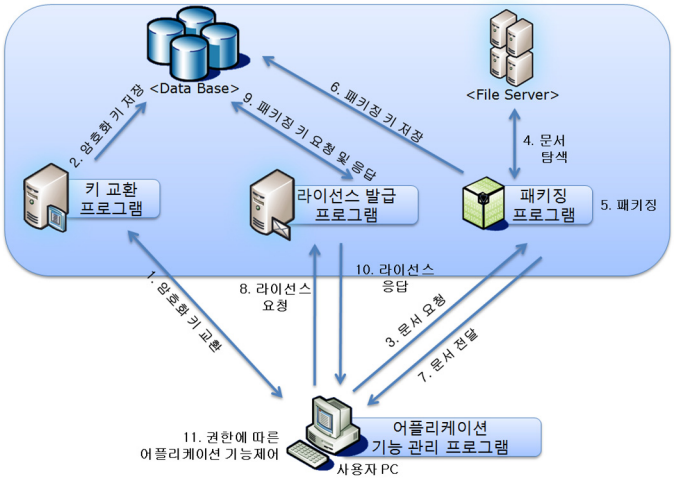
시스템 구성은 암호키 및 문서관리에 대한 역할을 수행하는 문서접근제어시스템과 사용자 PC 로 구성된다. 각 시스템이 수행하는 기능은 (그림 1)과 같다.



(그림 1) 시스템 구성 요소

- 문서접근제어시스템
  - ✓ 키 교환 모듈 : 공개키/개인키 교환 기능, 공유키 공유 기능
  - ✓ 패키징 모듈 : OMA-DRM 표준 준수, 문서 파일 암호화, 암호화 키를 데이터 베이스에 저장
  - ✓ 라이선스 발급 모듈 : 문서 열람을 위한 라이선스 파일 생성, 문서 복호화 및 권한 정보 저장
  - ✓ 데이터 베이스 : 키/관련 정보 저장
  - ✓ 파일 서버 : 문서 파일 저장
- 사용자 PC
  - ✓ 어플리케이션 기능 관리 모듈 : 문서 열람을 위한 키 교환 작업 수행, 문서 열람을 위한 라이선스 파일 요청, 권한에 따른 어플리케이션 기능 제어

시스템 별 동작 과정을 메시지 흐름 별로 표현하면 (그림 2)와 같다.



(그림 2) 시스템 동작 과정

1. 사용자 PC 는 초기 구동 시 문서보안을 위한 암호화 키를 문서접근제어시스템의 키 교환 프로그램과 교환 한다. 교환되는 키는 사용자의 공개키, 문서접근제어시스템의 공개키, 사용자와 문서접근제어시스템간의 공유키이다.
2. 문서접근제어시스템의 키 교환 프로그램은 사용자의 공개키, 공유키를 Data Base 에 저장한다.
3. 사용자는 열람하고자 하는 문서를 문서접근제어시스템의 패키징 프로그램에게 요청한다.
4. 문서접근제어시스템의 패키징 프로그램은 요청된 문서를 File Server 를 탐색하여 패키징 모듈에 전달한다.
5. 문서접근제어시스템의 패키징 프로그램은 요청 문서 암호화를 위해 패키징 키를 생성하여 암호화 한다.
6. 문서 암호화에 사용된 패키징 키는 Data Base 에 저장된다.
7. 패키징 키로 암호화된 문서는 OMA-DRM 표준에 따라 헤더 파일을 추가 하여 패키징되어 사용자에게 전달된다.
8. 사용자는 패키징된 문서 열람을 위하여 복호화키와 권한 정보가 포함된 라이선스를 문서접근제어시스템의 라이선스 발급 프로그램에게 요청한다.
9. 문서접근제어시스템의 라이선스 발급 프로그램은 해당 문서의 패키징 키와 사용자의권한 정보를 Data Base 에 요청하여 해당 정보를 획득한다.
10. 문서접근제어시스템의 라이선스 발급 프로그램은 수신된 패키징 키와 권한정보를 이용하여 라이선스를 생성하여 사용자에게 전달한다.
11. 사용자 PC 의 어플리케이션 기능 관리 프로그램은 수신된 라이선스의 패키징 키를 이용하여 문서를 복호화 하고 라이선스의 권한 정보를 이용하여 한글 2007 및 MS 워드프로세서 어플리케이션의 문서 열람기능을 제어 한다.

3. 키 교환 메커니즘

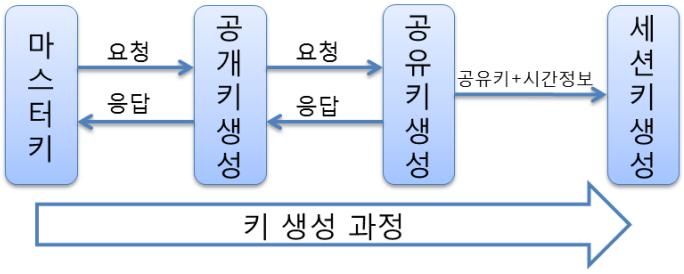
본 논문에서는 (그림 2)의 동작과정 중에서 문서의 패키징 및 메시지 송수신시 사용되는 암호화 키 교환 메커니즘에 대하여 디자인 하려 한다. 암호화를 위해서 사용되는 키는 마스터키, 공개키, 공유키, 세션 키로써 각 키의 생성 시기 및 용도는 <표 1>와 같다.

<표 1> 각 키의 생성시기, 생명주기, 용도

	마스터키	공개키	공유키	세션키
생성시기	사전운용 단계생성	사용자 PC 부팅시 초기 1 회 생성		매 메시지 송수신시
생명주기 (정책상)	제한 없음	1 년	6 개월	1 시간
용도	공개키 요청시	공유키 요청시	세션키 생성시	메시지 암호화시

- 마스터키: 시스템 사전 운용단계에서 별도의 등록 절차를 통하여 서버와 나누어 가지게 되는 키 값으로써 키의 생명주기에는 제한이 없으며 공개키 요청 및 나누어 가진 모든 키 값을 암호화 하는데 사용된다. 마스터키는 초기 로그인시 메모리에 상주 하면서 클라이언트에 저장된 키 복호화시 사용된다. 클라이언트 종료시 메모리에서 해지되도록 하여 추가 악용을 방지 하도록 설계한다.
- 공개키: 공개키/개인키 쌍으로써 초기 클라이언트 구동시 생성되며 개인키는 마스터키로 암호화하여 키 폴더에 저장하고 공개키는 서버로 전송하여 최초 1 회(1 년 주기) 등록한다. 공유키 요청을 위한 메시지 암호화 및 서명시 사용된다.
- 공유키: 클라이언트에서 생성한 임의 키값으로써 서버와의 공개키 공유절차가 종료된 후 생성되어 서버에게 전송된다. 공유키 생명주기는 6 개월로써 이후 새션키 생성시 활용된다.
- 세션키: 서버와 클라이언트간의 메시지 송수신시 사용되는 대칭키 암호화 방식의 키 값으로써 사전 공유된 공유키와 시간정보를 인자값으로 활용하여 생성하게 된다. 시간 정보는 년.월.일.시 값으로써 세션키의 생명주기 1 시간을 결정하는 요소로 작용한다.

마스터키는 사전에 정의되어 있으며, 공개키/개인키와 공유키는 서버-클라이언트간의 요청,응답으로 생성된다. 세션키는 별도의 작업 없이 사전 공유된 공유키와 시간정보를 이용하여 생성된다. 키 교환 단계를 도표로 표현하면 (그림 4)와 같다.



(그림 4) 키 교환 순서도

4. 키 교환 메커니즘 설계

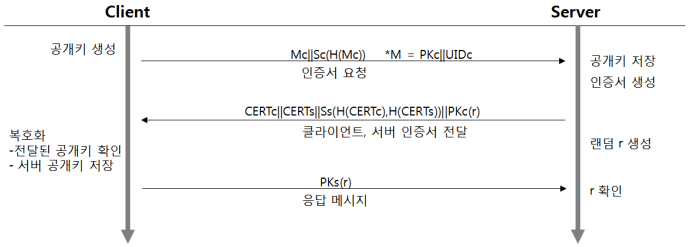
4.1 마스터키 생성

사용자가 별도의 등록과정 등을 이용하여 등록한 비밀번호를 활용하여 생성한다. 사용자가 입력된 비밀번호는 해쉬되어 서버에 저장되며 이 값을 마스터키로 활용한다. 서버에 저장된 마스터키는 사전에 등록된 비밀번호인지를 확인할 수는 있지만 해당 비밀번호가 무엇인지는 알 수 없다.

$$MasterKey = Hash(Password)$$

4.2 공개키 교환

공개키 공유는 초기 클라이언트 구동 시 자체 공개키/개인키 생성툴을 이용하여 생성하고 개인키는 마스터키로 암호화 하여 클라이언트의 키 폴더에 저장하고 공유키는 인증서 요청 메시지를 통하여 서버에 저장한다. 서버는 클라이언트의 공개키를 자신의 데이터 베이스에 저장하고 서버의 공개키를 클라이언트에게 전달 한다. 공개키 교환의 상세 동작 과정은 (그림 5)와 같다.



(그림 5) 공개키 교환

1) 각 클라이언트 c 는 공개키/개인키 생성 프로그램을 이용하여 공개키 PK 와 개인키 SK 를 얻는다. 각 클라이언트 c 는 안전한 영역에 개인키 PK 를 암호화 하여 저장한다.

2). 각 클라이언트 c 는 공개키 PK 와 UID 를 연접하여 메시지 M 을 생성한다.

$$M = PKc||UIDc$$

메시지 M 에 대하여 해시값을 생성하고 서명을 하여 서버에 전송한다.

$$Mc||Sc(H(Mc))$$

3) 서버는 클라이언트의 서명을 검증하고 각 클라이언트 별 인증서 CERTc 를 생성한 후 CERTc 를 DB 에 저장한다. 서버는 클라이언트의 인증서 CERTc 와 자신의 인증서 CERTs 와 이 값의 해쉬값을 생성하여 랜덤값 r 을 클라이언트의 공개키 PKc 로 암호화 하여 전송한다.

$$CERTc||CERTs||Ss(H(CERTc),H(CERTs))||PKc(r)$$

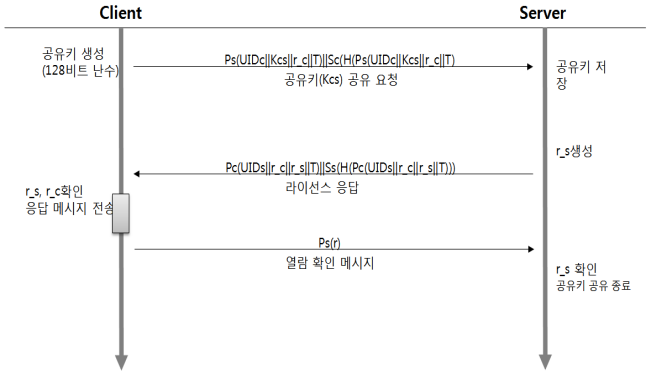
4) 클라이언트는 인증서를 검증하고 클라이언트에 CERTc 를 CERTs 와 함께 안전한 영역에 저장한다.

5) 클라이언트는 자신의 개인키 SKc 로 랜덤값 r 을 추출하여 서버의 공개키 PKs 로 암호화 하여 전송 한다.

$$Ps(r)$$

4.3 공유키 공유

공유키 공유는 서버의 공개키를 수신한 클라이언트가 128bit 의 난수를 생성하여 서버에게 전송하는 방식으로 진행된다. 전송되는 메시지는 사전에 공유된 공개키와 개인키를 이용하여 암호화 및 서명 후 전송된다.



(그림 6) 공유키 공유

1) 클라이언트는 서버와 공유할 공유키 Kcs 를 생성하기 위해 128bit 난수를 생성한다.

2) 자신의 UIDc 와 공유키 Kcs, 별도의 난수 r_c, 시간정보 T 를 연접하여 서버의 공개키로 암호화하고 암호화된 값의 해쉬를 개인키로 서명하여 전송한다.

$$Ps(UIDc||Kcs||r\_c||T)||Sc(H(Ps(UIDc||Kcs||r\_c||T)$$

3) 서버는 클라이언트의 서명을 검증하고 공유키 Kcs 를 저장 후 난수 r_s 를 생성하여 자신의 UIDs, 수신된 난수 r_c, 생성한 난수 r_s, 시간정보 T 를 연접하여 클라이언트의 공개키로 암호화하고 암호화된 값의 해쉬를 개인키로 서명하여 전송한다.

$$Pc(UIDs||r\_c||r\_s||T)||Ss(H(Pc(UIDs||r\_c||r\_s||T)))$$

4) 클라이언트는 r_c 를 확인하고, r_s 를 복호화 한다. 추출된 r_s 를 서버의 공개키로 암호화 하여 종료 메시지를 전송한다.

$$Ps(r\_s)$$

4.4 세션키 생성

DRM 시스템의 동작을 위한 모든 메시지 교환 및 문서 파일 암/복호화에는 세션 키를 활용한 대칭키 암호화 방식을 사용한다. 하나의 서버 시스템에는 다수의 클라이언트가 연결될 수 있으며 교환되는 메시지의 양도 방대하다. 매 메시지 교환 시 필요한 키 값을 서버와의 공유 작업을 통해 사용하게 되면 시스템 전체 동작 대기(Delay) 및 교환 작업을 위한 시스템 부하가 발생하게 된다. 따라서 세션키는 별도의 키 교환 작업 없이 사전에 공유된 공유키와 시간 정보를 이용하여 메시지 전송 전 단계에서 생성되어 사용 한다.

$$SessionKey = Hash(timestamp||SharedKey)$$

5. 결론

본 논문에서는 DRM 시스템의 보안성 강화를 위한 키교환 메커니즘을 디자인 하였다. 전체 동작을 위해서는 마스터키, 공개키, 공유키, 세션키의 4 종류의 키가 활용된다. 모든 메시지 송수신을 위해서 세션키를 사용하며 세션키 생성을 위해서 공유키가 필요하다. 공유키는 공개키를 이용하여 암호화 및 서명을 한다. 생성된 키들은 마스터키를 이용하여 암호화 하여 클라이언트에 저장되도록 하였다.

서로 다른 종류의 키 활용을 통하여 하나의 키 노출시 전체 시스템의 보안 노출을 최소화 시켰다. 디자인된 키 교환 방식은 키 값 및 메시지 송수신시의 무결성, 가용성, 부인방지 및 재전송 공격 방지에 효율적이었다.

참고문헌

[1] 박지현,정연정, 윤기송 ” DRM 기술동향”, ETRI, 2007.08
[2] William Stalling, “Cryptography and Network Security”, Pearson Education, 2006
[3] 이승제, “OMA 표준화 동향-OMA DRM”, TTA Journal, 2005.04
[4] 추연수. “DRM 시스템을 위한 안전한 복호화 키 분배 시스템 설계”, 한국컴퓨터종합학술대회, 2005
[5] Whitfield Diffie, Martin E. Hellman, “New Directions in Cryptography”, IEEE Transaction, 1976

표시	정의
s	서버
c	클라이언트
UIDs	객체 s 의 고유 아이디
CERTs	객체 s 의 인증서
PKs	객체 s 의 공개키
SKs	객체 s 의 개인키
Ps	객체 s 의 공개키 암호화 함수
Ss	객체 s 의 개인키로 서명 생성 함수
Ksc	객체 s 와 c 사이의 공유키
ksc	객체 s 와 c 사이의 세션키
H	암호학적 해쉬 함수
c	세션키 생성시 사용되는 카운터
r_s	객체 s 가 생성한 랜덤 값
T	타임 스탬프