문서 DRM의 보안성 강화를 위한 연구

김승환*, 엄정호**, 박선호*, 정태명**
*성균관대학교 전자전기컴퓨터공학과
**성균관대학교 정보통신공학부
e-mail:{shkim, jheom, shpark}@imtl.skku.ac.kr, tmchung@ece.skku.ac.kr

Seung-Hwan Kim*, Jung-Ho Eom**, Seon-Ho Park*,
and Tae-Myoung Chung**
*Dept of Electrical and Computer Engineering, Sungkyunkwan University
**School of Information and Communication Engineering, Sungkyunkwan

요 약

현재 제공되는 문서 DRM은 권한 그룹 또는 사용자에 따른 접근 제어를 제공하고 있다. 단순히 권한 그룹 또는 사용자 기반으로 문서를 제어 한다는 것은 문서를 보호하기 위해서는 많은 위험이 따른다. 오브젝트를 과하게 보호한다면 유연성이 떨어지고, 유연성을 높이고자 하면 보안강도가 떨어지게 되므로 보안강도와 유연성은 트레이드오프 관계에 있다. 따라서 본 논문에서는 유연성을 높이면서 문서의 보안강도 또한 높일 수 있는 방법에 대해 제시하고자 한다. 본 논문에서는 각 문서의 security level에 따라 문서 열기 권한을 다르게 적용함으로써 비밀문서의 보안강도와 유연성을 높일 수 있는 방법에 대해 제안한다.

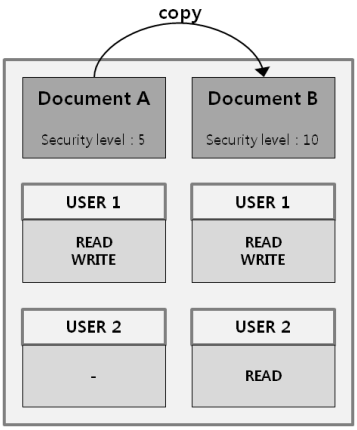
1. 서론

IT 기술의 발달로 문서가 디지털화 되어감에 따라 관리, 보관 등이 쉬워졌다. 또한 저장매체, 네트워크도 함께 발달함으로써 필요에 따라 유통이 쉬워진 반면 중요 문서, 아이디어, 소스코드 등의 유출도 증가하고 있다[1]. 이에 따라 문서를 보호하기 위한 기술이 개발되기 시작하였으며, 그 중 문서 DRM(Digital Right Management)이란 기술이 가장 널리 사용되고 있다.

문서 DRM이란 문서에 대한 저작권자의 권리나 이익을 보호하거나, 중요한 문서가 유출되었을 경우 이에 대해 보호하기 위한 기술이다. 각종 문서뿐만 아니라 다양한 디지털 매체에도 많이 적용되고 있는 기술이다. 문서 DRM을 통해 문서의 열람, 출력, 캡처 등 다양하게 권한을 부여하여 제어할 수 있다[1].

DRM을 사용하기 위해서는 각 사용자마다 권한이 부여되어야 한다. 하지만 사용자 마다 권한을 부여하는 것은 소규모 시스템의 환경에서는 문제가 되지 않는다. 하지만 대규모 사용자 환경에서는 많은 사용자로 인해 관리상의 어려움이 존재해 이에 따른 취약점이 나타날 수 있다. 따라서 본 논문에서는 직책간의 계층구조, 많은 사용자 등으로 인해 발생하는 복잡성은 역할기반 접근제어를 사용함으로써 보다 높은 보안성과 관리가 쉬워질 수 있다.

그리고 컴퓨터에서 다양한 작업을 동시에 처리할 수 있어 이를 이용한 취약점이 발생할 수 있다. 이 같은 취약점은 그림 1에서 설명하고 있다.



(그림 1) 발생 가능한 문제점

(그림 1)의 동작은 A문서와 B문서를 동시에 읽을 때 발생하는 동작이다. 권한 변경 전의 동작에서는 사용자가 A문서와 B문서 모두에 READ와 WRITE권한을 가지고 있기 때문에 security level이 높은 문서 A의 내용을 문서 B로 옮길 수 있다. 따라서 security level이 10이하만 읽을 수 있는 사용자의 경우도 이와 같은 과정을 통해 security level이 5인 문서의 내용을 볼 수 있는 문제가 있다.

이처럼 DRM시스템은 Object, 권한, 사용자 간에서의 적절한 접근 제어 정책이 필요하다. 또한 대규모 시스템 환경에서 적절히 object, 권한, 사용자를 관리할 수 있어야 한다. 따라서 본 논문에서는 역할 기반 접근제어를 DRM 시스템에 적용함으로써 보다 안전하고 쉽게 비밀문서를 보호하고자 한다. 또한 문서 자체에도 security level을 부여하여 문서가 가진 security level에 따라 권한을 조정하

여 문서가 의도치 않게 복사되는 것을 방지하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 DRM이 동작하는데 있어 필요한 요구사항에 대해 살펴본다. 그리고 3장에서는 본 논문에서 제안하는 시스템의 구조에 대해 살펴보고, 4장에서는 앞서 제시된 요구사항에 따라 제안하는 모델을 검증한 후 5장에서 결론을 맺는다.

2. DRM동작 요구사항

DRM이 안전하게 동작하기 본 논문에서 고려한 요구사항은 다음과 같다.

- **보안성**

콘텐츠를 안전하게 보호할 수 있어야 한다.

- **유연성**

정책이 적용된 규칙 내에서 사용상 불편함이 존재하지 않아야 한다.

- **용이성**

접근제어 정책을 관리하는 것은 용이해야 한다. 또한 DRM 시스템을 사용하는 방법이 쉬워야 한다.

- **불법적인 동작 불가**

콘텐츠를 불법 복제 및 불법 유통하는 것은 불가능해야 한다. DRM을 우회하여 콘텐츠에 접근하는 것이 불가능해야 한다.

- **안전한 키 관리**

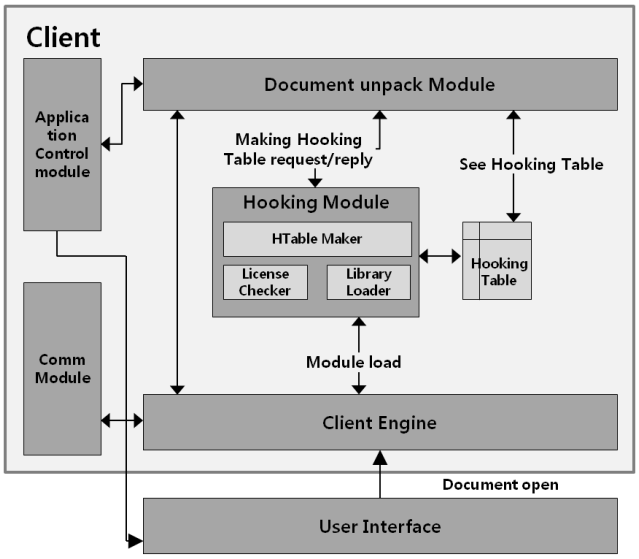
DRM에서 사용되는 모든 키는 안전하게 관리될 수 있어야 한다.

3. 제안하는 시스템의 구조

이번 장에서는 본 논문에서 제안하는 시스템에 대해 살펴본다.

3.1 클라이언트

(그림 2)는 클라이언트의 전체 구조에 대해 보여주고 있다.



(그림 2) 클라이언트의 구조

클라이언트는 사용자가 DRM 시스템을 사용할 수 있도록 하며, 이를 이용하여 각종 application을 제어하고, 서버와 통신을 하는 등의 역할을 한다. 각 모듈의 역할은 다음과 같다.

3.1.1 Client Engine

Client Engine은 Client가 정상적으로 동작하기 위해 모든 모듈, 라이브러리, 사용자의 정보를 관리한다. 또한 사용자 ID에 따라 권한을 요청하며, 사용자가 열고자 하는 문서에 따라 security level을 요청한다. Communication module과 직접적인 통신을 한다.

3.1.2 Hooking Module

기존에 작성된 문서를 열기, 수정, 인쇄 등을 하기 위해서는 열고자 하는 문서가 제작된 프로그램을 사용하여야 한다. 따라서 문서 DRM에서는 API Hooking 기술의 사용이 불가피하다. Hooking Module은 API Hooking Table을 작성해준다.

- **Library Loader**

Access Control Module로부터 권한 승인을 받았을 경우에 Client Engine으로부터 호출된다. Library Loader는 동작에 필요한 Library의 유무를 검사하고 존재할 경우 해당 Library를 로드한다.

- **HTable Making Module**

HTable Making Module은 Hooking Table을 만드는 모듈이다. Library Loader에 의해 로드된 Library에 포함된 정보를 테이블을 다른 모듈에서 참조할 수 있도록 작성한다.

- **License Checker**

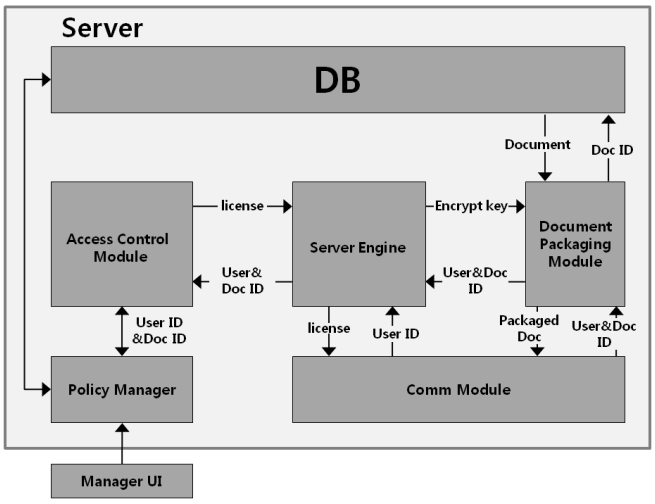
해당 모듈은 서버로부터 전달받은 License가 유효한 License인지 검증하는 모듈이다. 유효한 License일 경우 Library Loader를 호출하여 준다.

3.1.3 Application Control Module

서버로부터 수신한 license를 기반으로 문서를 열 때 어떤 권한으로 열어야 하며, 해당 문서를 열지 못 할 경우에 대한 이유를 사용자에게 알려준다. 가령 사용자가 높은 등급의 비밀문서를 열람하고자 할 경우 등급의 문제로 인해 열람할 수 없음을 알려준다. 이 외에도 서버로부터 수신되는 license에 따라 Application을 제어하여 사용자가 DRM을 우회하여 불법 유통, 복제를 할 수 없도록 한다.

3.2 서버

본 논문에서 제안하는 서버의 구조에 대해서 설명한다. 서버는 (그림 3)과 같이 구성되어 있다. 주 역할은 문서와 사용자의 권한간의 연관성, 문서의 security level간의 연관성에 대해 확인하여 license를 발급하는 역할을 한다.



(그림 3) 서버의 구조

3.2.1 Server Engine

Server Engine은 Server에서 수행하는 핵심적인 역할을 담당한다. License 발급, 키 관리를 주로 진행하며, 오류가 발생한 경우 오류처리를 한다. Server Engine의 구성 요소는 다음과 같다.

- **License Bank**

지정된 권한에 따라 license를 발급한다. License Bank는 Key Manager로부터 암호화에 사용되는 키 정보를 받아 license를 발급한다. 발급되는 license는 문서를 열 때 사용된다.

- **Key Manager**

DRM에서 사용되는 키를 관리한다. 키는 사용자가 사용하는 키, 문서를 암호화하는데 사용하는 키 등이 있다. 또한 키가 만료되었을 경우 재 발급하며, 필요한 경우에 키를 생성한다. 사용자가 삭제되는 경우 기존에 있던 키를 삭제하기도 한다. 키에 대한 정보를 얻거나, 키를 얻기 위해서는 Key Manager를 이용하여서만 가능하며, 다른 예외 상황에 의해 키를 얻어오는 것은 불가능하다.

3.2.2 Document Packaging Module

문서를 전송해주어야 하는 경우 문서의 암호화를 수행한다. 암호화를 수행할 때 필요한 key는 Server Engine에 요청하여 받을 수 있으며, 해당 문서는 DB에서 수신한다. 따라서 전송하는 문서를 암호화하는데 사용된 키를 DB에 저장한다.

3.2.3 Policy Manager

Policy Manager는 DRM을 사용하기 위한 보안 정책의 구성과 보안 정책의 제공 기능을 수행한다. Policy Manager의 보안 정책의 특징은 각 문서에 할당된 security level을 적용한 문서 정보 관리이다. Policy Manager는 이와 같은 보안 정책 구성과 관리 기능을 제공한다.

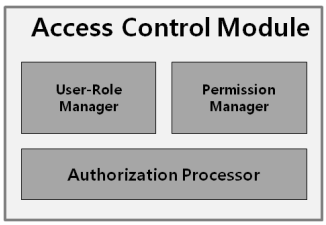
관리자가 설정하는 보안 정책은 Policy Manager에서 보안 정책 특징에 맞게 구성하여 DB에 저장한다.

3.2.4 Manager UI

Security Admin UI를 통해 현재 적용되고 있는 정책의 상태를 확인 할 수 있으며, 보안 정책이 변경될 경우에도 Security Admin UI를 이용하여 쉽게 변경이 가능하다.

3.2.5 Access Control Module

Access Control Module은 접근제어를 수행하는 모듈이다. (그림 4)는 Access Control Module을 보여준다.



(그림 4)Access Control Module

- **User-Role Manager**

사용자-역할 활성화를 담당하는 모듈이다. 사용자 정보를 이용하여 사용자 - 역할 활성화를 수행한다.

- **Permission Manager**

객체 정보와 오퍼레이션 정보를 이용하여 권한 정보를 생성하는 모듈이다.

- **Authorization Processor**

접근 제어를 수행하는 모듈이다. Policy Manager를 통해 정책 규칙 정보를 검색하고 접근 허가 여부를 결정한다.

4. DRM 요구사항 평가

DRM 요구사항 평가는 앞서 3장에서 제시한 요구사항을 본 논문에서 제안하는 시스템의 만족 여부에 관한 평가이다.

- **보안성**

보안성이 요구되는 문서의 경우 암호화를 적용하며, 암호화에 사용된 키는 Key Manager에 의해 안전하게 관리된다. 또한 문서 등록과정에서 사용자의 실수로 인해 적절하지 못한 권한으로 등록된 문서의 경우 관리자에게 알려줌으로써 문서가 유출되기 전에 수정이 가능하다.

- **유연성**

본 논문에서 제안하는 시스템은 특정 경우(관리자에 의해 보호되는 문서)에만 문서를 동시에 열 수 없도록 통제하고 있다. 이외의 경우 문서 등급(security level)에 따라 사용자의 역할 권한에 대한 operation을 조정하여 함께 열람이 가능하다. 따라서 강제로 보지 못 하게 하는 방법에 비해 유연성이 높다고 할 수 있다.

- ***용이성***

본 논문에서 제안하는 시스템은 역할에 기반하여 사용자를 할당하기 때문에 모든 사용자를 통제하지 않고, 존재하는 역할들만 관리하면 된다.

또한 사용자의 입장에서는 DRM시스템을 사용하는데 추가적인 작업이 필요하지 않는다.

- **불법적인 동작 불가**

기존에 제시된 DRM 시스템은 허용된 권한을 기반으로 각종 문서를 손쉽게 복사하여 붙여 넣을 수 있었다. 하지만 본 논문에서 제안하는 DRM은 허용된 권한이더라도 각 security level에 따라 허용된 권한에 따른 operation을 변경함으로써 이를 불가능하게 하였다.

- **안전한 키 관리**

키를 저장하는 DB가 따로 존재하며, 패키징, 세션키와 같이 자주 갱신되는 키와 자주 갱신되지 않는 키를 따로 보관한다. 또한 보관되는 키는 별도로 암호화를 하여 보관한다. 이 키 DB에 접근할 수 있는 것은 Key Manager만 접근할 수 있어, 발생 가능한 취약점을 최대한 줄였다.

5. 결론 및 향후 연구 계획

제공 중인 DRM은 두 가지의 문서제작 어플리케이션을 실행하여 문서 내용을 옮겨 적음으로써 동작 중인 DRM을 우회하여 문서를 유출할 수 있었다.

하지만 본 연구에서 제안한 DRM은 권한을 조정하고, 문서 제작 어플리케이션의 실행을 제어할 수 있다. 이와 같이 본 논문에서는 보다 안전하게 문서를 보호할 수 있는 DRM의 구조에 대해 제안하였다. 그리고 발생가능한 위협에 대해 제안하는 DRM에서 보호 할 수 있음을 확인하였다. 본 연구를 통해 내부자 유출로 인한 피해를 한층 더 줄일 수 있을 것으로 예상한다. 또한 모듈이 추가될 것을 충분히 고려하여 구조를 설계하였다. 따라서 향후 연구하는데 있어 쉽게 추가될 수 있을 것으로 예상된다.

향후 연구로는 본 논문에서 제안하는 시스템에 적용 가능한 문서 포맷을 구성하고 구현할 것이다. 또한 문서 포맷을 구성함으로써 현재 DRM에서 제공되지 않는 상호 연관성을 함께 해결할 수 있는 연구를 할 것이다. 그리고 DRM에 더욱 적합한 접근제어 정책에 대해 연구할 계획이다.

참고문헌

[1] Bill Rosenblatt, Bill Trippe, Stephen Mooney, "Digital Rights Management, Business and Technology.", Published by M&T Books, pp79-102.

[2] Pierangela Samarati and Sabrina de Capitani di Vimercati, “Access Control: Policies, Models, and Mechanisms”, FOSAD 2000, LNCS 2171, pp.137-196, 2001.

[3] Rivi S. Sandhu, Edward J. Conyne, Hal L. Feinstein and Charles E. Youman, “Role-Based Access Control Models”, IEEE Computer Vol. 29, Number 2, pp. 38-47, February 1996.