

소프트웨어 소스코드의 저작권 표현과 접근제한을 위한 비즈니스 모델

차병래¹⁾, 김경준*, 이동섭**, 김철원
호남대학교 컴퓨터공학과
*호남대학교 전파이동통신공학과
**호남대학교 정보통신공학과
e-mail:chabr@honam.ac.kr

Business Models for Access Control and Copyrights Expression of Software Source Code

Byung-Rae Cha, Kyung-Jun Kim*, Dong-Seob Lee**, Chul-Won Kim
Dept of Computer Eng., Honam University
**Dept of Radio Mobile Communication Eng., Honam University
**Dept of Information & Communication Eng., Honam University

요 약

소프트웨어 소스코드는 크게 오픈 소스와 상업용 비공개 소스로 구분될 수 있다. 소스코드는 자동차 조립라인과 같이 소프트웨어를 무제한 생산이 가능한 지식집약적인 부가가치 산업이다. 본 논문에서는 오픈-소스 코드에는 디지털라이선스를 이용한 저작권을 표현하고, 비공개-소스 코드는 보안 컨테이너 역할을 수행하는 크립텍스 모델을 제안한다.

1. 서론

소프트웨어는 마이크로프로세서와 하드웨어를 작동하게 하는 소스코드로 이루어져 있으며, 실질적으로 컴퓨터가 복잡한 비선형 방정식 계산을, MP3 플레이어가 음악을 재생하도록 한다. 또한, 시스템 소프트웨어는 애플리케이션이 동작하기 위해 필요한 여러 서비스도 제공한다. PDA에 필기체를 인식시킬 수 있고 컴퓨터에서 회사의 인사관리 시스템을 작동시킬 수 있는 것도 바로 소프트웨어 덕분이다. 그러나 이런 놀랄 만한 기능을 하는 소프트웨어 엔진(Software Engine)은 우리 눈에는 보이지 않는다.[1] 이 엔진은 소프트웨어 엔지니어들에 의해 인간의 언어와 비슷한 특별한 언어로 작성되고, 마이크로프로세서가 이해할 수 있는 0과 1로 이루어진 기계어로 번역되는 과정을 거친다. 이러한 보이지 않는 소프트웨어는 엄청난 부가가치를 넣고 있으며, 빌 게이츠를 전 세계에서 가장 부유한 사람으로 만들었고, 리눅스의 초기 버전을 개발한 리누스 토발즈는 오늘날 많은 소프트웨어 개발자들의 우상이 되었다. 소프트웨어는 비록 언어를 기반으로 하지만 기술(technology)이며, 오히려 다양한 목적으로 다양한 산업에 이용될 수 있다. 소프트웨어 역시 개발자들에 의해 설계, 소스코드 작성, 디버깅 과정을 거친다. 또한 종종 어려운 문제를 풀거나 복잡한 수학적인 알고리즘을 작성해야 하는 경우도 있다. 일단 개발이 완료되면, 소프트웨어는 책

보다 훨씬 저렴하게 복제되고 공급될 수 있으며, 따라서 개발 비용을 회수할 정도로 충분히 판매되면, 현금 지급이나 다름없다. 그러나 책이나, 음악, 영화처럼 소프트웨어 역시 불법 복제되기 쉽다. 그래서 대부분의 소프트웨어 회사들이 개발 내용이 담겨 있는 소스코드는 제외하고 사용자들이 판독할 수 없는 기계어만 유통시키고, 저작권법이나 특허법을 통해 자신들의 지적 재산을 보호하고 있다.

2. 관련 연구

소프트웨어 소스코드의 패턴 매칭에 대한 연구는 Rieger[2]과 Yang[3]에 의해서 연구되어졌다. Rieger는 소프트웨어의 소스코드가 중복되거나 복제된 부분을 시각화하여 탐지하는 연구를 수행하였고, Yang은 두 개의 다른 프로그램을 구문론적으로 동일한 코드를 찾는 연구를 수행하였다. 프로그램 표절 검출 S/W로는 국외는 SIM[4], Dup[5], Plague, YAP[6], YAP3, MOSS[7] 등이 있다. 국내는 KAIST의 clonechecker와 부산대의 LOFC가 있으며 [8], 프로그램 심의위원회의 exEyesLight[9]가 있다.

DRM(Digital Rights Management)은 디지털 콘텐츠의 보호를 위한 암호화 및 사용자 인증키 관리, 디지털 콘텐츠 유통 환경을 구성하는 주체들 간의 지적재산권 및 거래 규칙, 디지털 콘텐츠의 이용 및 분배, 사용 및 접근제어 등을 가능하도록 하는 디지털 저작권 관리에 대한 전반적인 기술이다[10-12]. 최근에는 파수닷컴[13]에서 소스코드의 보안을 위해 DRM ONE 패키지를 개발하여 시판하고 있다. 이 패키지의 DRM ONE for CODE라는 소스코드

1) 이 논문은 2007년도 정부재원(교육인적자원부 학술연구조성사업비)으로 한국학술진흥재단의 지원을 받아 연구되었음 (KRF-2007-331-D00763).

보안 솔루션은 소프트웨어 소스 자체에 대한 보안 취약점을 악용한 해킹 및 유출사고를 예방하기 위해 프로그램 소스코드를 중점적으로 보호하는 솔루션이다.

3. 소프트웨어 소스코드의 문제점

책이나 음악, 연극, 특허 등과 마찬가지로 소프트웨어 역시 정보화 제품(Information Goods) 중 하나이며, 소프트웨어는 일련의 명령어들에 의해 직간접적으로 컴퓨터 하드웨어가 동작하도록 한다. 다른 정보화 제품들과 마찬가지로, 소프트웨어는 네 가지 주요한 경제적 특징을 갖고 있다. 소프트웨어는 인간 두뇌의 지적 산물이며, 여러 기호로 이루어져 있다. 또한 한 사람의 사용이 다른 사람들의 사용에 전혀 영향을 미치지 않고, 재생산에는 거의 비용이 들지 않는다. 다른 정보화 제품처럼, 소프트웨어 역시 지적 재산권 보호 없이는 돈을 벌기 어려운 게 사실이다. 또한 소프트웨어 개발은 고정비용이 아주 높고 한계비용은 아주 낮은 극단적인 경제 현상을 발견할 수 있다.

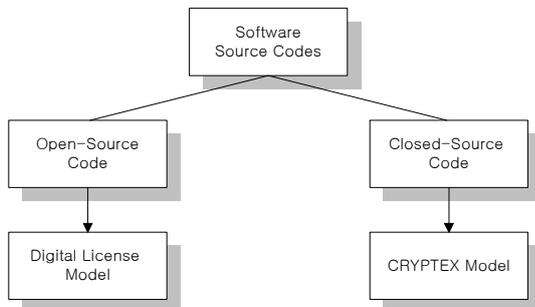


그림 1. 소스코드의 비즈니스 모델 제안

모든 정보화 제품들은 쉽게 복제될 수 있다. 특히 디지털로 되어 있는 소프트웨어는 더더욱 그렇다. 소프트웨어를 판매하는 회사에서 불법 복제는 가장 큰 골칫거리이다. 소프트웨어만 판매하는 회사들은 다른 제조업체들이 자사 소프트웨어를 설치할 수 있도록 라이선스를 판매한다.

다른 대부분의 재화나 서비스와는 달리, 소프트웨어는 모든 정보화 제품처럼 한 사람의 소비가 다른 사람들을 위해 가용될 양을 감소시키지 않는다. 오히려 한 사람이 더 이용함으로써 그 소프트웨어에 대한 가치가 증가될 수 있다. 일단 누군가 어떤 소프트웨어 제품의 소스코드를 얻게 되면 아무런 대가도 지불하지 않고 똑같은 소프트웨어 제품을 재생산할 수 있다. 이렇게 되면 시장 가격은 거의 0에 가까워질 것이고, 원개발자는 금전적 보상을 기대할 수 없게 된다. 소프트웨어 회사들은 이런 최악의 상황을 미연에 방지하기 위해 다음의 세 가지 주요 지적 재산권 보호 시스템에 의존하고 있다. 우선, 가능한 한 소스코드를 비밀로 유지하려 한다. 소프트웨어는 실제로 유통되기 전에 1과 0으로 이루어진 기계어로 변환되어 있다. 이론적으로는 이 분야의 전문가들이 기계어를 다시 사람이 이해할 수 있는 고급 언어로 변환할 수 있지만, 이런 형태의 역공

학(reverse engineering)은 거의 대부분의 상업적 소프트웨어 사용 계약서에 의해 금지되어 있고, 실질적으로 수백만 줄이 기계어를 변환하는 것은 거의 불가능하다. 게다가 영업 비밀에 관한 법률은 어떠한 개인 혹은 단체라도 승인없이 소스코드를 무단으로 배포하는 것을 금지하고 있다. 또한 소스코드를 저작권으로 보호한다. 책의 경우와 마찬가지로 소프트웨어 프로그램을 저작권자의 동의 없이 복제하면 저작권법에 저촉된다. 그럼에도 불구하고 다른 정보화 제품들처럼 불법 복제는 특히 지적 재산권이 취약한 나라들에서는 만연해 있다.

이러한 소프트웨어 소스코드의 지적재산권 문제를 해결하기 위해서 본 논문에서는 오픈소스와 비공개 소스코드에 대한 비즈니스 모델을 그림 1과 같이 제안한다.

4. 오픈-소스 코드의 저작권 표현을 위한 디지털 라이선스 모델

오픈소스 커뮤니티는 아주 독특한 방식으로 소프트웨어 개발을 수행한다. 우선 소프트웨어는 한 명 혹은 몇몇 개인에 의해 구상된 소프트웨어의 설계가 일반인에 공개됨을 시작으로, 직접 작성한 초기 버전의 소스코드도 함께 인터넷상에 공개된다. 이를 기반으로 인터넷상에 여러 개발자들이 자발적으로 참여하는 커뮤니티가 구성되고, 그 커뮤니티에서 여러 개발자들이 초기 버전의 프로그램을 기반으로 소스코드를 추가하며 프로그램을 완성해 나간다. 그리고 디버깅과 테스트는 커뮤니티 회원뿐만 아니라 일반인에게도 공개되어 누구나 참여할 수 있는 형태로 이루어진다. 이렇게 개발된 소프트웨어는 특정한 단체에서 관리되며 업그레이드 되어간다. 그러나 오픈소스 지지자들은 법적인 문제에 직면하게 되었으며, 한편으로는 오픈소스가 더 많이 사용되기를 원한다. 이는 곧 오픈소스 프로그램의 확산을 제한할 수 있는 저작권이나 특허, 혹은 영업 비밀 등의 지적 재산권을 사용하지 않겠다는 의미와 동시에, 상업적 목적으로 사용되는 것을 원치 않는다. 즉, 그들은 오픈소스 소프트웨어의 지적 재산권 확보를 원하는 것이다. GNU의 GPL(General Public License)는 그러한 딜레마를 해결할 아주 정교한 장치이며, 카피레프트(CopyLeft)라는 개념에 기반을 두고 있다. 이는 저작권을 의미하는 카피라이트(CopyRight)와 반대되는 개념으로 저작권의 공유를 뜻한다. 저작권법은 GPL로 소프트웨어를 배포하는 사람들이 그 소프트웨어를 이용하는 다른 사람들에게 여러 조건을 강제할 수 있는 법적 토대가 된다[1].

4.1 오픈 소스코드의 디지털 라이선스 모델

소프트웨어 개발자 A, B, C와 법원 D의 소프트웨어 불법 유출에 의한 문제 사항을 그림 2와 같이 제기한다. 소스코드의 원본의 소유자 A와 불법 유출자 B에 의해서 분쟁은 발생한다(그림 2의 ①). 법원은 이 문제를 해결하기 위해서 개발자 A와 불법 유출자 B의 소스 코드 제출을 요구한다(그림 2의 ②). 그러나 소프트웨어의 소스코드

는 전문가가 감별해야 하기 때문에 법원 D는 소프트웨어 전문 개발자 C에게 소프트웨어 소스코드의 원본과 불법 유출본과의 차이점을 의뢰하게 된다(그림 2의 ③). 소프트웨어 전문 개발자 C는 원본과 불법 유출본을 비교하기 위해서 소스코드를 이해해야 한다. C에 의해서 원본과 불법 유출본을 감별하는 것은 전적으로 C의 주관성 및 소스코드의 이해 능력에 전적으로 의지하게 되어 객관성이 부족하게 된다(그림 2의 ④). 여기서 제 2의 소스코드 불법유출 문제가 발생할 수도 있게 된다(그림 2의 ⑤). 의도하지 않은 소스 코드의 지적재산권이 침해되는 문제가 발생하게 된다. 즉 소스코드의 원본과 불법 유출본을 감별하기 위해서 C가 소스코드를 이해하다보니 자동적으로 C에게 제 2의 소스코드가 불법 유출되는 문제가 발생하며, 제 2의 분쟁 소지를 갖게 된다.

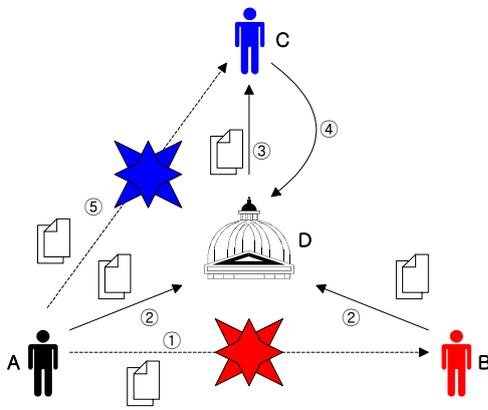


그림 2. 소프트웨어 분쟁에 의한 지적재산권 침해 문제 발생의 다이어그램

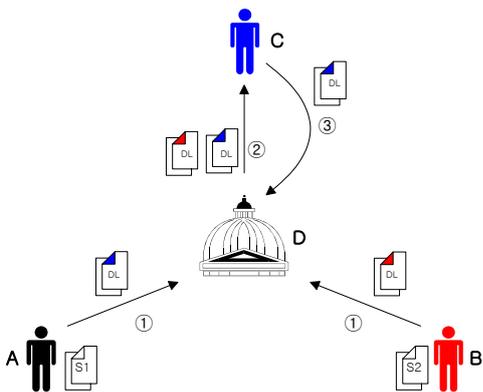


그림 3. 디지털 라이선스에 의한 지적 재산권 침해 방지

그림 2의 저작권 침해 발생 문제는 그림 3과 같이 디지털 라이선스를 적용함으로써 해결할 수 있다. 저작권 침해 문제의 발생을 소프트웨어 소스코드의 디지털 라이선스에 의해서 제 2의 분쟁을 사전에 제거할 수 있다. A와 B의 분쟁 발생시 소스코드를 법원에 제출하지 않고 A와 B가 갖고 있는 소스코드의 디지털 라이선스를 생성하여 법원에 제출하면 된다(그림 3의 ①). 법원은 패턴 매칭 프로그램

에 의해서 어느 정도 일치하는지를 객관적인 보고서를 얻을 수 있으며, 추가적으로 C에게 자문을 구함으로써 분쟁 해결에 많은 도움이 될 것이다(그림 3의 ②). 디지털 라이선스를 이용하면 또한 제 2 분쟁의 불씨를 제거하게 된다. C는 단지 소스코드가 아닌 디지털 라이선스를 검증하여 유출된 소프트웨어 소스코드가 원본에 어느 정도 일치하는 지에 대해 자문하면 된다(그림 3의 ③).

소프트웨어 소스코드의 디지털 라이선스에 의한 비즈니스 모델은 개발자 A의 지적재산권을 보호하고, 소스코드를 유출한 B를 좀 더 쉽게 판별하게 해주며, 또한 소스코드의 판별에 의한 제 2의 소스코드 유출을 사전에 막아서 제 2의 분쟁을 발생시키지 않는다. 더불어 C의 주관성에 의지하지 않고 객관적인 패턴 매칭 프로그램으로 확인한 보고서를 받을 수 있으며, C에 의해서 오랜 분석에 의한 자문에 크게 영향을 받지 않고, 짧은 시간의 패턴 매칭 프로그램 분석으로 분쟁 해결의 시간도 지체하지 않을 것이다.

4.2 디지털 라이선스의 기능

소프트웨어 소스코드의 디지털 라이선스를 구성하는 인덱스 패턴, 소프트웨어 아키텍처, 노드의 패턴 정보 그리고 메시지 축약과 암호화 등의 다양한 이러한 패턴 정보는 소프트웨어의 소스코드를 분류하거나, 클러스터링 그리고 검색할 수 있는 정보를 제공한다. 더 나아가서는 소프트웨어 버전 관리를 위한 정보도 제공할 수 있다. 소프트웨어 소스코드의 분류 및 검색 단계는 인덱스 패턴 정보에 의한 분류, 소프트웨어 아키텍처 패턴에 의한 분류 그리고 노드의 패턴 정보에 의한 분류로 구성된다. 또한 인덱스 패턴 정보에 의해서 소스코드의 예약어의 분포 및 패턴 정보를 확인할 수 있으며, 웹상에서 검색 및 색인 기능의 지원도 가능하다.

5. 상업용 비공개-소스 코드의 접근제어를 위한 크립텍스 모델

오픈-소스 코드의 디지털라이선스는 저작권과 소프트웨어의 아키텍처, 색인정보, 노드 정보를 이용하여 저작권 정보를 표시하는 수동적인 개체이며, 능동적인 수행능력을 갖지 않는다. 이를 지원하기 위한 적극적인 보호 기술은 비공개-소스 코드를 보호하기 위한 제반 기술의 지원과 프레임워크는 너무나도 빈약한 상태이다. 이를 해결하기 위한 하나의 방법으로서 먼저, 그림 4와 같은 크립텍스의 비즈니스 모델을 제안한다.

크립텍스(CRYPTEX)는 인증 및 인증되지 않은 주체로부터 객체인 소프트웨어의 소스코드를 안전하게 보호 및 접근제어를 지원하기 위한 제반 기술들의 집합을 통칭하는 보안 컨테이너 모델이다. 크립텍스 비즈니스 모델의 시스템 구성은 주체인 개발자, 객체는 비공개 소스코드를 포함하는 보안 컨테이너인 크립텍스 그리고 인증기관이 된다. 크립텍스의 비즈니스 모델은 크게 인증 단계와 접근제어

단계로 나눌 수 있다. 인증 단계는 인증기관, 지적재산권의 소유자, 크립텍스의 세 가지로 구성된다. 접근제어 단계의 구성은 인증서, 지적재산권의 소유자, 그리고 크립텍스이다. 크립텍스는 크게 비공개 소스코드와 이를 접근제어하는 알고리즘으로 구성된다.

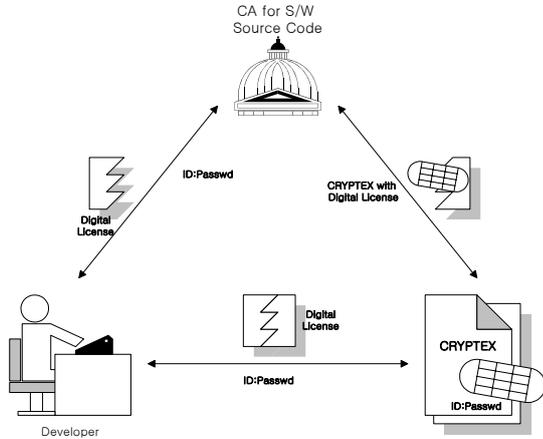


그림 4. 크립텍스의 비즈니스 모델

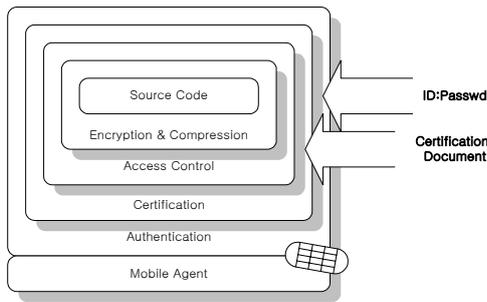


그림 5. 크립텍스 모델

크립텍스를 이용하여 단지 수동적인 문서 상태의 비공개 소스코드에 대해서 능동적이며, 접근제어 및 보호가 가능하며, 이동 및 위임기능을 부여할 수 있는 보안 컨테이너인 크립텍스 모델을 그림 5와 같이 제안한다.

크립텍스의 핵심 기능은 비공개 소스코드의 관리와 이를 접근제어하기 위한 알고리즘으로 구현된 이동 에이전트 (Mobile Agent) 소프트웨어이다. 소스코드는 코드 자체가 아스키코드 또는 유니코드로 이루어졌으며, 프로그래밍 언어로 소프트웨어가 수행할 작업들이 기술되어 있는 수동적인 문서 상태이다. 능동적인 일을 수행할 수 있는 프로세스 상태가 아니기 때문에 이를 보완하기 위한 기능을 갖는 보안 컨테이너인 크립텍스에 포함시켜서 이동, 관리 및 접근제어를 수행하게 된다. 크립텍스 자체는 의미가 없으며, 소스코드를 관리하기 위해서 능동적인 관리시스템 (캡슐 기능)의 역할 기능 수행과 네트워크를 통한 이동 기능을 갖는다. 관리 기능으로는 소스코드의 캡슐 기능, 인증서 관리 기능, 압축 기능, 디지털라이선스 기능, 사용자 인증 기능, 접근제어의 매칭 및 권한부여 기능, 그리고 네트워크를 통한 이동 기능을 갖는다. 인증 단계의 완료후의

사용자와 소스코드에 대한 접근제어, 접근 레벨에 의한 소스코드의 접근 제어 기능을 수행하게 된다. 접근제어 기능으로는 접근 정책과 레벨, 위임, 인증서의 시한 기능, 암호 기능 등을 수행한다.

6. 향후 연구 및 결론

소프트웨어는 제 3의 산업혁명 뒤에 있는 보이지 않는 엔진이 되고 있으며, 마이크로프로세서 기술에 기반을 둔 경제를 지탱하는 데 결정적인 역할과 인터넷에 근간을 두고 있는 산업에서도 중요한 역할을 수행하고 있다. 지식집약적이며 부가가치의 원천 요소인 소프트웨어 소스 코드에 대한 DRM이나 보안기술은 암호화이외에는 별다른 방법이 존재하지 않으며, 아직은 연구 초기 단계이다. 제안된 연구 내용으로는 오픈-소스 코드의 디지털 라이선스에 의한 저작권 표현과 비공개-소스 코드를 보호하기 위한 보안 컨테이너 역할을 수행할 크립텍스 모델을 제안하였다. 향후 연구로는 디지털 라이선스는 수동적인 저작권 관리를 위한 도구라면, 크립텍스는 소프트웨어 소스코드를 보호하기 위한 능동적인 도구가 될 것이다. 소스코드를 보호하기 위한 크립텍스를 구현하기 위한 연구가 필요하다.

참고문헌

- [1] David S. Evans, Richard Schmalensee, Andrei Hagiu, "Invisible Engines : How Software Platforms Drive Innovation And Transform Industries", MIT Press, Oct. 2006.
- [2] R. Matthias and D. Stephane, "Visual Detection of Duplicated Code," Proceedings ECOOP Workshop on Experiences in Object-Oriented Re-Engineering, 1988.
- [3] Y. Wu, "Identifying Syntactic Differences Between Two Programs," Software-Practice and Experience, Vol.21, No.7, pp.739-755. July, 1991.
- [4] <http://www.few.vu.nl/~dick/sim.html>
- [5] <http://glimpse.arizona.edu/javadup.html>
- [6] <http://www.ccsr.cam.ac.uk/~mw263/YAP.html>
- [7] <http://www.cs.berkeley.edu/~aikem/moss.html>
- [8] 조동욱, "디지털 재산권 보호를 위한 S/W 프로그램 품질 감정 기술과 틀의 분석", 한국콘텐츠학회 2003 춘계 종합학술대회 논문집, p.177-184, 2003.
- [9] <http://www.pdmc.or.kr/>
- [10] Intel, "Content Protection in the Digital Home," Vol.6, Issue 4, Intel Technology Journal, Nov. 2002.
- [11] R. Michael, "Utilizing Content Protection Technologies," Intel Developer Forum, Sep. 2002.
- [12] M. E. Ahmet, "Multimedia Protection in Digital Networks," CNIS 2003, 2003.
- [13] <http://www.fasoo.com>
- [14] 차병래, 특허 등록 : 10-0740222, "소프트웨어 소스 코드의 저작권 관리방법", 2007년 7월.