
프로그램 소스 코드의 원본을 공개하지 않는 소유권 증명을 위한 디지털 라이선스 설계

Design of Digital License for Ownerships Certification While Preserving the
Original Program Source Code

차병래, 김형중, 이동섭
호남대학교

Byung-Rae Cha(chabr@honam.ac.kr), Hyung-Jong Kim(joeun777@honam.ac.kr),
Dong-Seob Lee(dslee@honam.ac.kr)

요약

지적재산권 제도는 과거 산업사회의 태동과 발전과정에서 뿐만 아니라, 21세기 정보화 사회의 발전에 있어서도 중요한 역할을 하고 있다. 우리나라의 국가 경쟁력 제고를 위해서도 디지털콘텐츠에서 확대하여 프로그램 소스 코드에 대한 지적재산권 제도와 기술의 정비는 매우 중요한 의미를 지닌다. 프로그램 소스 코드의 소유권 분쟁이 발생시 소유권을 증명하기 위해서는 원본의 프로그램 소스코드를 공개해야만 하는 문제점을 갖고 있다. 본 논문에서는 프로그램 소스코드의 원본 공개로 인한 저작권 침해와 기술 유출을 막기 위한 프로그램 소스코드의 디지털 라이선스 방식을 설계 및 시뮬레이션한다.

■ 중심어 : | 디지털 라이선스 | 프로그램 소스 코드 | 소유권 증명 |

Abstract

The intellectual property system was very important to the past industrial society. It is so important to the 21C information age. It is a leading role to developing these information society. Not only the digital content control but the technology of program source code for the intellectual property is so much mean to international competition. On occurring disputation property, we have to prove the fact, there is a problem to show the original source code. In this paper, we simulate the digital licence for program source code to prevent the infringement of copyright and the tipping-off information.

■ keyword : | Digital License | Program Source Code | Ownerships Certification |

I. 서론

전통적으로 저작권은 주로 법적인 관점에서 다루어지고 연구되어 왔다. 저작권의 본질, 그것의 보호의 범위,

강제와 침해 등이 일반적인 연구의 대상이었다. 우선, 저작권은 여러 가지 창조적인 저작물 시장을 위한 안전하고 안정적인 환경을 제공하는 법적 시스템이다. 오늘날 비즈니스의 영역에서 라이선싱, 투자, 거래 및 이전

의 문제와 관련하여 저작권에 대한 관심이 크게 증대되고 있지만, 이것도 근본적으로는 저작권이 영향력 있는 법적 시스템이라는 속성에서 비롯되는 것이다. 최근 들어서 저작권의 경제적 영향에 대한 학계의 관심과 저작권의 국민경제적 역할에 대한 계량적 측정 등에 대한 정책 담당자들의 관심이 세계적으로 증대되는 것은 일상적인 경제생활의 생산, 유통, 소비의 전반에 걸쳐 지속적으로 확대되어 온 저작권의 영향력이 이러한 수준을 넘어섰음을 의미한다.

이러한 저작권의 경제적 가치의 증대에는 여러 가지 이유가 있다. 첫째, 저작권에 대한 관심은 부분적으로는 비물질적인 생산요소가 보다 주목을 받는 지식기반사회에서 증가하고 있는 지적재산(Intellectual Property : IP)의 역할에 대한 인식이 증대된 결과이다. 저작권은 창의성과 정보에 기반을 둔 산업의 성장과 산업의 생산성, 고용 및 투자를 위하여 중요하다. 둘째는, 디지털기술의 발달로 인하여 저작권보호의 대상물의 범위가 크게 증대되었다는 것이다. 소프트웨어, 멀티미디어 및 기타 기술기반 산물로부터 경제적 이득은 막대해지고 있다. 셋째는 디지털 혁명의 결과, 저작권 보호대상들이 전자상거래의 주 요소가 되었다는 점이다.

과거 산업사회의 태동과 발전과정에서 뿐만 아니라, 21세기 정보화 사회의 발전에 있어서도 이러한 지적재산권 제도는 중요한 역할을 하고 있다. 우리나라의 국가 경쟁력 제고를 위해서도 디지털콘텐츠에서 확대된 프로그램 소스 코드에 대한 지적재산권 제도와 기술의 정비는 매우 중요한 의미를 지닌다.

디지털 콘텐츠의 저작권 보호를 위한 연구에 비해서 프로그램 소프트웨어의 저작권을 관리하기 위한 기술은 아직도 초기 연구단계에 있다. 프로그램 소스 코드의 소유권 분쟁이 발생 시 소유권을 증명하기 위해서는 원본의 프로그램 소스 코드를 공개해야만 하는 문제점을 갖고 있다. 본 논문에서는 프로그램 소스 코드의 원본 공개로 인한 저작권 침해와 기술 유출을 막기 위한 프로그램 소스 코드의 디지털 라이선스 방식의 프로토타입을 제안한다.

II. 관련 연구

DRM 기술은 여러 목적으로 제작된 디지털 콘텐츠가 제작·유통·사용되는 과정에서 해당 콘텐츠 저작권이 보호될 수 있도록 관리해 주는 시스템이다. DRM을 이루는 기술은 크게 두 가지로, 저작권 보호 기술과 저작권 관리 기술로 구분지을 수 있다. 대부분의 상업화된 DRM 기술은 저작권 보호기술이 주를 이루고 있다. 프로그램 소스 코드에 대한 DRM 기술은 부재하며, 저작권 보호기술로는 1:1 관계의 거래에서만 임의의 효력을 갖은 암호화 기술뿐이며, 초기연구단계에 머물러 있다.

프로그램 소스 코드는 제품을 생산하는 공장의 설계면을 뛰어넘어서 제품생산라인과 같은 아주 중요한 디지털 콘텐츠 자원이다. 그러나 정작 이 자원들을 지원하는 DRM 기술은 너무나도 부족한 상태이다. 관련 연구는 프로그램 소스 코드의 저작권을 표현하는 디지털 라이선스의 프로토타입을 생성하기 위해 필요한 제반기술에 대해 기술한다.

1. DRM

DRM(Digital Rights Management)은 디지털 콘텐츠의 보호를 위한 암호화 및 사용자 인증기 관리, 디지털 콘텐츠 유통 환경을 구성하는 주체들 간의 지적재산권 및 거래 규칙, 과금·이용규칙에 대한 표현, 디지털 콘텐츠의 이용 및 분배, 사용 및 접근제어, 권한제어, 워터마킹, 불법복제 축적기술, 투명한 전자상거래를 위한 거래 내역의 관리 및 보고, 과금 처리 등을 가능하도록 하는 디지털 저작권 관리에 대한 전반적인 기술이다. 즉, 콘텐츠의 암호화와 권한제어를 통해 콘텐츠에 권한을 가지고 있는 사용자들만이 그 권한에 따라 콘텐츠를 열어 볼 수 있도록 함으로써 콘텐츠의 불법사용 방지, 자동과금, 결제대행, 부가서비스 등 디지털 콘텐츠의 생성, 유통, 사용에 관련된 전 분야의 서비스를 제공하는 것이다[1-4]. 본 연구는 프로그램의 소스 코드에 대해서 DRM을 지원하기 위한 기초 연구이다, DRM의 저작권 보호 기술보다는 저작권 관리 기술을 제안하며, DRM의 영역을 프로그램의 소스 코드까지 확장한다.

2. DOM

DOM(Document Object Model)은 HTML 문서 구조를 정의하는 모델을 언급할 때 사용한 것이다. W3C[5]는 HTML 뿐만 아니라 XML에 대해서도 상호간에 동작할 수 있는 공통적인 문서 객체 모델을 만들 목적으로 DOM Working Group 이라는 워킹 그룹을 만들었다. DOM은 객체들의 표준 집합을 통하여 응용 프로그램들에 대한 플랫폼 중립적이며 언어 중립적인 인터페이스를 정의한다. DOM의 처리과정은 XML 문서를 파싱하는 단계와 DOM 트리를 접근하는 단계로 구성된다. DOM에서는 XML 문서가 하나의 트리로 표현되며, 전체 문서는 일종의 중첩된 트리로서 표현된다[6]. 본 연구에서는 프로그램 소스 코드의 디지털 라이선스를 바이너리 파일이 아닌 XML 파일로 기술 및 생성하며, XML 문서를 DOM 기술로 처리할 수 있는 장점을 갖는다.

3. 파싱(Parsing)

프로그램의 소스 코드를 분석하기 위해서는 패턴 매칭과 파싱(parsing) 기술이 필요하다. 프로그램 소스 코드를 파싱하여 의미를 부여할 수 있는 구분된 컴포넌트를 토큰(token)이라 한다. 그리고 토큰은 각각 독립적인 유닛(unit)을 나타낸다. 프로그램 소스 코드가 갖는 추상적인 표현을 토큰화(tokenizing)를 거쳐야 하며, 구성된 토큰을 이용하여 디지털 라이선스의 정보 및 아키텍처를 구성하게 된다. 토큰은 두 가지 형식을 갖게 되는데, 외부 형식(external format)과 내부 형식(internal format)이다. 외부 형식은 프로그램 소스 코드를 작성할 때 사용되는 텍스트 형식이다. 예를 들어, PRINT라는 예약어는 외부 형식이다. PRINT를 1로 재정의하는 것이 내부 표현이다. 외부 형식을 효율적인 처리를 위해서는 내부 형식으로 전환한다. 프로그램의 디지털 라이선스의 정보 표현을 생성할 때 내부 형식으로 표현하면 문자열보다는 정수를 사용하는 것이 훨씬 빠르게 수행되기 때문에 내부 형식으로 표현하면 프로그램의 디지털 라이선스 생성처리를 위한 성능상의 이점이 있다.

III. 프로그램 소스 코드의 디지털 라이선스 설계

프로그램에 대해 진정으로 이해하기 위해서는 코드와 아키텍처를 보다 면밀하게 구분할 필요가 있다. 코드 또는 소프트웨어는 컴퓨터에 의해서 수행되어질 프로그램을 구성하는 벽돌과 시멘트에 해당한다. 한편 아키텍처는 코드를 벽돌 삼아 쌓아올린 구조물을 의미하는 것이다. 코드는 행위에 제약을 가할 수도 있고 구조에 영향을 미치기도 한다. 그렇지만 모든 것이 오로지 코드에 의해서만 결정되는 것은 아니다. 본 논문에서는 프로그램 소스 코드의 디지털 라이선스를 생성하는 과정을 제안하고 [그림 1]에 나타내었다.

1. 디지털 라이선스를 위한 DOM 트리의 구성

DOM 모델은 문서 객체들의 표준 집합을 통하여 응용 프로그램들에 대한 플랫폼 중립적이며 언어 중립적인 인터페이스를 정의한다. 본 연구에서는 프로그램 소스 코드의 디지털 라이선스 생성에 DOM 모델을 적용한다. 프로그램 소스 코드의 디지털 라이선스는 디지털 라이선스(Digital License)라는 루트 노드(Root Node)로 구성되어 그 아래에 두 개의 자노드(Child Node)를 갖는다. 자노드는 프로그램 소스 코드의 저작자에 대한 정보를 갖는 저작자 DOM 트리와 프로그램 소스 코드의 아키텍처를 나타내는 프로그램 아키텍처 DOM 트리로서 [그림 1]과 같이 이루어진다.

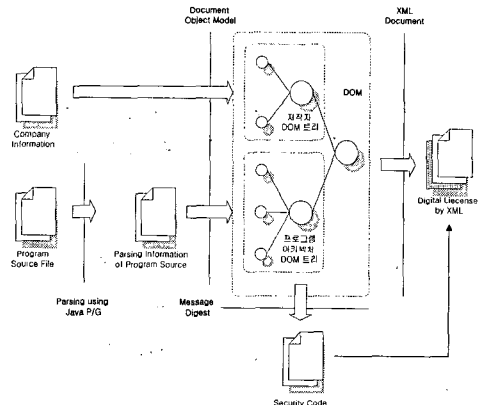


그림 1. 프로그램 소스 코드의 디지털 라이선스를 생성 과정

프로그램 소스 코드에 대한 암호화 기법은 단지 패키징 기법 중의 하나일 뿐이다. 본 연구에서는 프로그램 소스 코드를 파싱하여 예약어, 라이브러리 그리고 함수 등을 DOM 트리 구조로 구축하는 방법을 [그림 2]와 같이 제안한다. C언어 프로그램 소스 코드의 예를 들어, 모든 C언어의 프로그램은 main() 함수와 서브 함수로 구성된다. 그러므로, main() 함수가 디지털 라이선스의 프로그램 아키텍처 부분의 DOM 트리의 루트 노드가 되며, 서브 함수는 main() 함수의 자노드로 구성된다. 각 노드에는 사용된 라이브러리와 변수들 자료형, 연산자들의 패턴 정보들을 갖는다.

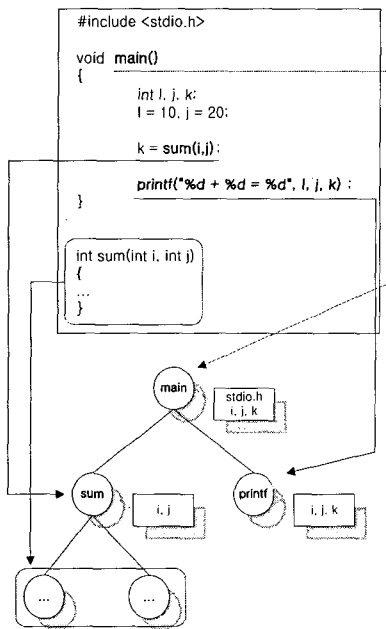


그림 2. 프로그램 소스 코드의 프로그램 아키텍처 DOM 트리의 변환 과정

2. 프로그램 소스 코드 분석

대부분의 DRM 기술은 바이너리 파일에 대한 DRM 기술이다. 원래의 디지털 저작물인 바이너리 코드를 패키징을 수행하여 디지털 저작권 관리가 이루어진다. 그러나 디지털 저작물의 원천 요소인 프로그램 소스 코드에 대해서는 암호화 이외에는 별다른 방법이 존재하지 않으며, 아직은 연구 초기 단계이다. 암호화 방법 역시

프로그램 소스 코드 개발자와 프로그램 소스 코드 구입자간의 1차적인 거래에만 저작권 관리를 지원할 뿐 그 이후에 발생하는 거래에 대한 저작권을 보호해 주지 못하고 있다.

프로그램 소스 코드에 대한 DRM 기술이 연구 초기 단계인 이유는 소스 코드 자체가 아스키(ASCII) 코드나 유니코드(Unicode)로 구성되어 있기 때문이다. 이런 이유로 인하여, 암호화를 통한 1차적 거래에만이 1:1 관계의 디지털 저작권 관리가 이루어지고 있다.

프로그램 소스 코드에 대한 디지털 라이선스를 생성하기 위해서는 먼저 프로그램 소스 코드를 구성하는 프로그래밍 언어의 예약어, 라이브러리 그리고 함수 등을 이용한다. ANSI 표준 C 언어인 경우는 32개의 예약어를 정의하고 있다. 또한 많은 C 컴파일러들은 컴파일러 환경의 효율성, 다른 언어와 상호 프로그래밍, 인터럽트, 메모리 관리 등을 지원하기 위해서 다른 예약어 11개를 추가하고 있다. Java 언어의 경우는 48개의 예약어를 정의하고 있다.

IV. 프로그램 소스 코드의 디지털 라이선스를 위한 자노드 패턴의 생성과정

프로그램 소스 코드의 디지털 라이선스를 생성 시에는 최상의 루트 노드는 프로그램 디지털 라이선스(PGDL : Program Digital License)이다. 루트 노드인 PGDL은 프로그램 저작자 DOM 트리 노드와 프로그램 아키텍처 DOM 트리 노드라는 두 개의 자노드를 갖는데, 왼쪽 자노드는 저작자에 대한 정보를 표현하는 DOM 트리 구조로 구성된다. 일반적인 소프트웨어 등록증의 내용과 정보의 무결성을 보장하기 위한 메시지 축약(Message Digest)[7] 코드가 추가된다.

프로그램 소스 코드의 디지털 라이선스의 핵심 부분은 오른쪽 자노드인 프로그램 아키텍처 DOM 트리이다. 이 자노드는 프로그램 소스 코드를 파싱하여 생성된 예약어 토큰과 프로그램의 구조에 의해서 트리 형태의 아키텍처로 [그림 2]와 같은 형태로 생성된다.

1. 프로그래밍 언어의 예약어

모든 프로그래밍 언어는 예약어를 갖고 있다. 예약어를 코드라 하며, 코드를 이용하여 토큰을 생성하고, 토큰과 프로그램의 구조인 토큰의 위치를 이용하여 프로그램의 아키텍처를 구축한다. 예를 들어, ANSI 표준 C 언어의 경우는 32개의 예약어와 ANSI 표준 C 언어에 등록되지 않은 11개의 보조 예약어를 갖는다. C 언어로 작성된 프로그램의 호환적 측면을 위해서는 ANSI 표준 C 언어의 예약어만을 사용한다.

2. 함수를 이용한 DOM 자노드 구성

프로그램 아키텍처 노드의 루트는 main() 함수가 된다. main() 함수의 루트 노드 아래에 여러 개의 자노드를 포함한다. 자노드는 프로그램의 소스 코드의 블록과 제어문 그리고 함수 호출로 구성된다.

2.1 블록을 이용한 자노드 생성

프로그램의 소스 코드에는 중괄호("{", "}")로 나타내어지는 블록을 포함하고 있다. 블록은 수행될 코드의 묶음으로서 이를 자노드로 생성한다. 블록은 서브 함수와 연속으로 수행될 명령 문장 그리고 블록을 내포할 수 있다. 그러므로 블록을 하나의 자노드로 생성한다.

2.2 제어문에 의한 자노드 생성

프로그래밍 언어의 제어문에는 조건문, 반복문 그리고 분기문이 있다. 제어문은 주어진 조건에 의해서 프로그래밍된 임무의 진행 순서나 절차가 바뀌어 진다. 동일한 프로그램을 수행하더라도 주어진 조건의 상황에 따라 프로그램의 수행은 달라지게 된다. 그러므로 제어문을 자노드로 생성한다.

2.3 함수의 호출에 의한 자노드 생성

C 언어는 최소한 한 개 이상의 함수로 구성된다. 크게 main() 함수와 서브 함수로 구분되며, 서브 함수를 자노드로 생성한다. 서브 함수는 또한 다른 서브 함수나 자신을 다시 호출할 수 있으며, 이를 자노드로 생성한다. 함수 호출에서 무한 루프와 재귀적인 호출에 대해서는 동일한 함수가 3번 이상 호출이 발생하면 연속적인 자

노드를 2개만 생성한다.

3. 자노드의 연산자와 입출력 자료형 패턴 생성

함수의 코드 중에 연산자가 존재하지 않는 경우에는 자노드를 생성하지 않는다. 연산자를 포함하는 함수만이 입력에 대해 새로운 출력을 생성할 수 있다. 즉, 연산자를 포함하는 함수이어야만 임의의 프로세싱으로 간주하고, 연산자를 포함하지 않는 코드는 단지 화면의 출력 이거나, 악의적인 경우의 원본 코드를 속이기 위한 빈 코드의 삽입으로 간주할 수 있다. 그러므로, 이런 코드에 대해 대처할 수 있는 트리의 축약가정이 필요하다. 모든 노드에는 노드만이 갖는 패턴을 생성하게 된다. 노드에 나타내는 패턴은 입력된 자료형과 수, 연산자 그리고 출력된 자료형과 수로 노드의 패턴을 내부 형식(internal format)으로 생성한다.

노드의 패턴 데이터를 이용하게 되면, 프로그램의 아키텍처가 동일하더라도 [그림 3]과 [표 1]과 같이 노드의 패턴 데이터로 프로그램의 기능이 다름을 구분할 수 있게 된다. 즉, sum 노드를 입력 패턴은 두 개의 정수형 데이터, 연산자 패턴은 +, 출력 패턴은 정수형 데이터로 표현되는 노드를 표현함으로써 프로그램의 아키텍처만을 비교하는 단점을 극복할 수 있게 되며, 프로그램의 구별 능력을 증대시키게 된다.

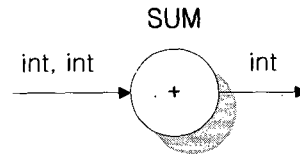


그림 3. sum 노드의 표현

표 1. sum 노드의 연산자와 입출력 패턴 데이터

```

<sub_func name="sum">
<input>int,int</input>
<operator>+</operator>
<sub_func>empty</sub_func>
<output>int</output>
</sub_func>
    
```

4. 빈 노드의 트리 축약

연산자를 포함하지 않은 함수 노드는 DOM 트리의 축소와 실제 소스코드의 분석과정에서 차별화를 위한 어떠한 의미를 제공하지 않는다. 그러므로 연산자 패턴을 포함하지 않은 함수 노드를 제거해도 소스코드를 같은 의미로 볼 수 있다. 그러므로 [그림 4]와 같이 연산자 패턴을 갖지 않은 함수 노드를 제거할 수 있으며, 더불어 DOM 트리의 축소를 수행할 수 있다.

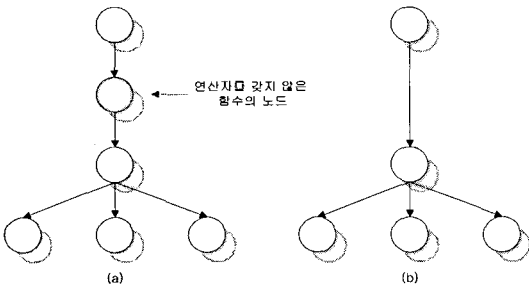


그림 4. 연산자를 갖지 않은 함수 노드의 제거

5. 중복 노드 제거와 트리 순회를 이용한 인덱스 패턴 생성

프로그램 소스코드의 디지털 라이선스에는 인덱스 패턴을 갖고 있다. 인덱스 패턴에 의해서 디지털 라이선스에 기록된 프로그램에 대한 순차패턴과 클러스터링 정보를 제공한다. 만약, 임의의 두 프로그램의 인덱스 패턴을 비교해서 동일한 인덱스 패턴으로 매칭되면, 프로그램의 아키텍처와 노드의 패턴에 의해서 두 프로그램은 정밀한 분석이 이루어진다. 인덱스 패턴의 생성 과정은 DOM 트리의 순회 방법인 중순위(In-order) 방법에 의해서 패턴을 나열하게 되며, 나열된 순차 패턴에서 처음 나타난 다음의 중복된 패턴을 제거하여 인덱스 패턴을 생성하게 된다.

6. 프로그램 버전 관리

인덱스 패턴은 프로그램을 구분 및 분류하는데도 이용되지만, 프로그램의 버전(Version) 관리에도 필요한 정보를 제공하는데 사용된다. A라는 프로그램이 개발되어 유지보수되면 버전업 과정이 이루어진다. 버전업 과

정을 위해서는 최소한 인덱스 패턴은 동일하더라도 프로그램 아키텍처나 노드의 패턴이 동일해서는 안된다. 프로그램의 버전업은 과거의 프로그램 버전과는 많은 부분은 동일하더라도 일부분에서는 노드 패턴의 변경, 노드 위치의 변동, 새로운 노드의 추가와 구 노드의 제거 등의 과정이 나타나게 될 것이다. 이러한 정보에 의해서 프로그램의 버전업 관리가 이루어지게 된다.

V. 다중 클러스터링 정보 생성

프로그램 소스코드의 디지털 라이선스의 정보를 구축하기 위해선 단 하나의 정보를 생성하기 보다는 많은 정보를 제공하기 위한 다양한 정보를 생성하여야 한다. 프로그램의 소스코드를 제공하지 않는 대신에 그에 해당하는 충분한 정보를 제공하여야만 디지털 라이선스의 기능을 수행할 수 있기 때문이다.

1. 인덱스 패턴

프로그래밍 언어의 예약어에 의한 예약어의 발생 순차 패턴과 중복 제거에 의한 인덱스 패턴을 생성한다. 기본적으로 이 인덱스 패턴에 의해서 프로그램을 구분 및 분류 그리고 비교를 수행한다. C 언어의 예약어는 32개이므로 main() 함수를 루트 노드로 설정한 32!(2.6313083693369353016721801216e+35)의 가능한 인덱스 패턴이 존재할 수 있다.

2. 프로그램 아키텍처

인덱스 패턴이 동일한 경우에는 어느 정도 유사한 프로그램으로 간주할 수 있다. 그래서, 노드로 구성된 프로그램 아키텍처에 의해서 프로그램을 분류할 수 있다. 코드는 컴퓨터에 의해서 수행되어질 프로그램을 구성하는 기본적인 구성 요소에 해당한다. 한편 아키텍처는 코드라는 기본 구성 요소로 이루어진 구조물을 의미한다. 코드는 행위에 제약을 가할 수도 있고 구조에 영향을 미치기도 한다. 그렇지만 모든 것이 오로지 코드에 의해서만 결정되지 않으며, 아키텍처에 의해서도 분류 정보를 제공할 수 있다.

3. 노드의 패턴 정보

인덱스 패턴과 프로그램 아키텍처가 동일한 경우에는 거의 동일한 프로그램이라고 할 수 있다. 그러나 프로그램도 생명주기(Life-Cycle)을 갖기 때문에 과거의 프로그램의 패턴을 상속될 수 있다. 이런 경우를 버전업되었다고 하는데, 노드의 패턴 정보에 의해서 이러한 정보를 제공할 수 있다.

4. 메시지 축약과 암호화

모든 데이터는 메시지 축약(MD:Message Digest)에 의해서 데이터 무결성을 보장받는다. 악의의 행위에 대해 데이터의 무결성을 보장하고, 개인 정보의 유출을 막기 위해서는 임의의 항목들은 암호화가 수행된다. MD의 특성은 임의의 입력된 대용량 데이터 집합을 짧고 축약된 식별자 데이터를 생성한다[7]. 또한 메시지 축약은 일방향 함수의 특성을 갖으며, 이는 이상적으로 주어진 축약 값을 생성하는 입력이 무엇인지를 계산하는 것이 불가능하다는 것을 의미한다.

VI. XML 기반의 디지털 라이선스 설계 및 보완사항

1. XML 기반의 디지털 라이선스 설계

프로그램 소스 코드의 디지털 라이선스는 XML 기반

의 프로그램 소스 코드에 대한 프로그램 작성자, 프로그래밍 언어, 등록 날짜, 등록번호, 버전 번호, 프로그램 소스 코드의 DOM 정보 등이 기술되어 있다. 다음의 [표 2]와 같이 프로그램 소스 코드의 디지털 라이선스를 위한 PGDL.dtd를 정의한다. 프로그램 소스 코드의 디지털 라이선스의 무결성과 비밀성을 보장하기 위해서 XML로 만들어진 디지털 라이선스 코드의 일부분을 암호화를 수행한다. 본 연구에서는 JCE[8]의 패키지를 이용하여 메시지 축약을 수행하며 [그림 5]와 같이 나타난다.

2. 보완사항

프로그램 소스 코드가 상당히 길고, 복잡한 특성을 갖고 있다. 또한 서로 다른 프로그램이 유사한 제어 구조로 작동된다면, 즉 유사한 알고리즘으로 구성되어 작동하면서도 약간 또는 일부만을 수정하여 작성한 경우에 대한 소유권 검증 과정에서 유일성을 확인하기 어려울 것으로 판단된다. 또한 유일성을 명확히 판단하기 어려운 경우도 발생할 수 있을 것이다. 그러므로, 유일성을 제공할 수 있는 패턴 생성 분야에 연구가 더 필요하다고 생각한다. 추가하여, 프로그램에 대한 보호 및 저작권 정보를 좀더 명확하게 기술할 수 있는 향후 연구가 필요하다.

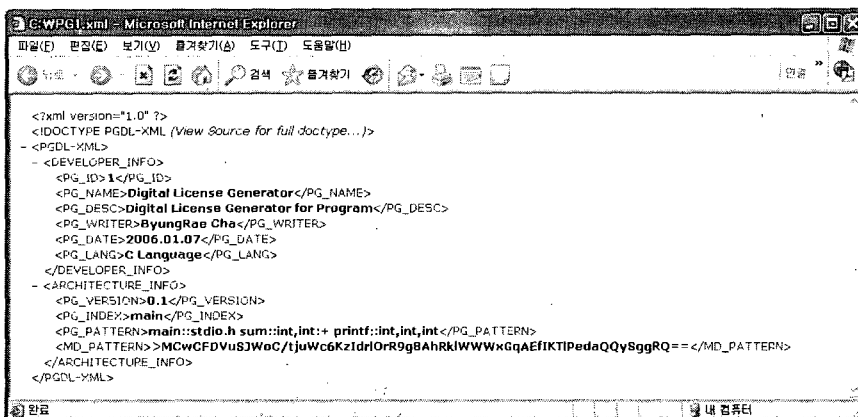


그림 5. 프로그램 소스 코드의 디지털 라이선스

표 2. PGDL.dtd 정의

```

<!-- PGDL.DTD : DTD for Digital License of
Program Source Code -->
<!ELEMENT PGDL-XML (DEVELOPER_INFO
| ARCHITECTURE_INFO)*>
<!ELEMENT DEVELOPER_INFO (PG_ID, PG_NAME,
PG_DESC, PG_WRITER, PG_DATE, PG_LANG))
<ATTLIST DEVELOPER_INFO PG_ID
CDATA #REQUIRED
PG_NAME CDATA #REQUIRED
PG_DESC CDATA #REQUIRED
PG_WRITER CDATA #REQUIRED
PG_DATE CDATA #REQUIRED
PG_LANG CDATA #REQUIRED>
<!ELEMENT ARCHITECTURE_INFO (PG_VERSION, PG_INDEX,
PG_PATTERN, MD_PATTERN)>
<ATTLIST DOM_INFO PG_VERSION
CDATA #REQUIRED
PG_INDEX CDATA #REQUIRED
PG_PATTERN CDATA #REQUIRED
MD_PATTERN CDATA #REQUIRED>
    
```

Ⅶ. 결론

과거 산업사회의 태동과 발전과정에서 뿐만 아니라, 21세기 정보화 사회의 발전에 있어서도 이러한 지적재산권 제도는 중요한 역할을 하고 있다. 우리나라의 국가 경쟁력 제고를 위해서도 디지털콘텐츠에서 확대된 프로그램 소스 코드에 대한 지적재산권 제도와 기술의 정비는 매우 중요한 의미를 지닌다. 디지털 저작물의 원천 요소인 프로그램 소스 코드에 대해서는 암호화이외에는 별다른 방법이 존재하지 않으며, 아직은 연구 초기 단계이다. 암호화 방법 역시 프로그램 소스 코드 개발자와 프로그램 소스 코드 구입자간의 1차적인 거래에만 저작권 관리를 지원할 뿐 그 이후에 발생하는 거래에 대한 저작권을 보호하지 못하고 있다. 프로그램 소스코드의 소유권 분쟁이 발생시 소유권을 증명하기 위해서는 원본의 프로그램 소스코드를 공개해야만 하는 문제점을 갖고 있다. 본 논문에서는 프로그램 소스코드의 원본 공개로 인한 저작권 침해와 기술 유출을 막기 위한 프로그

램 소스코드의 디지털 라이선스의 설계를 제안하였다. 향후 연구로는 프로그램 소스 코드에 대한 정보를 포함 가능한 다양한 패턴 정보의 생성과 생성된 패턴 정보의 검색을 위한 연구가 필요하다.

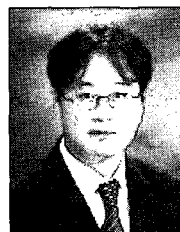
참고 문헌

- [1] Intel, "Content Protection in the Digital Home," Vol.6, Issue4, Intel Technology Journal, Nov., 2000.
- [2] M. Ripley, "Utilizing Content Protection Technologies," Intel Developer Forum, Sep., 2002.
- [3] Intel, "Advanced Digital SetTop Box Design - White Paper Revision 1.0," Sep., 2003.
- [4] A. M. Eskicioglu, "Multimedia Protection in Digital Networks," CNIS2003, 2003.
- [5] <http://www.w3.org/>
- [6] H. Matuyama, K. Tamura, and N. Uramoto, "XML and Java Developing Web Applications," Addison Wesley, 1999.
- [7] J. Garms and D. Somerfield, "Professional Java Security," Wrox, 2001.
- [8] J. Knudsen, "Java Cryptography," O'Reilly, 1998.

저자 소개

차 병 래(Byung-Rae Cha)

정희원

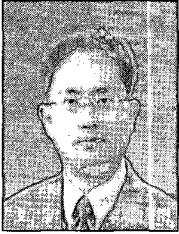


- 1997년 2월 : 호남대학교 컴퓨터 공학과(공학석사)
- 2004년 2월 : 목포대학교 컴퓨터 공학과(공학박사)
- 2005년 3월~현재 : 호남대학교 컴퓨터공학과 전임교수

<관심분야> : 컴퓨터 시스템 보안, 콘텐츠 보호

김 형 중(Hyung-Jong Kim)

정회원



- 1997년 2월 : 조선대학교 전자공학
학과(공학석사)
- 2000년 2월 : 조선대학교 전자공
학과(공학박사)
- 2005년 3월~현재 : 호남대학교
전파이동통신공학과 전임교수

<관심분야> : 이동통신시스템, 콘텐츠 보호

이 동 섭(Dong-Seob Lee)

정회원



- 1985년 2월 : 광운대학교 전자공
학과(공학석사)
- 1985년 1월~1999년 12월 : 대우
통신 수석연구원
- 2000년 1월~2005년 2월 : (주)우
영 전무이사

- 2005년 3월~현재 : 호남대학교 정보통신공학과 전임
교수

<관심분야> : 임베디드 시스템, 지적재산권에 대한
가치평가, 콘텐츠 보호