논문 2023-4-1 http://dx.doi.org/10.29056/jsav.2023.12.01

위탁개발 소프트웨어와 프로젝트의 저작권 이용허락 관리를 위한 PBOM 및 SBOM 설계

박경엽*, 김현수*, 장성일*, 조용준*, 신동명*†

Design of PBOM and SBOM Supporting the Management of License for Outsourced Software Development and Project

Kyung-Yeob Park*, Hyun-Soo Kim*, Sung-Il Jang*, YongJoon Joe*, Dong-Myung Shin*†

요 약

최근 공공, 산업, 경제 등 모든 산업 전반에 걸쳐 소프트웨어가 활용되고 있으며, 이러한 환경에서 기업들은 소프트웨어를 위탁개발하는 사례가 늘어나고 있다. 이는 기업에게 생산 효율 증진 등의 여러 이익을 가져다주지만, 이로 인해 산출되는 많은 소프트웨어 및 프로젝트가 저작권 보호 및 관리없이 운영되어 산출된 소프트웨어를 무단 이용하는 등의 문제가 존재하며, 이를 관리 및 보호할 수 있는 기술의 도입이 시급하다. 이러한 문제를 해결하기 위해 본 연구에서는 위탁개발 소프트웨어 프로젝트를 관리할 수 있는 PBOM과 산출 소프트웨어를 관리할 수 있는 SBOM의 데이터 구조 설계 및 생성 알고리즘을 통해 프로젝트 및 소프트웨어 정보 관리와 저작권 이용허락 관리를 지원할 수 있는 기술을 제안하고, 기존 기술과의 비교를 통해 기존에 제공되지 못했던 요구사항에 대한 충족 여부를 분석하였다.

Abstract

Recently, software is being used across all industries, including public, industrial, and economic sectors. Accordingly, the number of cases of companies outsourcing software is increasing. This brings many benefits to companies, such as increased production efficiency, but many software and projects produced through outsourcing are operated without copyright protection and management, resulting in problems such as unauthorized use of the produced software. Therefore, it is necessary to introduce technology that can manage and protect such copyrights. To solve this problem, we design the data structures of PBOM, which can manage contract development software projects, and SBOM, which can manage output software. In addition, we proposed technologies that can support project and software information management and copyright license management, and analyzed whether requirements that were not previously provided were met through comparison with existing technologies.

한글키워드 : 소프트웨어, 저작권, 이용허락 관리, 소프트웨어 자재명세서, 프로젝트 자재명세서

keywords : software, copyright, license management, software bill of materials, project bill of materials

* 엘에스웨어(주) 소프트웨어연구소 연구개발본부
† 교신저자 : 신동명(email: roland@lsware.com)

접수일자: 2023.11.30. 심사완료: 2023.12.09.
게재확정: 2023.12.20.

1. 서 론

최근 기업들의 참여로 오픈소스 생태계가 급성장하며, 소프트웨어 산업 패러다임에 큰 전환이 이루어지고 있으며, 이러한 전환이 가속화됨에 따라서 모든 분야에서의 소프트웨어 융합이 이루어지고 있다[1]. 따라서 소프트웨어 개발은 기존의 점진적 개발이 아닌 컴포넌트 기반 개발로 전환되고 있으며, 기업은 생산성 향상 및 전문성의 확보를 위해 소프트웨어 개발을 외부 업체에 위탁하는 경우가 늘어나고 있다.

이와 같이 소프트웨어 위탁개발의 수요가 증가하면서 많은 프로젝트와 그에 따른 산출 소프트웨어가 발생하지만 이러한 프로젝트를 관리하고 산출 소프트웨어에 대한 저작권을 관리하는 기술은 부재하여 다양한 분쟁이 발생하고 있는 실정이다. 예를 들어, 프로젝트를 발주한 원청 업체가 프로젝트 수주자인 하청 업체가 개발한 소프트웨어의 저작권 권리를 계약과 달리 부당하게 요구하는 경우나 오픈소스 개발 시 상용 소프트웨어의 무단 사용으로 인한 저작권 침해가 발생하는 등 여러 문제가 발생하고 있다.

따라서 본 논문에서는 이러한 문제를 해결하기 위해 프로젝트와 해당 프로젝트 산출물인 소프트웨어의 저작권 이용허락 관리를 제공할 수 있는 PBOM(Project Bill of Materials) 및 SBOM(Software Bill of Materials) 데이터 구조를 제안하고 이에 대한 평가를 수행한다. 본 논문은 다음과 같이 구성된다. 2장은 관련연구로써 기존 SBOM과 소프트웨어 라이선스(이용허락) 기술에 대해 설명하고, 3장에서는 제안하는 PBOM 및 SBOM 데이터 구조와 생성 과정에 대하여 기술하며, 4장에서는 제안한 기술과 기존 기술과의 비교를 통해 분석을 기술하며, 마지막으로 5장에서 결론에 대하여 기술한다.

2. 관련 연구

2.1 SBOM

SBOM은 소프트웨어를 빌드하기 위해 사용된 구성요소(컴포넌트) 정보와 각 구성요소 간의 관계를 나타내기 위한 메타데이터이다. SBOM은 제조업 분야에서 사용된 원자재를 명시하기 위한 자재명세서(Bill of Materials, BOM)의 개념을 소프트웨어에 적용한 것으로, 이는 소프트웨어 공급망 관리를 용이하게 하고, 알려진 혹은 새로운 취약점을 추적할 수 있도록 한다[2].

SBOM는 SPDX(Software Package Data eXchange), CycloneDX, SWID(SoftWare IDentification)와 같은 세 가지 표준 포맷이 존재한다. 각각의 표준 중 SPDX는 오픈소스 라이선스 관리, CycloneDX는 오픈소스 보안 취약점, SWID는 소프트웨어 수명 주기 관리를 중점으로 하며, 세 가지 표준 중 일반적으로 사용되는 형식은 SPDX 형식이다[3][4]. 표 1은 가장 일반적으로 사용되는 형식인 SPDX 표준의 문서 구조에 대하여 정리하였다.

표 1. SPDX 문서 구조
Table 1. structure of SPDX documents

포함 정보	설명
문서 생성 정보 (SPDX Document Creation Information)	- SPDX 문서 버전, 데이터 라이선스, SPDX 고유 식별 번호, 문서 이름 등
패키지 정보 (Package Information)	- 패키지 이름, 패키지 SPDX 고유 번호, 패키지 정보, 패키지 파일 이름, 패키지 공급자 등
파일 정보 (File Information)	- 패키지에 포함된 각 파일의 라이선스 및 저작권 정보
스니펫 정보 (Snippet Information)	- 웹 / 다른 SW 제품으로부터 사용한 내용의 저작권 및 라이선스 조건이 첨부된 코드 정보
감지된 기타 라이선스 정보 (Other Licensing Information Detected)	- 패키지에 포함된 라이선스 중 표준에 정의되지 않은 라이선스 정보
SPDX 구성요소 관계 (Relationships between SPDX Elements Information)	- SPDX 구성요소(컴포넌트) 사이의 관계를 나타내는 정보
주석 정보 (Annotations Information)	- 파일, 패키지 또는 전체 문서에 대한 설명을 기술

미국 정보 통신국(National Telecommunication and Information Administration, NTIA)은 SBOM을 생성할 때 필수적으로 생성되어야 할 최소 요소를 정의하였으며, 정의된 SBOM의 최소 구성요소는 데이터 필드, 자동화 지원, 실제 사용과 프로세스로 구성된다. 데이터 필드에는 공급자명과 소프트웨어 구성요소 명, SBOM 작성자 등이 존재하며, 자동화 지원 필드에는 SBOM의 규격화된 표준, 마지막 실제 사용과 프로세스 필드에는 SBOM의 배포 및 전달 등의 정보가 저장된다[5][6]. 표 2는 이러한 SBOM의 최소 구성요소 항목과 설명에 대하여 정리한다.

표 2. SBOM 최소 구성 요소
Table 2. Minimum components of SBOM

최소 요소	설명
데이터 필드 (Data Fields)	- 필수적으로 추적 및 유지 관리해야 하는 각 구성요소에 대한 기본 정보 (e.g., 구성요소의 공급자, 이름, SBOM 작성자 및 작성 일시 등)
자동화 지원 (Automation Support)	- 소프트웨어 생태계 상에서의 적용을 위한 자동 생성 및 기계 가독성 등을 포함한 자동화 지원 - SBOM 생성 및 소비를 위한 포맷을 의미 (e.g., SPDX, CycloneDX, SWID tag)
실제 사용과 프로세스 (Practice and Process)	- SBOM 유형/생성/사용에 대한 전반적인 운영 관련 정의 (e.g., SBOM 생성 빈도수, 접근제어, 배포 및 전달 등)

2.2 소프트웨어 라이선스

소프트웨어는 일반 기업에서 개발하고 구매를 통해 사용권을 제공하는 상용 소프트웨어, 무상으로 프로그램을 사용할 수 있는 프리웨어(Freeware), 누구나 자유롭게 확인하고 수정 및 배포를 수행할 수 있도록 코드를 공개적으로 제공하는 오픈소스 소프트웨어 (Open Source Software)등으로 나누어진다. 일반적으로 소프트웨어를 구입했을 때는 소유권이 취득되는 것이 아닌 소프트웨어에 대한 이용허락을 받는 것으로 라이선스 계약 형태를 가진다[7]. 오픈소스에는 GPL, Apache License 등의 라이선스 정보가 명시되어 있고, 해당 오픈소스를 활용하여 SBOM을 생성할 경우 라이선스 정보가 같이 포함되지만, 오픈소스가 아닌 일반 소프트웨어를 사용하여 N차 저작물을 생성하여 SBOM으로 생성할 경우, 별도의 라이선스 정보가 포함되어야 하지만 제대로 준수되고 있지 않은 실정이다[8].

이러한 소프트웨어 라이선스 보호 및 관리를 위해서 다양한 기술 개발이 진행되었는데, 예를 들어 저작자 식별코드를 생성하여 소스코드에 삽입하고 향후 해당 소스코드 내에서 저작자 정보를 자동으로 추출 및 검증하는 기술과 소스코드를 난독화하여 해당 소스코드가 유출되더라도 이를 식별할 수 없도록 만드는 기술 등이 존재한다. 하지만 해당 기술들은 기술적 어려움이 존재하여 실제 도입이 미진하고, 소프트웨어 개발의 전 생애주기(Life Cycle)에 걸쳐 저작권 보호 및 관리를 지원할 수 없다는 단점이 존재한다[9][10].

3. 프로젝트 및 산출 소프트웨어 관리를 위한 BOM 설계

그림 1. PBOM 및 SBOM 생성 흐름도
Fig. 1. Flow of PBOM and SBOM Generation

제안하는 기술은 SBOM 표준 중 하나인 SPDX 2.3 버전을 기준으로 설계하였으며, 프로젝트에 대한 정보와 프로젝트 계약 시 체결된 저작권 이용허락 정보를 기록할 수 있는 PBOM과 해당 프로젝트를 통해 산출된 소프트웨어에 대한 정보 및

소프트웨어의 저작권 이용허락 정보를 포함하는 SBOM으로 구성된다. 그림 1은 이러한 PBOM 및 SBOM 생성에 관련된 프로세스를 간략화하여 나타낸다. 프로젝트 발주자는 의뢰하고자 하는 프로젝트 공고를 업로드하고 프로젝트 수주자와의 계약을 체결한다. 이 때 발주자와 수주자 간에 계약 조건은 PBOM을 통해 저장되며, 이후 프로젝트 조건이 변경될 때마다 신규 PBOM이 생성된다.

프로젝트 진행 시 생성되는 소프트웨어의 소스코드는 git 프로젝트 단위와 같이 규격화된 소스코드 관리 환경을 통해 소스코드의 commit을 수행할 때마다 새로운 SBOM이 생성된다. 그림 2는 이러한 SBOM 생성 시나리오를 표현한다.

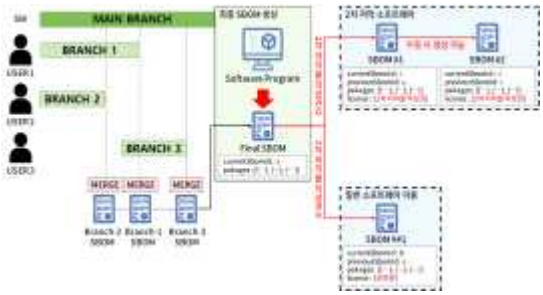


그림 2. SBOM 생성 시나리오
Fig. 2. SBOM Generation Scenario

표 3. PBOM 및 SBOM 데이터 구조
Table 3. Data structure of PBOM and SBOM

	대분류	중분류		설명
PBOM	PBOM Information	spdxVersion		PBOM 생성 시 사용한 SPDX 버전
		pbomSpdxId		현재 PBOM ID
		previousPbomSpdxId		이전 PBOM ID
		createdDate		PBOM 생성 시각
		creator		PBOM 생성자
	Project Information	Project Related Information		라이선스 등 프로젝트 관련 정보
	PBOM Extend	Software Information	sbomSpdxId	산출물 소프트웨어의 현재 SBOM ID
			previousSbomSpdxId	산출물 소프트웨어의 이전 SBOM ID
		Contract Information	contractId	계약서 ID
			contractUrl	계약서 URL
		Original Software	provideSoftwareName	기존 소프트웨어 수정 계약이었을 시 제공된 소프트웨어명
			provideSbomSpdxId	기존 소프트웨어 수정 계약이었을 시 제공된 소프트웨어의 SBOM ID
		Hash Manage	pbomSpdxHash	현재 PBOM의 Hash
			previousPbomSpdxHash	이전 PBOM의 Hash
			sbomSpdxHash	현재 SBOM의 Hash
			previousSbomSpdxHash	이전 SBOM의 Hash
SBOM	SBOM Information	spdxVersion		SBOM 생성 시 사용한 SPDX 버전
		sbomSpdxId		현재 SBOM ID
		previousSbomSpdxId		이전 SBOM ID
		createdDate		SBOM 생성 시각
		creator		SBOM 생성자
		packages		산출 소프트웨어 개발 시 사용한 패키지 정보
		files		산출 소프트웨어 개발 시 사용한 파일 정보
		relationships		산출 소프트웨어 개발 시 사용한 소프트웨어 간 관계 정보
	SBOM Extend	Project Information	Id	프로젝트 ID
			pbomSpdxId	현재 PBOM ID
			previousPbomSpdxId	이전 PBOM ID
		Contract Information	contractId	계약서 ID
			contractUrl	계약서 URL
			license	산출 소프트웨어의 저작권 (계약 시 생성)
			licenseOwner	산출 소프트웨어의 저작권 소유자 (계약 시 생성)

3.1 PBOM 및 SBOM 데이터 구조 설계

PBOM 및 SBOM은 서로 상호참조하여 어떤 BOM을 제공받아도 추적이 가능하고, 기존 SBOM에서 제공하던 정보뿐만 아니라 프로젝트 정보와 저작권 정보도 같이 포함할 수 있도록 설계하였다. 표 3은 이러한 기능을 제공하는 제안 기술의 데이터 구조에 대하여 명세한다. 각 구성 요소에 대해서는 아래 절에서 자세히 서술한다.

3.1.1 PBOM Information

PBOM Information 필드에는 PBOM을 생성할 시 사용했던 SPDX에 대한 정보와 PBOM 생성자 등과 같은 기본적인 정보가 포함된다. 생성되는 PBOM ID는 PBOM 생성 프로세스 시작 시 부여되어 포함되며, 이전 PBOM ID를 통해 수정된 프로젝트의 이력을 추적할 수 있다.

3.1.2 Project Information

Project Information 필드는 프로젝트 공고 생성 시 프로젝트 발주자가 입력한 정보들이 포함되며, 해당 필드에는 현재 프로젝트의 상태나 시작일, 비용 등이 입력된다. 또한 해당 필드에 프로젝트 산출물 발생 시 양도할 저작권 이용허락 정보가 포함되어 있다. 이는 실제 계약 발생 시에 변경될 수 있다.

3.1.3 PBOM Extend

PBOM Extend 필드에 Origin Information 필드의 경우 프로젝트 발주가 발생했을 때 해당 프로젝트가 기존에 있던 소프트웨어를 수정하는 프로젝트인지 혹은 기존 소프트웨어 없이 새로운 소프트웨어를 개발하는 프로젝트인지 확인할 수 있는 필드이다. 이외에 필드는 프로젝트 계약 이후에 생성될 수 있는 데이터로써 산출물 발생 시 기록할 수 있는 Software Information 필드와 프로젝트 계약 정보가 저장되는 Contract Information 필드, 마지막으로 외부에서 PBOM 및 SBOM을 제공받았을 때 해당 데이터의 무결성을 검사할 수 있는 Hash Manage 필드로 구성된다.

3.1.4 SBOM Information

SBOM Information 필드는 기존 SBOM과 같이 해당 소스코드가 사용한 패키지 및 파일 정보와 각 소프트웨어 간 관계 정보를 포함한다. 이 뿐 아니라 PBOM Information 필드와 같이 SBOM 생성에 사용한 SPDX의 정보가 기록되며 현재, 이전 SBOM ID가 포함되어 소스코드 및 소프트웨어의 패키지 변경 이력 확인을 용이하게 한다.

3.1.5 SBOM Extend

SBOM Extend 필드에는 소프트웨어 유통 시 해당 SBOM만으로 저작권 정보를 확인할 수 있도록 저작권 이용허락 정보를 포함하는 Contract Information 필드가 존재하며, PBOM에서 SBOM으로의 일방적 참조가 아닌 SBOM에서 PBOM으로의 상호참조가 가능하도록 하는 Project Information 필드가 존재한다.

3.2 PBOM 및 SBOM 데이터 생성 단계

표 4는 초기 PBOM 생성에 대한 알고리즘을 명세하였다. 초기 PBOM 생성은 프로젝트 수주자와 발주자 간의 계약 정보를 포함해야 하기 때문에 데이터베이스에 저장되어 있던 계약 ID를 통해 계약 정보를 불러와 PBOM 생성을 수행한다. 이 과정에서 해당 계약이 기존 소프트웨어를 수정하는 계약인지 최초 소프트웨어 개발을 수행하는 계약인지 검사한다. 만약 기존에 존재하던 소프트웨어를 수정하는 경우라면 PBOM 생성 요청 시에 해당 소프트웨어의 정보와 SBOM을 함께 제공해주어야 한다.

표 4. 초기 PBOM 생성
Table 4. Generate Initial PBOM

```
Algorithm 1: Generate Initial PBOM
Input : ContractID, (OriginSWSbom, OriginSWInfo)
Output : Pbom
if isExists(ContractID) != True then
    return Err
else
    GeneratePbomID()
    if isExists(OriginSWInfo) == True then
        GeneratePbom(PbomID, ContractID, InfoC
         ontract, OriginSWSbom, OriginSWInfo)
    else
        GeneratePbom(PbomID, ContractID, InfoC
         ontract(ContractID))
```

표 5는 초기 PBOM 생성이 완료된 이후 생성되는 모든 PBOM에 대한 알고리즘을 설명한다. PBOM 생성 시 초기 PBOM은 반드시 존재해야 하므로 이전 PBOM 값이 없을 경우 에러를 반환하며, 이후 PBOM 생성 전 SBOM 생성 여부에 대하여 검사한다. 이는 SBOM과 PBOM이 상호 참조 관계에 있어 PBOM의 생성 프로세스가 종료되기 전 SBOM이 생성되어야 하기 때문이다. 일반적인 프로젝트 내용에 대한 수정일 경우 별도의 SBOM을 생성하지 않지만, SBOM을 생성하려는 경우 PBOM의 내용 변경과는 상관없이 PBOM이 생성되어야 한다.

표 5. PBOM 생성
Table 5. Generate PBOM

```
Algorithm 2: Generate PBOM
Input : BeforePbomID, (SourcePath)
Output : Pbom
if isExists(BeforePbomID) != True then
    return Err
else
    GeneratePbomID()
    if isGenerateSbom() then
      GeneratePbom(PbomID, BeforePbomID, Ad
       dInfoArray(InfoPbom(BeforePbomID), Ge
       nerateSbom(PbomID, SourcePath))
    else
      GeneratePbom(PbomID, BeforePbomID, Ad
       dInfoArray(InfoPbom(BeforePbomID)))
```

표 6은 SBOM 생성 알고리즘에 대하여 작성한다. SBOM 생성 시에는 산출 소프트웨어의 원 프로젝트 추적을 위해 PBOM ID를 입력으로 하여 SBOM을 생성한다. 또한, SBOM의 필드에는 저작권 정보가 존재하기 때문에 PBOM ID를 기반으로 PBOM에 존재하는 저작권 정보를 불러와 SBOM 필드에 추가한다.

표 6. SBOM 생성
Table 6. Generate SBOM

```
Algorithm 3: Generate SBOM
Input : PbomID, SourcePath
Output : Sbom
if isExists(PbomID) != True // isEmpty(SourceP
ath) == True then
    return Err
else
    GenerateSbom(PbomID, SourcePath, InfoPbo
     m(PbomID))
```

4. 제안 기술 분석

본 장에서는 기존 기술에 대비하여 제안한 PBOM 및 SBOM 데이터 구조가 제공해줄 수 있는 장점인 무결성, 추적 용이성, 저작권 이용허락 관리 가능성에 대해 제안한 기술이 이를 충분히 제공해줄 수 있는지를 분석한다.

4.1 무결성

무결성이란 제공된 데이터의 정확성과 일관성 및 완전성이 유지되는 것을 보장하는 특성이다. 기존 SPDX에서도 Hash값을 통한 무결성 제공이 존재하기는 하나 본 제안 기술을 활용할 경우, 외부에서 불특정 인원이 제공하는 PBOM 및 SBOM에 대하여 PBOM Extend에 기록된 PBOM 및 SBOM Hash값을 통해 특정 프로젝트 및 그 산출 소프트웨어까지 해당 데이터가 변조되지 않은 데이터인지 검사가 가능하여 충분한 무결성을 제공할 수 있다.

4.2 추적 용이성

본 절에서 설명하는 추적 용이성은 특정 PBOM 및 SBOM이 주어졌을 때 이를 통해 이전 버전이나 산출물 및 프로젝트 정보를 추적할 수 있는지 말하는 특성으로, 기존 SPDX의 경우 소프트웨어가 이용한 라이브러리 정보를 확인할 수는 있지만 이전 버전의 SPDX를 추적할 수는 없다. 본 제안 기술은 새로운 PBOM 및 SBOM이 생성될 때 이전 PBOM 및 SBOM의 ID를 포함하고 있어 이전 버전으로의 추적이 가능하며 PBOM과 SBOM이 상호 참조 관계에 있기 때문에 어떤 프로젝트에서 산출된 소프트웨어인지나 해당 프로젝트에서 어떤 소프트웨어가 산출되었는지 추적하기 용이하다.

4.3 저작권 이용허락 관리 가능성

저작권 이용허락 관리 가능성이란 BOM을 통해서 어떤 소프트웨어가 올바른 사용자에게 올바른 저작권 이용허락 정보로 사용되고 있는지 확인하고 관리할 수 있는 특성으로 기존 SPDX의 경우 소프트웨어 취약점 관리를 위해 설계되어 있고 오픈소스나 해당 소스코드가 다른 소프트웨어 제품에서 복사한 저작권 정보만이 기록되고 해당 소스코드 및 소프트웨어의 저작권 이용허락 정보는 표현할 수 없다는 문제가 있다. 본 제안 기술에서는 프로젝트 발주 시부터 어떤 저작권 이용허락 정보를 제공하기로 했는지가 기록되고, 실제 계약 시에 체결된 저작권 이용허락 정보와 유통되고 있는 산출 소프트웨어 간 비교를 통해 적합한 사용자와 목적으로 사용되고 있는지 확인할 수 있다. 만약 프로젝트 진행 중간에 제공하기로 한 저작권 이용허락 정보가 달라져도 PBOM과 SBOM에 기록되기 때문에 확인이 용이하다.

위의 기술된 제안 기술이 제공해줄 수 있는 3가지 특성을 통하여 프로젝트 발주자는 프로젝트 발주부터 프로젝트 진행, 계약 정보 변경 등을 관리할 수 있다는 장점이 있으며 프로젝트 수주자는 개발 중인 소프트웨어 정보 변경이나 소스코드 변경, 계약 정보 위반 방지 등을 모두 제공받을 수 있다는 장점이 있다. 표 7은 각 절에서 분석한 세 가지 항목에 대해 기존 기술과 제안 기술을 비교한 표이다.

표 7. 기존 SPDX와 제안 기술 간 비교
Table 7. Comparison between SPDX and proposal method

	무결성	추적 용이성	저작권 이용허락 관리 가능성
기존 SPDX	△	△	×
제안 기술	○	○	○

5. 결 론

최근 기업들의 오픈소스 생태계 참여로 인해 소프트웨어 융합 개발 사례가 많아지고, 소프트웨어 산업 패러다임의 변화가 발생하였으며, 또한 기업들은 생산성 증가를 위해 소프트웨어 위탁개발 하는 경우도 많아지고 있는 추세이다. 이에 따라 각종 프로젝트와 해당 프로젝트의 산출물인 소프트웨어의 수도 같이 증가하고 있지만 이를 지원할 수 있는 저작권 관리 기술의 부재로 인해 프로젝트 계약 내용 미준수나 산출 소프트웨어의 저작권 이슈와 같은 다양한 분쟁이 발생하고 있다. 따라서 어떤 프로젝트가 어떤 계약 조건으로 이루어지고 진행되었는지와 해당 프로젝트에 산출 소프트웨어는 어떤 것인지 명시할 수 있는 기술 개발의 필요성이 대두되었다.

이에 본 논문에서는 프로젝트 정보를 포함하는 PBOM과 산출 소프트웨어 정보를 저장할 수 있는 SBOM 데이터 구조를 설계하였다. 본 논문에서 제안한 기술을 통해 프로젝트 발주자 및 수주자는 프로젝트 발주 공고부터 프로젝트 진행 등

일련의 프로세스 및 프로젝트 내용 변경 이력을 추적 및 확인할 수 있을 뿐만 아니라 개발 중인 소프트웨어의 변경이나 계약 정보 무단 변경 및 무단 사용 방지 등을 제공받을 수 있을 것으로 기대한다. 향후 더욱 다양한 프로젝트들의 요구사항 분석과 데이터 설계가 이루어진다면 더 안전하고 신뢰성 있는 소프트웨어 유통망 구축에 기여할 수 있을 것으로 예상된다.

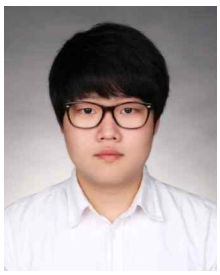
본 연구는 문화체육관광부 및
한국콘텐츠진흥원의 2023년도 SW저작권
생태계 조성 기술개발 사업으로 수행되었음
(과제명 : 클라우드 서비스 활용 구축 형태별
대규모 소프트웨어 라이선스 검증 기술개발,
과제번호 : RS-2023-00224818, 기여율: 100%)

참 고 문 헌

[1] Software Policy & Research Institute, “SBOM(Software Bill of Materials) Introduction & Activation Method Research”, 2023.
[2] Korea Information Security Industry Association, “2022 SBOM Research Report for Supply Chain Protection”, 2022.
[3] Hyo-Hyun Son, Dong Hee Kim, So Jeong Kim, “A Study on the Software Supply Chain Security Policyfor the Strengthening of Cybersecurity : Based on SBOM Policy Cases”, Journal of Digital Convergence, 2022.
[4] Won-Ok Ryoo, Soo Myoung Park, Seung-Yun Lee, “Software Supply Chain Management and SBOM Trends”, Electronics and Telecommunications Trends Vol.38, No.4, 2023.
[5] Software Policy & Research Institute, “A study on Global Open Source Technology Ecosystem Analysis - Focusing on Projects in the Linux Foundation”, 2023.
[6] NTIA, “The Minimum Elements For a Software Bill of Materials (SBOM)”, 2021.07.
[7] Korea Copyright Commission, Korea Software Property·Right Council, “Research on software licensing trends and policies”, 2010.
[8] Korea Copyright Commission, “Open Source Software License Guide 3.0”, 2016.
[9] Jing, Nan, Qi Liu, and Vijayan Sugumaran, “A blockchain-based code copyright management system”, Information Processing & Management, 58.3, 2021.
[10] Xu, Hui, et al., “Layered obfuscation: a taxonomy of software obfuscation techniques for layered security”, Cybersecurity, 3.1, 2020.

“”,

저 자 소 개



박경엽(Kyung-Yeob Park)

2019.02 서울과학기술대학교 컴퓨터공학과 석사
2019.01-현재 엘에스웨어㈜ 소프트웨어연구소 연구개발본부 선임연구원
<주관심분야> 정보보호, IoT 보안, 블록체인, 빅데이터, 분산신원증명, 저작권 기술

김현수(Hyun-Soo Kim)

2019.02 단국대학교 소프트웨어학과 학사
2023.08 숭실대학교 AI,SW융합학과 석사
2019.01-현재 엘에스웨어㈜ 소프트웨어연구소 연구개발본부 선임연구원
<주관심분야> 소프트웨어 공학, 딥러닝, 컴퓨터 비전, 분산신원증명, 빅데이터

장성일(Sung-Il Jang)

2019.08 숭실대학교 컴퓨터학과 석사
2021.08 숭실대학교 소프트웨어학과 박사과정 수료
2021.09-현재 엘에스웨어㈜ 소프트웨어연구소 연구개발본부 수석연구원
<주관심분야> 블록체인, 클라우드, 제로트러스트 기술, 오픈소스 소프트웨어, MSA

조용준(YongJoon Joe)

2011.03 큐슈대학교 전기정보공학과 졸업
2013.03 큐슈대학교 정보학부 석사
2016.03 큐슈대학교 정보학부 박사과정 수료
2013.04-2016.03 일본 학술진흥원 특별연구원
2016.04-현재 엘에스웨어㈜ 소프트웨어연구소 연구개발본부 수석연구원
<주관심분야> 병렬·분산 컴퓨팅, 게임이론, 분산 제약 최적화 문제

신동명(Dong-Myung Shin)

2003.08 대전대학교 컴퓨터공학과 박사
2001-2006 한국정보보호진흥원(KISA) 응용기술팀 선임연구원
2006-2014 한국저작권위원회 저작권기술팀 팀장
2014-2016 한국스마트그리드사업단 보안인증팀 팀장
2016-현재 엘에스웨어㈜ 소프트웨어연구소 연구개발본부 연구소장/상무이사
<주관심분야> 오픈소스 라이선스, 저작권 기술, 시스템/네트워크 보안, SW 취약점 분석·감정, 블록체인 기술