논문 2016-2-1

오픈소스 소프트웨어 라이선스 파일 식별 기술

윤호영*, 조용준*, 정병옥*, 신동명*†

Measurement for License Identification of Open Source Software

Ho-Yeong Yun*, Yong-Joon Joe*, Byung-Ok Jung*, Dong-Myung Shin*†

요 약

본 논문은 오픈소스 소프트웨어의 배포과정에서 라이선스 정보가 누락, 훼손, 변경, 충돌됨에 따라 발생하는 무의적인 저작권 침해를 미연에 방지하고자 라이선스 파일을 추출/식별하는 기술을 연구하였다. 라이선스 파일이 갖는 특성을 파악하기 위해 n-gram과 TF-IDF 기법을 활용하여 322개의 라이선스 내용을 분석하였고, 이를 활용하여 패키지 내에서 라이선스 파일을 추출하였다. 추출한 라이선스는 코사인 측정법을 통해 확보한 라이선스간의 유사도를 산정하여 라이선스 정보를 식별하였다.

Abstract

In this paper, we study abstracting and identifying license file from a package to prevent unintentional intellectual property infringement because of lost/modified/confliction of license information when redistributing open source software. To invest character of the license files, we analyzed 322 licenses by n-gram and TF-IDF methods, and abstract license files from the packages. We identified license information with a similarity of the registered licenses by cosine measurement.

한글키워드 : 오픈소스 소프트웨어, 라이선스, TF-IDF, 코사인 유사도 측정

keywords : open source software, Licence, TF-IDF, cosine simliarity

1. 서 론

오픈소스 소프트웨어는 소스코드의 공개, 무료 로열티, 빠른 기술 발전 속도 등의 강점을 바탕으로 급속하게 발전하는 추세이다. 하지만 기업들이 오픈소스 소프트웨어를 활용하는 과정에서 기업들은 커스터마이징, 호환성 및 상호운용성, 성숙도, 분기, 시스템 통합 및 지원 등의 전략 리스크와 코드 적합성, 문서화, 우발사태대책, 외부지원 등과 같은 운영 리스크에 직면하기도 한다. 또한 라이선스, 제 3자의 권리침해, 면책 및 보증과 같은 법률적인 관점에서의 리스크가 존재하는데 이는 법적 분쟁으로 이어질 소지가 있기 때문에 이를 적절히 관리할 필요성이 대두되고 있다[1].

이와 관련해서 해외에서는 오픈소스 소프트웨어 라이선스의 요구사항을 충족하기 위해 컴플라이언스에 관한 거버넌스를 구축해왔으며, 오픈소스 소프트웨어 프로젝트에 참여하는 기업의 지식

* 엘에스웨어(주)
† 교신저자: 신동명(email: roland@lsware.com)
접수일자: 2016.11.20. 심사완료: 2016.12.3.
게재확정: 2016.12.22.

재산권 관리를 위한 논의가 활발하게 진행되고 있다. 2014년 국내 오픈소스 소프트웨어 시장 규모는 2013년 대비 19.1% 성장한 548억 원으로 추정됐으며, 2009년부터 연평균 32.2%씩 성장하였다. 2013년부터 2018년까지는 연평균 12.3%씩 성장하여 820억 원 규모의 시장을 형성할 것으로 전망되었다[3]. 하지만 매출 100억 이상의 국내 기업 중 오픈소스 소프트웨어를 사용한 기업은 60.8%지만, 오픈소스 소프트웨어 관리체계를 구축한 기업은 10.8%로 아직 정착단계에 이르는 수준이며, 중소기업의 경우 관련 전문가조차 확보하기가 힘든 상황이다. 따라서 해외의 오픈소스 소프트웨어 활동과 논의를 적극적으로 참고하여, 관련된 법적 리스크를 최소화하기 위한 노력이 필요한 시점이다.

오픈소스 소프트웨어는 명시된 라이선스 규정을 명확히 지켜야하지만 오픈소스 소프트웨어의 특성상 최초 개발자가 배포하게 되면 다른 개발자들이 기능을 추가, 수정하여 2차, 3차적으로 배포하게 되는데, 이 과정에서 라이선스의 정보가 누락, 훼손, 변경, 충돌되는 경우가 발생할 수 있다. 또한 대형 프로젝트의 경우 방대한 양의 파일이 존재하기 때문에 이 중에서 라이선스의 정보를 담은 파일을 육안으로 구별하는 작업은 쉽지 않다. 이에 본 연구는 개발자가 오픈소스 소프트웨어를 활용하고자 할 때, 해당 오픈소스 프로젝트 내에서 라이선스 정보를 추출해주는 시스템을 개발하는 연구를 진행하였다.

본 논문의 구성은 다음과 같다. 2장의 관련연구에서는 라이선스를 추출하는 기법에 대한 기존 연구를 조사하였고, 본 연구에 적용한 TF-IDF와 코사인 유사도 측정법에 대해 기술하였다. 3장에서는 현재 등록된 라이선스의 전문을 n-gram을 이용하여 빈도를 측정하여 TF-IDF값을 도출하였고, 이를 기반으로 라이선스 파일의 특성을 기술하였다. 4장에서는 라이선스 파일의 위치 및 상태에 대해 총 4가지 케이스를 가정하여, 라이선스 파일을 추출 및 식별하는 실험을 진행하였다. 5장에서는 결론과 향후 연구 방향에 대해 제시하였다.

2. 관련 연구

2.1 라이선스 파일 분석 툴

라이선스 파일을 분석해주는 오픈소스 소프트웨어는 FOSSology, ohcount, OSLC, ALSA 등이 있으며, 상용 소프트웨어로는 Blackduck의 Protext, MDS Technology의 Palamida가 있다.

HP에서 개발한 FOSSology는 GPL 라이선스가 적용되었는데, Binary Symbolic Alignment Matrix(bSAM) 알고리즘을 기반으로 라이선스 파일을 분석한다[4]. FOSSology는 다른 소프트웨어에 비해 느리다는 단점이 존재한다. ohcount는 오픈소스 소프트웨어 개발자 커뮤니티 사이트인 올로(Ohloh)가 공개한 소프트웨어로 소스 코드 라인 수를 계산하는 툴로 알려져 있다[5]. OSLC 또한 특정한 소스코드가 포함하고 있는 오픈소스 소프트웨어를 검출하는 소프트웨어로 정규 표현식(Regular Regression) 기법을 사용하고 있다[6]. Tuunanen 외 2인이 발표한 소프트웨어 라이선스 자동 분석툴인 ALSA 또한 정규 표현식 기법을 사용하였으며, OSLC, FOSSology와의 비교실험을 통해 우수성을 입증한 바 있다[8].

상용 소프트웨어 중 가장 큰 규모를 자랑하는 Blackduck의 Protex는 방대한 데이터베이스를 기반으로 오픈소스 컴플라이언스 관리 솔루션을 개발하였다. 이는 기업을 대상으로 서비스를 제공하기 때문에 일반인이 사용하기에는 무리가 있다[9]. MDS Technology가 제공하는 오픈소스 관리 솔루션인 Palamida는 오픈소스의 스캔과 분석을 통해 오픈소스 관리체계 구축을 자동화하고 프로젝트의 오픈소스 라이선스와 보안 취약점을 관리한다[10].

2.2 TF-IDF 측정기법

TF-IDF(Term Frequency - Inverse Document Frequency)는 정보 검색과 텍스트마이닝에서 주로 활용되는 기법으로 여러 문서로 이루어진 문서군이 있을 때, 어떤 단어가 특정 문서 내에서 중요한 의미를 지니는지에 대해 나타내는 통계적 수치이다. 문서의 핵심어를 추출하거나, 검색 엔진에서 검색 결과의 순위를 결정하거나, 문서들 사이의 비슷한 정도를 구하는 등의 용도로 사용된다.

TF는 단어의 빈도를 뜻하며, 특정 단어가 문서 내에서 얼마나 자주 등장하는지를 나타내는 값이다. 단순하게 단어의 빈도와 중요도는 비례한다고 해석할 수도 있지만, 단일 문서가 아닌 다수의 문서가 있는 경우에는 이는 다르게 해석가능하다. 단어 자체가 문서군 내에서 자주 사용되는 경우는 해당 단어는 흔하게 등장하는 것을 의미, 즉 중요도가 낮다고 해석할 수 있다. 이를 문서의 빈도를 뜻하는 DF라고 하며, 이 값의 역수를 IDF라고 한다.

$$w_{i,j} = TF_{i,j} \times \log(\frac{N}{DF_i})$$

$TF_{i,j}$ = number of occurrences of i in j
DF_i = number of documents containing i
N = total number of documents

TF-IDF는 TF와 IDF의 곱한 값으로 수식으로 표현하면 위와 같다. DF값을 역수로 변환하게 되면 곡선의 형태를 띄기 때문에 로그값을 취하여 TF의 값과 동일하게 선형의 값을 갖게 변환하는 과정을 거친다.

2.3 코사인 유사도 측정기법

코사인 유사도는 내적공간에 존재하는 두 벡터간 각도의 코사인값을 이용하여 측정된 벡터값으로 계산된다. 코사인 유사도의 결과값은 -1에서 1사이 값을 가지며, -1에 가까울수록 서로 완전히 반대되는 경우이며, 0은 서로 독립적인 경우, 1에 가까울수록 완전히 같은 경우를 의미한다.

코사인 유사도를 활용하여 두 문서간 유사도를 측정하는 경우는 TF-IDF값이 벡터값이 되는데, 단어의 빈도수를 통해 나온 TF-IDF값이 음수일 경우는 없기 때문에 문서간 유사도를 구하는 문제의 경우는 0과 1사이의 값을 갖는다.

$$\cos(\theta) = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n}(A_i)^2} \times \sqrt{\sum_{i=1}^{n}(B_i)^2}}$$

코사인 유사도 측정의 수식은 위와 같다. 이때 A와 B의 값은 각 문서의 TF-IDF값을 뜻한다.

3. 라이선스 파일 특징 분석

3.1 SPDX 라이선스 전문 수집

SPDX(The Software Package Data Exchange)는 소프트웨어 패키지와 함께 오픈 소스 컨텐츠, 라이선스, 저작권과 관련된 표준 정보를 명시한 곳이다. 해당 표준의 목적은 소프트웨어 공급망 내에 있는 기업들이 소프트웨어 라이선스 의무사항을 좀 더 쉽게 준수할 수 있게 하는 것이다. SPDX에는 라이선스 리스트와 해당 전문을 조회할 수 있다. 2016년 11월 현재 2.5버전으로 총 322개의 라이선스가 업로드 되어있다. 원활한 실험을 위해 모든 라이선스의 전문을 템플릿에 맞게 각각 크롤링하였다.

3.2 어절 및 문서의 빈도수 측정

수집한 라이선스의 전문을 토대로 어절의 빈도수(TF)를 n-gram으로 측정하였다. n-gram은 대표적인 확률적 언어 모델로 n개 단어의 연쇄를 확률적으로 표현하는 것을 뜻한다. 즉 n의 값은 어절의 개수를 뜻한다. 어절은 최대 3개까지 추출하였

고, 불용어(Stopwords)를 고려하였다. 어절이 1개인 경우와 2개인 경우는 불용어를 모두 제외하였으나, 3개일 때는 불용어를 모두 제외할 경우 빈도수가 높은 어절이 추출되지 않는 모습을 보여 1어절까지는 불용어를 허용하도록 설정하였다.

방대한 양의 텍스트를 효과적으로 처리하기 위해 정규 표현식을 활용하였으며, N이 2일 때의 정규식의 예시는 그림 1과 같다. 어절의 빈도가 도출되면, 어절별 문서의 빈도(DF)를 계산한다. 문서 전체의 개수인 322는 DF값의 분모가 되며 해당 어절이 322개의 문서 중에 몇 개의 문서에 포함되는지의 여부가 분자의 값이 된다.

```
var wordPattern = new Regex(@"\w+");
```

그림 1. N = 1일 때, 정규 표현식 예시
Fig. 1. Normalized equation at N=1

N=1일 때, 불용어를 제외한 수집된 단어의 수는 총 6,372개였으며, 상위 10개 단어의 빈도수는 표 1과 같다. “license”가 7,670개, “work”가 3,978개 “code”가 3,676개로 높은 빈도를 나타냈다. 대부분의 단어들이 라이선스와 관련된 단어인 것으로 유추가능하다.

표 1. N=1일 때, TF값이 상위 10개인 단어
Table 1. Upper 10 words at N=1

Words	**TF**
license	7,670
work	3,978
code	3,676
software	2,727
rights	2,499
terms	2,294
source	1,817
version	1,769
copyright	1,734
original	1,664

N=2일 때는 출현한 총 단어의 수는 20,513개였으며, 상위 10개의 단어별 빈도수는 표 2와 같다. “source code”가 680번으로 가장 많이 출현하였고, “covered code”가 535번, “creative commons”가 341번으로 많이 출현하였다. 마찬가지로 라이선스의 특징을 나타내는 단어들이 많이 출현하였다.

표 2. N=2일 때, TF값이 상위 10개인 단어
Table 2. Upper 10 words at N=2

Words	**TF**
source code	680
covered code	535
creative commons	341
public license	315
original code	288
derivative works	286
initial developer	259
original work	216
copyright notice	214
rights granted	193

N=3일 때는 불용어를 허용하지 않았을 때 총 단어의 수는 3,512개였다. 가장 높은 빈도수를 보이는 단어는 66회 출현한 “general public license”였다. 영어의 특성상 불용어를 제외한 3어절이 연속적으로 오는 경우는 드물기 때문에 N이 3일 때에 한해서는 불용어를 1어절 허용하였다. 불용어를 허용하였을 때는 총 단어의 수가 12,992개였으며, 빈도수가 높은 상위 10개의 단어는 표 3과 같다. 가장 많이 출현한 단어는 각각 263회, 226회, 195회 출현한 “the source code”, “the original code”, “term and condition”이다.

표 3. N=3일 때, TF값이 상위 10개인 단어
Table 3. Upper 10 words at N=3

Words	**TF**
the source code	263
the original code	226
terms and conditions	195
including without limitation	148
all rights reserved	147
the rights granted	141
the covered code	138
or consequential damages	135
the initial developer	135
the original work	128

3.3 TF-IDF 분석

본 절에서는 3.2에서 도출한 TF값에 IDF값을 곱한 TF-IDF값이 큰 상위 10개의 단어들을 도출하였다. N=1일 때, TF-IDF값이 높은 상위 10개의 단어들은 표 4와 같다. TF값의 상위 10개 단어들 중 “software”, “rights”, “source”, “version”, “copyright”, “copyright”가 TF-IDF값의 상위 10개 단어에 포함되지 않았으며, “covered”, “licensor”, “contributor”, “initial”, “licensed”, “developer”가 포함되었다. TF-IDF값이 높은 단어들도 “work”, “covered”, “licensor”로 TF값의 상위 단어와 상이하였다.

표 4. N=1일 때, TF-IDF값이 상위 10개인 단어
Table 4. Upper 10 words at N=1, TF-IDF

Words	TF	IDF	TF-IDF
work	3,978	0.565	2247.830
covered	1,351	1.444	1950.598
licensor	1,442	1.344	1937.666
license	7,670	0.206	1581.608
contributor	1,592	0.915	1456.265
code	3,676	0.395	1450.749
terms	2,294	0.576	1321.469
initial	658	1.823	1199.737
licensed	1,072	1.038	1113.115
developer	501	2.219	1111.821

N=2일 때의 TF-IDF 분석 결과는 표 5와 같다. TF대비 TF-IDF의 결과에서 새롭게 도출된 단어는 “licensed material”, “licensed product”, “covered software”, “digitally perform”이었으며, TF-IDF값이 가장 높은 단어는 “covered code”였다. TF값이 가장 높은 단어였던 “source code”는 TF-IDF값이 상대적으로 작았는데, 이는 빈도수는 가장 높았지만 문서의 아이덴티티를 가장 높게 반영할 만한 단어는 아닌 것으로 해석가능하다.

표 5. N=2일 때, TF-IDF값이 상위 10개인 단어
Table 5. Upper 10 words at N=2, TF-IDF

Words	TF	IDF	TF-IDF
covered code	535	2.442	1306.656
creative commons	341	2.278	776.813
original code	288	2.407	693.290
initial developer	259	2.639	683.516
licensed material	142	3.829	543.667
licensed product	104	4.676	486.298
covered software	119	3.983	473.952
original work	216	1.968	425.064
source code	680	0.582	395.484
digitally perform	120	2.884	346.102

N=3일 때의 상위 10개의 어절에 대한 TF-IDF 결과는 표 6과 같다. TF-IDF값이 높은 단어는 “the original code”, “the initial developer”, “the covered code” 순이었다. N=2일 때와 비슷하게 가장 빈도가 높았던 “the source code”는 TF-IDF의 값이 상대적으로 저조했다. “this public license”과 “or publicly digitally”, “the licensed material”, “indemnity or liability”가 TF-IDF 값이 상위인 단어에 새롭게 출현하였다.

표 6. N=3일 때, TF-IDF값이 상위 10개인 단어
Table 6. Upper 10 words at N=3, TF-IDF

Words	TF	IDF	TF-IDF
the original code	226	2.407	544.040
the initial developer	135	2.639	356.273
the covered code	138	2.556	352.683
this public license	98	3.577	350.578
the source code	263	1.169	307.547
the original work	128	2.219	284.058
or publicly digitally	90	2.884	259.576
the licensed material	63	3.983	250.916
terms and conditions	195	1.102	214.836
indemnity or liability	97	2.164	209.872

4. 라이선스 파일 추출 및 식별 실험

4.1 실험 데이터 Set

오픈소스 패키지 파일 내에서 라이선스 정보를 추출/식별하기 위해 표 7과 같이 4가지의 케이스를 가정하여 데이터를 생성하였다.

표 7. 라이선스 위치를 고려한 실험 데이터 설정
Table 7. Data setup at license location

Case	설정 내용
Case 1	소스코드 파일과는 별도의 파일로 존재하는 경우
Case 2	소스코드 파일 내에 삽입되어 있는 경우
Case 3	Case 1의 라이선스 정보가 일부 손상된 경우
Case 4	Case 2의 라이선스 정보가 일부 손상된 경우

Case 1은 License.txt, Readme.txt와 같이 라이선스 정보가 별도의 파일로 존재하는 경우를 가정하였고, Case 2는 여러 소스코드 파일 중 하나의 소스코드 파일에 라이선스 정보가 삽입되어있을 때를 가정하였다. Case 3과 Case 4의 경우는 각각 Case 1과 Case 2의 라이선스 정보가 손상된 경우를 가정하였다.

라이선스는 Apple Public Source License 1.2, BSD 3-Clause No Nuclear License 2014, Netizen Open Source License 1.1, XSkat License로 총 5개의 라이선스에 대해 라이선스 추출 및 식별 실험을 진행하였다.

4.2 실험 결과

실험은 여러 파일 내에서 라이선스 정보를 갖는 파일을 추출(Extraction)하는 과정과 추출한 라이선스가 어떤 정보를 갖는지 식별(Match)하는 과정으로 이루어졌다. 실험 결과는 표 8과 같다.

표 8. Case별 라이선스 추출 및 식별 실험 결과
Table 8. Experiment results to extract license by case

Case	License Name	Extraction	Match
Case 1	Apple Public Source License 1.2	O	O
	BSD 3-Clause No Nuclear License 2014	O	O
	Netizen Open Source License	O	O
	SNIA Public License 1.1	O	O
	XSkat License	O	O
Case 2	Apple Public Source License 1.2	O	△
	BSD 3-Clause No Nuclear License 2014	O	O
	Netizen Open Source License	O	O
	SNIA Public License 1.1	O	O
	XSkat License	O	O
Case 3	Apple Public Source License 1.2	O	△
	BSD 3-Clause No Nuclear License 2014	O	O
	Netizen Open Source License	O	O
	SNIA Public License 1.1	O	O
	XSkat License	O	X
Case 4	Apple Public Source License 1.2	O	△
	BSD 3-Clause No Nuclear License 2014	O	O
	Netizen Open Source License	O	O
	SNIA Public License 1.1	O	X
	XSkat License	O	X

패키지 내의 대부분 파일들은 소스코드 파일이며, 전문으로 이루어진 라이선스 파일과는 달리 언어별 연산자로 이루어져 있기 때문에 쉽게 구분이 가능하다. 추출 실험의 경우 모든 Case의 라이선스에 대해서 추출에 성공하였다.

추출한 라이선스는 보유하고 있는 322개의 라이선스와 코사인 유사도 측정을 통해 대조하였다. 라이선스 파일이 패키지 내에 별도의 파일로 존재하는 Case 1의 경우는 모든 라이선스가 정확히 식별되었다. 소스코드 내에 라이선스 정보가 들어있는 Case 2의 경우는 Apple Public Source License 1.2 라이선스를 정확히 식별하지 못했다. 실험결과에서 △는 해당 라이선스를 식별하였지만, 버전 정보는 정확히 식별하지 못한 경우를 뜻한다. Apple Public Source License는 현재 1.0, 1.1, 1.2, 2.0이 존재하는데, 몇 개의 항목이 추가, 변경되었을 뿐 크게 변경되지 않았는데, 이를 정확하게 식별하지 못하였다. 손상된 라이선스 정보가 별도의 파일로 존재하는 Case 3의 경우는 Case 2와 동일하게 Apple Public Source License의 버전 정보를 식별하지 못하였다. 또한 XSkat License에 대해서는 정확하게 식별하지 못하였는데, 문서의 특성을 파악 가능할 정도로 해당 라이선스의 전문이 길지 않기 때문이었다. 이는 손상된 라이선스 정보가 소스코드 파일 내에 삽입되어 있는 Case 4도 동일하였다. Case 4에서는 SNIA Public License 1.1 라이선스 또한 식별하지 못하였는데, 이는 코사인 유사도를 분석하는 단어 개수를 상수로 지정했기 때문이다. 코사인 유사도를 분석하는 단어의 개수는 비교할 문서의 수가 많아질수록 탐색속도에 영향을 미치기 때문에 이를 제어하였는데, SNIA Public License 1.1과 같이 상위 단어가 큰 특징을 보이지 않는 라이선스에 대해서는 식별하지 못하는 결과를 보였다.

5. 결론 및 향후 연구방향

오픈소스 소프트웨어는 초기(initial) 개발자가 원본(original) 코드를 공개하면, 다른 개발자(contributor)들이 코드를 수정하여 재배포하게 된다. 라이선스를 준수해야하는 오픈소스 소프트웨어는 2차, 3차 배포과정에서 라이선스 정보가 누락, 훼손, 변경, 충돌되는 경우가 발생하는데 이는 저작권 침해로 인한 법적 분쟁으로 이어지기도 한다. 본 연구는 오픈소스 소프트웨어 패키지에서 라이선스 파일을 식별하여, 개발자가 준수해야할 라이선스 조항을 인지해주는 것을 목적으로 한다.

TF-IDF분석을 통해 라이선스 문서가 갖는 특징을 추출하였고, 코사인 유사도 분석을 통해 유사도가 높은 파일을 라이선스 문서라고 판별하였다. 라이선스 파일의 위치(별도 파일/ 소스코드 파일 내)와 손상 정도(전문/일부손상)를 설정하여 라이선스 파일을 추출/식별해본 결과, 모든 경우에 대해서 추출은 성공하였다. 다만 추출한 라이선스가 어떤 라이선스인지 식별하는 과정에서 버전 정보를 정확히 식별하지 못하거나 짧은 전문을 갖는 라이선스에 대해서 잘못 식별하는 경우도 발생하였다. 라이선스 문서에 대해 TF-IDF 분석한 뒤 이를 기반으로 유사도를 측정하는 지금의 단계에서 나아가 유사도가 상위권인 문서들에 대해 전수조사를 진행하는 등의 세밀한 작업을 보완한다면 라이선스 문서 식별 정확도가 높아질 것으로 기대된다.

Acknowledgment

본 연구는 문화체육관광부 및 한국저작권위원회의 2016년도 저작권기술개발사업의 연구결과로 수행되었음

참 고 문 헌

[1] 한국저작권위원회, “오픈소스SW 라이선스 분쟁 대응 방안 가이드”, 2013.
[2] Federal Financial Institutions Examination Council, “Risk Management of Free and Open Source Software”, 2004.
[3] 미래창조과학부, “15년 공개SW 개발지원사업 안내서”, NIPA, 2015.
[4] R. Gobeille, “The FOSSology Project”, Proceedings of the international working conference on Mining software repositories, 2008.
[5] Ohloh, https://www.openhub.net/p/ohloh, 2016.
[6] Open Source License Checker, https://sourceforge.net/projects/oslc, 2016
[7] T. Tuunanen, J. Koskinen, T. Kärkkäinen, “Automated Software License Analysis”, Automated Software Engineering, 2009.
[8] Blackduck, http://blackducksoftware.com, 2016.
[9] Palamida, http://www.palamide.com, 2016.
[10] Protecode, http://www.protecode.com, 2016.

저 자 소 개

윤호영(Ho-Yeong Yun)

2012년 한성대학교 산업경영공학과 학사 졸업.
2016년 연세대학교 정보산업공학과 박사과정 수료
2016년-현재 엘에스웨어(주)
<주관심분야 : 최적화 이론, 알고리즘>

조용준(Yong-Joon Joe)

2011년 큐슈대학교 전기정보공학과 학사 졸업
2016년 큐슈대학교 정보학과 박사과정 수료
2016년-현재 엘에스웨어(주)
<주관심분야 : 게임이론, 분산 최적화 이론, 인공지능 >

정병옥(Byung-Ok Jung)

2007년 대전대학교 컴퓨터공학 석사
2006년-2016년 ㈜디지캡
2016년-현재 ㈜엘에스웨어

<주관심분야 : 클라우드 보안 서비스, 빅데이터, 응용 보안>

신동명(Dong-Myung Shin)

2003년 대전대학교 컴퓨터공학과 박사
2001년 - 2006년 한국정보보호진흥원 응용기술팀 선임연구원
2006년 - 2014년 한국저작권위원회 저작권기술팀 팀장
2014년 - 2016년 한국스마트그리드사업단 보안인증팀 팀장
2016년 - 현재 엘에스웨어(주) 연구소장/이사
<주관심분야 : 오픈소스 라이선스, 시스템/네트워크보안, SG인증/보안, SW취약점 분석・감정>