Journal of the Korea Convergence Society
Vol. 9. No. 4, pp. 31-45, 2018

ISSN 2233-4890
https://doi.org/10.15207/JKCS.2018.9.4.031

사물인터넷(IoT)발전을 위한 소스프로그램 보호방안 연구: 프로그램의 보호와 유사표절 연구

이종식
성균관대학교 인터랙션사이언스학과

A Study of protective measures of the source program for the development of the Internet of Things (IoT): Protection of the program as well as plagiarism research

Jong-Sik Lee
Department of Interaction Science, Sungkyunkwan University

요 약 최근 인터넷 기술이 급격히 발전하면서 컴퓨터 관련 기술이 함께 발달하면서 컴퓨터와 스마트 디바이스를 둘러싼 소프트웨어 분쟁이 심화 되고 있고 또한 각국의 정책적으로 소프트웨어 개발에 소리 없는 전쟁이 진행되고 있는 상태이다. 특히 최근 빅데이터와 사물인터넷 IoT (Internet of Things) 인터넷 기반의 관련 산업이 급격히 발전하고 있으며 여기에 사용된 java 와 C 언어,xcode의 오픈소스 기반의 소스프로그램을 만들고 개발하기 위하여 수많은 시간을 투자하여 개발이 이루어지고 있다. 기본적으로 소프트웨어의 침해를 방지하고자 보안 방법이 존재 하지만 생각보다 쉽게 복제되는 것이 현실이다. 이런 상황일수록 소스프로그램의 라이센스에 대한 원천 권리자의 권리보호 매우 중요한 사항이다. 물론 제작자의 원천기술을 당연히 보호되어야 함이 마땅하나 너무 지나친 보호는 프로그램으로 인한 산업발전과 문화발전에 저해 될 수 있는 소지가 존재한다. 따라서 본 연구는 IoT 응용프로그램에 대한 유사표절을 데이터 마이닝 기법으로 연구하였으며 이는 프로그램 복제로 인한 창작자의 프로그램을 보호하고 나아가 프로그램으로 인한 개인정보유출과 침해에 대한 방안을 제안하였다.

주제어 : 사물인터넷, 프로그램보호, 공정이용, 소프트웨어저작권법, 프로그램 표절

Abstract Recent dramatical development of computer technology related to internet technology intensifies the dispute over software of computer or smart device. Research on software has been flourished with political issuing of fierce competition among nations for software development. Particularly industrial growth in ethernet based big data and IoT (Internet of Things) has promoted to build and develop open source programs based on java, xcode and C. On these circumstances, issue on software piracy has been confronted despite the basic security policy protecting intellectual property rights of software and thus it is of substantial importance to protect the rights of originality of source program license. However, the other issue on source technology protection of developer is the possibility of hindrance to advancement in industry and culture by developing programs. This study discuss the way of enhancing legal stability of IoT application program development and reinforcing precision in inspection of program plagiarism by analyzing the source programs with newly introducing text mining technique, thus suggests an alternative protective way of infringement of personal information due to duplicating program.

Key Words : IoT, Program protection, Fir Use, Software program copyright, program plagiarism

*Corresponding Author : Jong-Sik Lee (jongsic@skku.edu)
Received February 2, 2018 Revised March 15, 2018
Accepted April 20, 2018 Published April 28, 2018

I. 서론

우리나라는 최근 모든 컴퓨터와 모든 사물을 연결하는 사물인터넷 Iot (Internet of Things) 기술이 급격히 발전하면서 인터넷은 모든 사물과 사물 간의 연결이 가능하도록 하면서 기존의 개인정보 보안과 함께 연동하고 제어할 수 있는 소프트웨어 분쟁이 심화 되고 있으며 각 나라마다 IoT 사물 인터넷의 기술의 기반인 소프트웨어 개발이 활발하게 진행되고 있는 상황이다[1]. 특히 최근 M2M (Machine-to-Machine)와 사물인터넷 IoT (Internet of Things) 기반의 (Industrial Internet of Things, IIoT) 관련 산업이 급격히 발전하고 있고 여기서 거치치 않고 IoE (Internet of Everything)로 까지 확장되고 있으며, 향후 경제적인 파급효과가 상당할 것으로 예상된다[2]. 현재 IoT 는 물리적으로 존재하는 구동 장치(Actuator) 혹은 센서 등에 ICT 기술을 접목하고 네트워크에 연결시켜 사람과 상호작용 없이 상황에 따른 서비스를 제공하는 무선센서망(Wireless Sensor Network, WSN), IoT 개념은 물체를 네트워크에 연결 함으로써 독립적으로 분산되어 있던 서비스들을 연결하여 새로운 서비스 창출하고 있다[3]. 이런 H/W 와 S/W 의 영역에서 많은 기술이 사용이 되는데 실제 외부에 보이지 않는 S/W 기술의 역할이 매우 중요한 가운데 있다. IoT 의 많은 어플리케이션은 java 와 C 언어 그리고 xocde 기반으로 소프트웨어 프로그램을 만들어지고 있으며[4] 또한 이를 응용하여 다양한 분야에 개발이 예상된다. 앞으로 2020년이면 500억개 이상의 다양한 사물들과 연동하고 제어함으로 다양한 분야에 적용된 새로운 융복합 시대가 올 것으로 기대되며[2] IoT 기술을 표준화 하기위해 EU FP7, IETF, IEEE, 3GPP 등 다양한 표준기구 및 연구단체에서 IoT에 대해 연구를 진행하기 시작하였고,[5] 현재는 많은 표준화 조직 및 기구들이 관련 표준을 개발하고 있다[6]. 현재의 추세는 IoT의 발전을 위하여 오픈 소스(open source)의 소프트웨어 혹은 하드웨어를 저작자의 권리를 지키면서 원시 코드를 누구나 열람할 수 있도록 하고 있는데 실상은 그렇지 못한 상황이다. [7-9] 물론 해당 산업에 방해되지 않는 한도 내에서 저작자의 권리를 지키기 위하여 많은 어려움이 존재하는데 그 해결방법은 물리적으로 하드웨어 보안 방법[10]이 존재 하지만 생각보다 쉽게 복제되는 것이 현실이며 기술이 나아지면 나아질수록 IoT 자체 소프트웨어 침해와 나아가서 IoT 을 이용한 많은 개인정보와 보안 침해가 늘어날 것이다[11]. 따라서 본 연구에서는 IoT 의 향후 상용화되기 위하여 적절한 지식재산권법에서의 저작자의 프로그램 보호는 IoT 의 발전에 중요한 기반이 될 것이며 또한 IoT 프로그램에 대한 텍스트 마이닝 기법으로 표절 검사를 수행하였으며 나아가 프로그램 저작자의 권익과 나아가 IoT 발전에 따른 심각한 개인정보 유출에 좀 더 방어 할 수 있을 것이다.

1.1 IoT 구성요소

사물인터넷에 대하여 다양한 분야에서 여러 정의를 내어 놓고 있는 상태이다. 기존에 ITU(2005)에서 "ITU Internet Report"를 통하여 IoT 의 개념을 소개 했는데[7] 기존의 통신 기술이 사물과 사물 그리고 사물과 사람 간의 언제(Anytime) 어디서나(Anyplace) 무엇이든지(Anything) 서로 주고받은 개념이라고 할 수 있다[12,13].

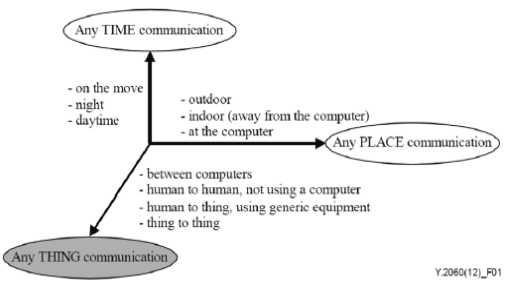

Fig. 1. Basic Concept of IoT(Internet of Things) [7,12,13]

IoT는 ITU-T World Summit on the Information Society 에서 Fig. 1과 같은 IoT의 구성 모델을 제안하고 있으며 주변의 다양한 스마트 디바이스와 통신이 가능하도록 구성하여서 사물과 사물간의 통신을 위해서 센서와 단말 사이에는 많은 정보가 전송이 되어야하며 전송된 정보를 처리하기 위해서는 다양한 소스프로그램으로 만든 어플리케이션이 존재한다. 본 연구에서는 스마트폰을 중심으로 한 IoT 기반의 통신을 위해서 필요한 IoT 통신을 위한 응용프로그램을 살펴보고 분석 연구하였다. Fig. 2는 현재 IoT 환경을 구축하기 위해 제공되는 오픈소스 형태를 보여주며 오픈소스 형태의 구성은 오픈소스 소프트웨어 (OSS : Open Source Software)와 오픈소스 하드웨어 (OSHW: Open Source Hardware) 그리고 Open Data , Open API 등 다양한 오픈소스영역과 함께 이루어지고 있으며 기준으로 하기와 같이 구성이 되어 진다.

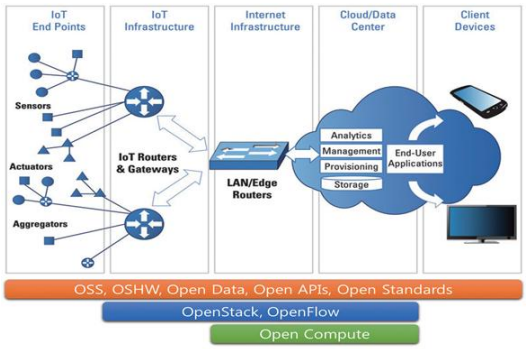

Fig. 2. Open source based things Internet components[16]

현재 오픈소스 소프트웨어 자체에 대해서는 자유로운 배포(Free Redistribution) 가능하며 소스코드 공개(Source Code Open)와 2차적 저작물 허용(Derived Works)을 해야 되지만 원작자의 소스코드의 변경 하지 못하며 라이선스 적용상의 동일성 유지가 되어 한다[17]. 또한 프로그램 저작자의 권리를 보장하기 위해 저작자 인격권을 보장 해 줘야 하는데 저작인격권이란 "저작자가 자신의 저작물에 대해 갖는 정신적·인격적 이익을 법률로써 보호 받는 권리"라고 할 수 있으며, 저작권법에서는 이를 공표권, 성명표시권, 동일성 유지권의 세 가지로 나누어 규정하고 있다. 또한 저작재산권과는 달리 그 성질상 일신 전속 권으로 소스코드 수정 제한(Integrity of the Author's Source Code) 과 공표권과 성명표시권 그리고 라이선스적용상의 동일성 유지(License must not be specific to a product)을 유지하여야 한다[17,18]. 따라서 원천 소스프로그램에 대한 복제도 금지되어야 하지만 프로그램에 대한 법적인 저작 인격권도 보장이 되어야 하며 IoT 관련하여 저작권법에서 오픈소스코드와 오픈소스코드가 아닌 소스코드에 대하여 좀 더 세부적인 규정이 저작인격권을 깊게 다룰 필요가 있으며 오픈소스 소프트웨어(OSS)에 대하여 원저작물을 번역·편곡·변형·각색·영상제작 그 밖의 방법으로 작성한 창작물」이 "2차적저작물(저작권법 제5조 제1항) 로서 보호를 받을 수 있는데 프로그래머가 OSS를 창작하고 다수의 프로그래머가 개작(modify)하는 형태로 작성되는 OSS는 독창성(originality)의 추가, 창작성의 부여가 있는 한 2차적 저작물일 것이다. 따라서 OSS 라고 하여 무조건 자유로이 배포 이용하는 것이 아니라 법적인 테두리 내에서 유지 보호 할 부분도 존재하며 기존 미국의 판례 Jacobsen v. Katzer 사건[18]에서는 오픈소스 라이선스에 있어서 저작인격권 규정을 적극적으로 이용할 수 있는 가능성을 Jacobsen 판결이 시사하고 있는 점은 매우 주목할 만하다. 따라서 IoT 및 향후 소프트웨어 분쟁에 관련하여 오픈소스와 소스프로그램에 관련된 세부적인 법적인 조항이 필요한 상태이다.

1.2 IoT의 종류와 보안의 필요성

IoT 의 모든 기술들은 다양한 기술과 프로토콜로 구성 되어 있으며[19] 수많은 센서와 디바이스, 미들웨어 프렛폼, 서비스 플렛폼과 유기적으로 결합되어 있다.

Table 1. Types of Security Attacks in IoT Environments

Security Vulnerabilities and Attack Types	Subject areas on IoT
Worm and Virus	IoT communication / network, device, gateway platform, application service
DoS and Distributed DoS	IoT communication / network
An unauthorized approach	IoT devices, gateways, platforms, Application Service
Unpatched Systems OS / OS Security Vulnerability	IoT devices, gateways, platforms, Application Service
Improper use of Antivirus software	IoT platform, application services
Improper use of firewalls	IoT communication / network
Unauthorized service access	IoT Application Service
Replication attack	IoT devices, gateways
Unauthorized I / O access	IoT devices, gateways, platforms, Application Service
Improper System Logging	IoT platform, application service
Setting errors and mistakes	IoT devices, gateways, platforms, application services
Confidentiality / integrity attack	IoT communication / network, device, gateway, platform, application service
Insecure passwords	IoT devices, gateways
Unprotected firmware	IoT devices, gateways
Privacy Invasion	IoT platform, application service

※ J S Choi, & H W Kim (2013)[14][15]

또한 사용자 인터페이스 측면에 항상 노출이 되어 있는 상태이며 많은 사물과 많은 사람을 연결하면 더욱 더 보안의 취약성이 커질 수밖에 없는 상황이다[20]. Table. 1은 IoT 정보 공유 시에 존재 할 수 있는 보안의 취약성의 유형이다. IoT의 효과적인 환경을 구축하기 위해서는 가장 대표적인 프라이버시 침해 문제를 거론하지 않을

수 없다[21]. 현재 IoT 서비스를 하기 위해서는 적절한 프라이버시 침해 대응 기법을 제공하지 않는다면 해당 IoT 서비스를 사용할 수 없을 정도로 심각한 프라이버시 침해 문제가 발생할 수 있으며[22] 특히 국내에서는 2011년 9월 30일부터 강화된 개인정보보호법이 시행되었기 때문에(실제는 2012년 10월 1일부터 시행됨. 1년 동간의 유예 기간을 둠) IoT 서비스 개발자 등, IoT 관련 분야 연구/개발자들이 프라이버시 침해 문제에 대해 적극적인 대응책을 세우지 않으면 IoT 서비스 활성화/상용화는 불가능한 상황으로 정체되어 있을 것이다[11]. 따라서 우리는 현재 가장 기초적인 스마트폰과 사물 사이의 통신의 프로그램에서 적절한 정책과 대응이 필요한 상황이다.

1.2.1 홈 IoT (HIoT)

현재 사물인터넷은 스마트 홈 영역은 홈 IoT 로 발전되고 있는데 이는 주거인의 생활을 돕기 위한 서비스가 이미 홈네트워크 또는 홈오토메이션 이라는 명칭으로 소개되고 있으며 주거 생활에 필요한 통신, 가전 교육, 보안, 의료, 엔터테인먼트 등 가정에 필요한 다양한 형태의 서비스가 예상 된다[23]. 하지만, 기존에 기술적 또는 산업적으로 한계를 보이며 실제생활 속에 보급이 되지 않고 있었으나 최근 IoT 기술의 보급으로 인해서 현실화 가능성이 높아지고 있다[24]. 또한 최근 빅데이터 그리고 IoT, SNS를 결합한 융합제품이 선보이고 있다[60]. 그러나 홈 IoT는 몇 가지 문제점을 가지고 있는 상태인데 위하여 법적인 프라이버시문제와 기술적인 방어가 필수적으로 필요한 상태이다. 살펴보면 홈 IoT 기술은 사람과 사물이 네트워크로 연결되어 있으며 사물과 사물이 네트워크로 연결되어 있으며 자동으로 많은 데이터들이 처리되게 되는데 이런 데이터들이 위험에 노출이 되어 있으며 또한 개인정보 노출 위험과 각종 기기 오작동 위험이 항상 존재한다. 특히 최근 스마트폰의 결재기능등의 위험 노출도 대두되고 있다[61]. 따라서 만약 홈 IoT는 바이러스 침투와 같은 외부의 부적절한 침입을 통해서 홈 IoT 시스템의 특정 디바이스가 감염되거나 홈 네트워크 전체가 감염되는 경우 주거시설에 대한 보안이 심각 하게 피해를 입을 수 있는 상황이다. 또한 사용자의 정보(개인정보, 사용정보, 개인 선호 정보 등)을 수집하기 때문에 프라이버시 문제가 항상 존재한다. 따라서 실 생활에 아주 편리한 부분도 존재하지만 또한 많은 보안 상 문제점을 가지고 있는 상태이다.

1.2.2 Industrial IoT (IIoT)

현재 산업사물인터넷(IIoT)는 다양한 ICT 기술과 무선센서네트워크 기술과 결합하여 산업제어 분야에 많은 성과와 혁신을 이어가고 있다 하지만 산업사물인터넷(IIoT)는 중요한 보안 문제에 놓여있다[25]. 이는 국가 산업에도 막대한 지장을 초래함으로 그 피해는 막대한 경제적인 손해와 국가적 혼란이 야기 될 수 있다. 그래서 IoT 프로그램의 복제에 대한 강력한 법적인 보장과 기술적인 보안이 필수적으로 필요한 상태이다. 또한 보안을 위해 여러 가지 방법이 존재 하지만 특히 최초에 저작자가 만든 독창적인 IoT 프로그램이 오픈소스와 함께 자유로이 복제가 가능하다면 1차적으로 해킹에는 무방비 상태이다. 그래서 산업 IoT 보안을 해결하기 위해서는 개선된 방식의 인증이 필요하며 비정상적인 신호나 해킹에 대하여 미래 예측 할 수 있는 시스템이 필요한 상태이다.

1.3 IoT의 보안 관련 동향

최근 랜섬웨어 범죄 집단과 악성코드 유포 집단 간 경쟁 심화 되면서 데이터를 인질로 금전을 요구하는 랜섬웨어는 러시아어권 지역에서 시작돼 서유럽, 미국, 캐나다, 호주, 유럽 및 아시아 지역으로 확산되고 있다[26]. 수익을 올릴 수 있다는 점 때문에 랜섬웨어의 규모는 더 커질 것으로 예상되고 있는 가운데, 2016년에는 랜섬웨어 범죄 집단이 기존의 악성코드 유포 집단과 갈등을 일으킬 가능성이 예상되며 2016년 보안 이슈 전망을 소개하면서 가장 먼저 IoT 기기가 폭발적으로 증가하고 있지만 보안 위험은 간과하고 있다고 지적하고 IoT 기기에 보안을 내재화하는 문제가 아주 중요해질 것이며 반드시 개발 초기부터 IoT 보안을 적용해야 한다고 설명하고 있다[27]. 또한 생체인식 보안 본격화 되면서 최근 2년간 생체인식 기술 활용이 급증하였으며 주요 기업들을 중심으로 디지털 기기 자체의 새로운 센서나 파이도(FIDO), 터치아이디(TouchID)와 같은 생체인식 인증 체계를 도입함에 따라 앞으로 생체인식 활용은 더욱 늘어날 것으로 전망이며 생체인식 보안의 활성화로 개인 사용자 입장에서 보안은 한층 강화되고, 디지털 기기의 잠금 해제나 구매·결제의 편의성은 크게 높아지는 대신 인터넷 보안에 대한 연구는 더 필요한 상황이다.

2. 관련 연구

2.1 IoT 관련 법적 연구 및 동향

2.1.1 컴퓨터 프로그램보호의 법적인 동향

컴퓨터 프로그램[1]은 사용자의 요구에 따라 많은 변화를 거듭하고 있다. 컴퓨터 프로그램은 오랜 시간 동안 저작권법으로 보호가 되어오고 있으며 저작권으로 보호는 형식이 필요 없고 장기간 보호가 되고 비용이 작게 소요되는 장점이 있으며 아이디어와 기능자체를 보호하지 못하고 오직 표현만을 보호함으로 프로그램의 표현을 밝히는데 한계가 있는 상태이나 일부 프로그램 소스코드를 효과적으로 보호 할 수 도 있다. 또한 이용자의 요구에 따라 프로그램은 더 높은 창의력과 아이디어를 요구하게 된다. 그러나 표현이 이루어지지 않으면 보호가 될 수 없는 부분이 항상 존재하게 된다. 또한 프로그램은 누구나 쉽게 복제할 수 있고 이런 불법 복제로 인하여 개발자의 노력을 상실 할 수 있는 여지가 충분히 있을 수 있는 상황이다. 또한 프로그램 자체는 외부에 도용 될 가능성이 충분히 있는 상태로 기존에 소스코드의 노하우를 법률로서 완전히 방어하기 위해서는 한계가 있는 상태이다. 프로그램은 그 가치의 원천이 되는 디자인 노하우가 풍부하다. 어렵게 개발된 디자인 기술혁신의 대부분은 프로그램의 작용 중에 명백하게 노출될 수 있다[35]. 우선 저작권법에서 컴퓨터 프로그램의 요건을 정의하고 있고 요건에 충족하면 저작권법으로 보호를 받을 수 있으나 프로그램의 저작물성과 특허 물성여부는 세부적으로 검증할 필요가 있는 상태이다. 우리는 먼저 컴퓨터 프로그램을 알기에는 컴퓨터의 전반적인 구조를 생각해 볼 필요가 있다. 보통 컴퓨터 프로그램을 구현하기 위해서는 프로그램 상에 알고리즘, 데이터 구조, 제어구조 등 내부구조에 들어 있으며 프로그램의 이런 구성요소를 다르고 새롭게 함으로 인하여 창의적인 아이디어와 사상이 들어간다. 컴퓨터를 프로그램은 시스템 프로그램과 응용 프로그램으로 나뉘며 시스템 소프트웨어란. 하드웨어의 동작을 지시하고 제어하는 모든 종류의 프로그램으로서 컴퓨터 시스템의 작업수행을 제어하고 응용 소프트웨어의 수행을 지원한다. 응용 소프트웨어는 사용자가 특정한 업무의 처리를 위해 작성한 프로그램의 집합을 말하며 컴퓨터 시스템을 이용하여 사용자각자가 당면한 분야의 문제를 해결하기 쉽도록 도와주는 역할을 하며 [28,36] IoT는 사용자의 요구에 따라 소프트웨어의 진화는 계속될 것으로 예상한다.

1) 저작권법 제2조 제16호 "컴퓨터프로그램저작물"은 특정한 결과를 얻기 위하여 컴퓨터 등 정보처리능력을 가진 장치(이하 "컴퓨터"라 한다) 내에서 직접 또는 간접으로 사용되는 일련의 지시·명령으로 표현된 창작물을 말한다.

2.2 소프트웨어 프로그램의 특허 및 저작권 보호

우리법원은 컴퓨터프로그램보호법상 프로그램이라 함은 특정한 결과를 얻기 위하여 컴퓨터 등 정보처리능력을 가진 장치 내에서 직접 또는 간접으로 사용되는 일련의 지시·명령으로 표현된 것을 말하고 프로그램저작권은 프로그램이 창작된 때로부터 발생한다고 명시하고 있다[37]. 또한 법원은 일부 프로그램에 대하여 창작성을 인정하는데 특히 서체 프로그램에 대하여 창작성을 인정하고 있다[38]. 또한 글꼴파일이 컴퓨터프로그램보호법상 보호받을 수 있는 프로그램에 해당되지 않는다고 한 사례 있으며 이유는 표현이 동일하다고 하여 저작권 침해로 지정하게 되면 아이디어 또는 서체도안 자체를 보호하는 결과가 되므로, 적어도 여러 표현 방식이 가능하고 그 표현된 방식이 다른 방식들에 비하여 독창적인 경우에만 프로그램을 보호해야 됨을 알 수 있고 [39]이는 프로그램의 기준으로 보았을 때 오픈소스에 해당 한다고 볼 수 있을 것이며 특별히 프로그램의 독창적인 저작물성 부분에 대하여 잘 분석하여야 침해나 권리에 대한 대응이 가능 할 것이다. 또한 서체 프로그램을 저작물성의 침해 판단함에 있어서 그 해당 글꼴파일이 어떻게 만들어지고 그 내용이 무엇인지 프로그램과 비교하여 배열방법이나 원시코드가 파악하여 실질적 유사성을 가지고 있는지를 고려하여야 한다[40]. 따라서 글꼴 파일이 창작성이 있는 프로그램인지에 대해서도 논의가 필요한 상태이며[41] 프로그램 파일의 전환 행위 실제 복제침해에 해당되지 않도록 주의가 필요하다. 또한 컴퓨터 프로그램을 설치함에 있어서 시리얼 번호의 복제 및 배포자체는 컴퓨터 저작권 침해로 보지 않고 있다[42]. 따라서 컴퓨터 프로그램에 있어서 결과론적으로 실질적 유사성이 복제 침해에 중요한 기준임을 알 수 있다.

2.3 개작한 2차적 프로그램의 저작권 귀속

기존에 프로그램 자체를 개작하게 되면 2차적 프로그램으로 되는데 이는 원 프로그램 저작자와 위탁된 2차적 개발자 사이에 특약이 없는 한 개작된 프로그램의 창작자에게 저작권이 있다고 할 것이다[43]. 또한 프로그램저작권은 당사자 사이에 계약만으로 전부 또는 일부 양도가 가능하며 원 프로그램을 개작한 2차적 프로그램에 대

하서는 원 프로그램의 동의 여부를 불문하고 2차적 프로그램 작성자에게 귀속된다[44].

2.4 프로그램 저작권의 제한

또한 정해진 장소에서만 복제가 이루어져야 하며 프로그램의 보호법에 제 12조 2항에 복제의 예외 조항을 두고 있는데 '교육법 및 다른 법률의 규정에 의한 교육기관'에서만 복제가 이루어져야 한다. 또한 저작권법에서는 프로그램에 대하여 별도로 적용범위의 제한을 두고 있다[45].

2.5 소프트웨어위탁개발계약의 법적성질 및 권리

프로그램을 개발함에 있어서 소프트웨어의 위탁 개발이 많이 이루어지고 있는 상황이다. 이런 상황에서 계약서상의 개발 과 공급계약의 법적인 성질과 계약 내용을 정확히 하는 것이 중요하며 계약서상의 의미가 명확하지 않는 경우에는 여러 가지 당사자와 간의 조건을 면밀히 검토하여 합리적으로 해석해야 한다[46]. 또한 프로그램의 개발 시에 주문자가 위탁개발사에 주문을 하여 프로그램이 개발이 되면 우리법원은 주문자가 전적으로 기획하고 자금을 투자하고 개발자의 인력만 빌려 개발을 위탁하여 오로지 주문자만을 위한 프로그램에 대해서는 주문자에게 프로그램저작자의 권한을 부여하고 있는 상태이다[47].

또한 컴퓨터 프로그램 저작권은 당사자 사이에 양도계약이 체결이 되면 별다른 철차 없이 양수인에게 이전되는 것이고 저작권 양도의 목적으로 하는 계약이 무효이면 저작권은 처음부터 이전되지 않았다고 보아야 한다. 또한 프로그램의 복제에 대하여 저작권 침해를 엄중히 처벌하고 있으며 프로그램의 창작자에게 권리가 귀속되는 것은 당연한 사례이다[48].

2.6 IoT 사물인터넷의 컴퓨터프로그램 보호

사물 인터넷(Internet of Things, 약어로 IoT)은 각종 사물에 센서와 통신 기능을 내장하여 인터넷에 연결하는 기술을 의미한다. 여기서 사물이란 가전제품, 모바일 장비, 웨어러블 컴퓨터 등 다양한 임베디드 시스템이 된다. 사물 인터넷에 연결되는 사물들은 자신을 구별할 수 있는 유일한 아이피를 가지고 인터넷으로 연결되어야 하며, 외부 환경으로부터의 데이터 취득을 위해 센서를 내장할 수 있다. 현재 IoT 분야에서는 시간과 공간을 초월하여 연결하고 소통하는 문화를 형성하고 있으며 많은 관련 기관들이 관심과 기대가 높은 분야인 것이 사실이다[49]. 하지만 변화를 거듭 할수록 많은 이슈도 존재하는데 특히 네트워크 환경에서 보안 문제가 취약하고 여러 가지 프로그램이 복제 될 수 있음으로 이와 관련된 법제가 기반이 되어야 한다고 여겨진다.

2.7 IoT프로그램의 보호와 공정이용에 관한 연구

IoT 프로그램의 보호와 관련하여 일반적으로 대부분의 프로그램은 오픈소스의 기반으로 IoT 프로그램이 이루어지는데 오픈 생태계는 기술적 측면에서는 원천 소스를 공개함으로써 누구든 큰 비용 없이 다운로드 받아 개선하면서 사용할 수 있기에 적은 초기 개발 비용이 소요되고, 오픈소스 프로젝트를 통해 최신 기술 정보 및 문제점과 해결책을 공유하는 형태로 자유롭게 운영되기에 독점 프로그램에 비해 기술 혁신 속도가 빠르며, 다른 오픈소스 프로젝트들을 함께 활용함으로써 새로운 응용들을 손쉽게 확장 개발 가능할 수 있다는 특징을 갖고 있다. 경제적 측면에서 빠른 기술 발전에 따르는 개발 비용을 절감하고 투자 위험 부담을 나누며, 개발된 결과물을 통해 이익들을 공유할 수 있으며, 협력 모델을 통해 보다 계속되는 혁신이 가능하도록 한다는 점이 가장 큰 특징이라 할 수 있는데 이와 반대로 오픈 소스 만 공개되는 것이 아니라 저작자의 노력의 산물인 프로그램까지 복제되는 경우가 많으며 나아가 개인정보보호에도 심각한 영향을 미친다. 최근 미국에서 판결된 Oracle America, Inc. v. Google Inc. 판결을 알아보면 우리에게 시사하는 바가 크다. 그 이유는 오픈소스를 이용한 프로그램의 저작 물성에 대하여 논하고 있는데 최근 미국 법원에서는 대표적인 소프트웨어에 대하여 저작권으로 보호 할 것인지 그리고 특허로서 그 권리를 보호 해 줄 것 인지에 대하여 관련된 오라클 vs 구글의 판례를 소개 할 수 있다[29]. Oracle v. Google 사건은 특허권과 저작권에 관한 복합적인 소송 판례이며 오라클과 구글의 판례를 통하여 소프트웨어의 오픈소스 SSO가 저작권으로 보호되기 위한 저작물성의 판단에서 물성을 인정하는 것에 의미가 있고 또한 특히 오픈소스SSO의 저작물성을 판단하는 데 있어서도 아이디어·표현 이분법은 여전히 적용하고 표현된 SSO가 아이디어 그 자체인지 아이디어의 표현에 해당하는지의 여부가 판단기준으로 제시되고 있다는 점에서 향후 파기환송 후 법원에서 합체의 원칙, 짧은 어구 원칙,

필수장면이론, 사실상의 표준 등 저작물성 부정의 요건과 침해주장에 대한 항변으로서 공정사용의 원칙 적용기준을 제시할 것으로 예상되며 공정이용과 저작물성의 인정사이에 오라클과 구글의 사이의 논란은 계속 될 것으로 예상된다. 따라서 미국 법원이 소프트웨어의 보호에 있어 특허권으로서의 보호는 제한적으로 해석하는 반면, 저작권으로서의 보호는 전향적으로 그 범위를 확대하는 것으로 판단될 수 있다. 결과적으로, 컴퓨터프로그램은 동작 또는 시현이 '이용자에게 공개되는 표현'이라는 점에서 동작 또는 시현에 있어서도 창작성이 인정되어야 할 필요가 있고, 컴퓨터프로그래밍 기술의 발달로 소스코드의 중요성이 약화되고 있는 한편 SSO의 보호가 중요한 핵심쟁점으로 부상하고 있다는 점을 고려할 때, 일정한 요건 하에서 SSO를 보호할 필요가 있다. 따라서 이런 미국 기준이 프로그램에 대한 세계 각국의 보호에 관한 영향이 클 것으로 생각되며 프로그램에 대한 기능적 저작물의 범위를 벗어나서 저작물성에 대하여 좀 더 심도 깊게 연구할 필요가 있는 상태이다.

2.8 소스프로그램의 특허법 적용에 관한 검토

특허제도의 목적은 기술혁신을 장려하고 기술 이전과 확산을 장려하며 산업 발전 이바지함[2] 에 있는데 또한 독점적 권리를 일정 기간주어서 그 특허를 활용할 수 있게 하고 발명을 증진하며 발명을 통하여 가술의 진보가 일어나며 대중은 특허를 통하여 새롭게 응용하기 위하여 특허 정보를 사용할 수 있다. 컴퓨터 프로그램의 특허제도를 반대하는 반대론자의 기준으로 볼 때에 지나친 독점적 권리로 인한 보호가 지나치게 넓은 경우에는 역효과가 나타날 수 있으며 기술혁신을 제약하기도 하며[50] 최근 빠른 속도를 거듭하는 이동통신과 스마트 폰의 환경 그리고 ICT 환경의 발달 함에 따라 프로그램에 구현된 아이디어를 보호가 많이 필요한 상태이지만 지금의 저작권법은 표현을 기준으로 보호하는 상황으로 프로그램를 적절히 보호하기에는 부족한 부분이 존재한다[51]. 또한 한편으로 옹호론자는 특허제도의 가장 중요한 특징 아이디어와 사상을 보호할 수 있고 넓은 권리를 주장 할 수 있다. 또한 동일한 개념을 기초로 하여 독립적인 프로그램에서도 보호가 이루어지고 기술이 축적 될 수 있으며 창조적인 아이디어의 프로그램이 회사 자산으로 됨으로 회사에는 강력한 자산이 될 수 있다. 또한 수많은 컴퓨터프로그램과 스마트폰 기반 소스프로그램 저작자의 창조적 아이디어와 사상이 내재되는 경우가 많다. 특히 시대가 복잡해지고 경쟁이 치열해질수록 좀 더 참신한 아이디어의 프로그램을 원한다. 시대가 프로그램의 문화도 변화 시키고 있으며 무엇보다 프로그램의 개발자는 기존에 단순한 프로그램을 벗어나 보다 창조적인 형태의 프로그램을 만들고자 노력한다. 이런 지식재산권의 측면에서 특허의 프로그램 적용은 시대에 부응 할 부분도 충분히 가지고 있다고 봐야 할 것이다. 또 다른 한편의 프로그램 특허 반대론자의 주장은 프로그램이라는 것이 점증적, 연속적, 재사용이 프로그램 전반에 걸쳐 이용되고 있다고 주장하며 정해진 형태의 프로그램은 시스템과 네트워크 간에 상당히 제한 되었다고 주장한다. 또한 후속 개발자의 개발의 자유를 막게되고 선택의 폭이 작아지는 역할을 하게 된다. 또한 기존에 특허로 보호된 프로그램이 진보성이나 현실의 변화에 반영하지 못하면 시대에 뒤떨어진 특허로 전략하게 될 것이다. 우리는 여기서 앞에서 제안된 스마트폰의 앱이나 IoT 사물인터넷을 활용한 다양한 네트워크가 앞으로 구성이 될 것이다. 시대가 변하면 변할수록 복잡하고 다양하며 창조적 아이디어와 사상이 특허로서 보호될 필요가 있을 것이며 그러나 너무 지나친 프로그램의 특허화는 후속 연구들의 폭을 좁게 만들 수 있어서 시대와 프로그램의 특성에 따라 신중히 검토할 필요가 있을 것이다[52]. 물론 특허되기 위해서 발명은 일정 요건, 즉 신규성, 진보성, 산업상 이용가능성을 충족시켜야 하는데 컴퓨터프로그램은 일반적으로 복잡하며 프로그램 특허 출원서류는 매우 복잡하며 이러한 요건과 복잡성 때문에 특허제도를 이용하는 비용, 즉 출원하고 권리를 유지하고 방어하는 비용이 업체에게는 많은 부담이 되는 것이 현실이다. 하지만 그렇다고 프로그램의 특허제도를 경시한다면 앞으로의 창조적 아이디어와 사상의 기본 원천적인 프로그램을 지키지 못함으로의 손해가 더 클 것이다. 따라서 다가오는 5G 세대에서는 프로그램의 특허를 스마트폰이나 IoT 기반의 프로그램이 적용 될 수 있도록 보다 세밀하고 현실성에 맞는 조항이 신설되거나 변경이 필요 하다고 여겨진다. 지금의 환경과 기술이 바탕이 되지 않는 법제는 추후 복합적으로 일어날 침해와 저작자나 특허권자의 올바른 권리 보존을 어렵게 할 수 있음을 인지하고 보다 신중하고 시장

2) 저작권법 1조 제1조(목적) 이 법은 저작자의 권리와 이에 인접하는 권리를 보호하고 저작물의 공정한 이용을 도모함으로써 문화 및 관련 산업의 향상발전에 이바지함을 목적으로 한다. <개정 2009.4.22>

환경에 적합한 세부적인 프로그램에 대한 특허의 보완이 필요할 것으로 보인다.

2.9 컴퓨터프로그램의 저작권법적용에 관한 검토

우리 저작권법은 컴퓨터프로그램을 저작물로 규정하고 있으며 컴퓨터프로그램은 본래 어문저작물의 한 유형으로 보호되었으나, 컴퓨터프로그램이 내재하고 있는 특성인 효율성 및 외부적 요인 등에 의하여 표현이 제한될 수밖에 없는 등의 특징으로 인하여 예술적 특성을 기본으로 하는 일반적인 어문저작물의 경우와 동일하게 보호할 수 없는 상태이다. 특히 저작권법은 창작성 있는 표현만을 보호대상으로 하고 단순히 아이디어에 해당하는 부분은 보호대상에서 제외하는 '아이디어 · 표현 이분법'을 기본 원칙으로 채택하고 있으며, 이는 저작권법상의 보호대상을 확정하고, 저작권의 침해 여부를 판단하는 데 매우 중요한 역할을 하고 있다[53]. 이런 조건에 따라서 컴퓨터프로그램을 구성하는 요소들도 표현에 해당하는 요소와 아이디어에 해당하는 요소로 구분하여 주로 저작물성을 평가 하는데 현재 컴퓨터의 프로그램의 대표적인 성향은 단일의 표현형식이 아닌 여러 가지 형태의 표현 등이 결합된 것으로, 소스코드, 목적코드, 마이크로코드, 흐름도, 사용자매뉴얼, 알고리즘 및 스크립트 등이 일체 혹은 다른 형태로 하나의 컴퓨터 프로그램을 구성하고 있는 상황이다. 따라서 컴퓨터프로그램을 저작권으로 보호함에 있어서 아이디어·표현 이분법 체계를 취하고 있으며 이런 가운데 각각의 구성요소별로 저작물성을 따로 비교하여 저작물성을 경우가 늘어나고 있다. 특히 우리나라의 경우 대개 개발자가 직접 작성한 소스코드와 이를 컴파일 하여 생성한 목적코드는 저작권법에서 이론 및 판례를 통하여 명확하게 보호대상으로 삼고 있고 있는 상태이다.

또한 우리나라의 저작권법은 저작물의 이용이 어떤 경우 공정이용에 해당하는지에 관하여는 저작권법 제23조 내지 제38조에서 구체적으로 열거하고 있으나, 저작권 침해 문제와 관련해서는 저작권의 내용을 열거한 후 그에 따른 권리가 침해되었을 경우의 구제수단을 규정하고 있을 뿐이고, 저작권이 침해되지 않은 경우와 침해된 경우를 구분할 수 있는 어떠한 기준도 제시하지 않고 있는 상태이며 이는 완전한 복제의 경우에 있어서 큰 어려움이 없으나, 침해 프로그램에 대하여 어느 정도의 수정이 가해진 경우에는 침해 여부에 대한 판단에 어려움이 있게 되며 구체적인 사안은 법원의 영역으로 옮겨져 판례를 통하여 그 기준과 방법이 논의되게 되는데 일부변경 된 부분에 대하여 유사침해를 밝힐 때 어려움에 존재하는 상태이다. 또한 컴퓨터프로그램 등의 저작권 침해 소송에서 원고가 피고의 저작권 침해를 입증하기 위해서는 원고가 유효한 저작권을 보유하고 있어야 하며 피고가 원고의 저작물성에 의거하여 만들어져야 하며 피고의 저작물이 원고의 저작물과 동일성 또는 종속성 즉, 실질적 유사성을 갖고 있을 것을 입증해야 하는데 본 연구 IoT의 소스프로그램의 실질적인 유사성 비교가 중요한 부분을 차지하고 있으며 이를 연구하기 위하여 텍스트 마이닝 기법을 이용하여 연구하였다.

3. IoT 소스프로그램에 대한 표절 대처 방안

3.1 소스프로그램에 대한 표절 동향

산업 분야에서는 IoT가 도입되면서 점점 더 네트워크에 긴밀하게 연결되고 있다. 이러한 변화로 전통적으로 보호가 어려운 기간시설로까지 사이버 공격이 확대되고 있는 상황에서 강력한 암호화 필요성 대두되고 있으며 모든 곳에 암호화하라(Encrypt everywhere)'는 말은 IT 업계에서 하나의 주문(mantra)이 되고 있다.[54] 인터넷과 같이 불안전하고 취약한 네트워크에서 인간과 시스템 간에 많은 커뮤니케이션과 교류가 이뤄지면서 오가는 데이터에 강력한 암호화를 적용해야 할 필요성은 오랫동안 제기돼 왔다. 그 결과 이제는 일반적으로 암호화가 도입되고 있지만 여전히 많은 신규 기기와 앱에 암호화가 허술하게 구축되면서 취약점을 이용해 공격자들이 통신 데이터에 접근 가능한 상황이다. 이러한 상황에서 우리는 기존의 암호화를 유지하기 위해서는 하기와 같은 소스코드의 보안 절차를 제안한다.

3.2 프로그램 관련 기술적 표절 검사기법

기존의 프로그램 표절검사 기법으로는 주로 소스코드나 프로그래밍 언어의 구조를 이용하여 표절 검사를 하는데 모두 소스코드에 기반을 둔 검사 방법이기 때문에 소스코드가 반드시 필요한 상태이다. 또한 표절 검사 기법은 크게 문자열 기반과 토큰 기반, AST (Abstract Syntax Tree) 기반, PDG(Program Dependency Graph)

기반으로 분류할 수 있다[30-34]. 본 연구에서는 Apple IoS 의 OS X platforms 기반의 IBM 에서 만든 IoT 오픈 소스코드 프로그램을 출시하였는데 이 중에 일부인 스마트 폰과 사물간의 인터페이스 용도의 Login View Controller 와 IoT Start View Controller 의 프로그램을 대상으로 텍스트 마이닝으로 원본 소스프로그램과 수정된 소스프로그램 사이의 실질적 유사성을 연구하였다. 하기는 기존에 프로그램 복제의 대표적인 유형은 Table 2와 같다.

Table 2. Comparison of checking methods for replication type [19]

Replication target program change	Matrix counting	Token pattern matching	Gene sequence analysis	AST
Replicate the original program	O	O	O	O
Change simple description	O	O	X	O
Blank and format conversion	O	O	O	O
Change the name of a variable or function	O	O	O	O
Add unnecessary variables and sentences	△	△	△	△
Change control structure	X	X	X	△
Type change	X	X	X	△
Replace code blocks, statements, and functions	X	X	X	O
Replace Operand / Operator	X	X	X	O

※ (Duplication judgment: O, small effect: Δ, very sensitive X)

물론 위와 같이 프로그램의 표절의 여러 방식이 존재하지만 실제 동일하지 않는 이상 표절을 찾기란 쉽지 않는 부분이며 아직도 불필요한 변수 및 문장 추가 그리고 제어구조 바꾸기, 타입 바꾸기는 실질적 유사도를 판단하기 어려운 상황이다.

4. 실험 및 평가

4.1 실험

본 실험에서는 R 프로그래밍을 이용하여 기존 IoT에 사용된 프로그램에서 사용된 소스코드를 분해하여 일괄적으로 분류하였고 원본 프로그램에서 수정 변환하여 같은 기능을 하는 프로그램을 각 각 분류하여 하기 Fig. 3과 Fig. 4와 같은 절차로 변환하여 검사 하였다. 이 방식은 데이터 마이닝 기법을 이용하여 두 가지 프로그램의 소스 텍스트와 구문의 연관성 분석하기 위하여 추출하여 유사도를 확인 하였다.

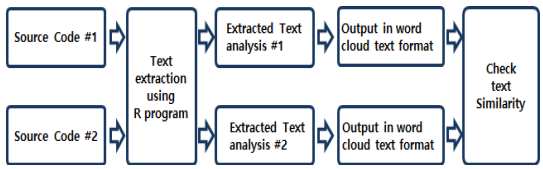

Fig. 3. Programs Text-mode similar classification

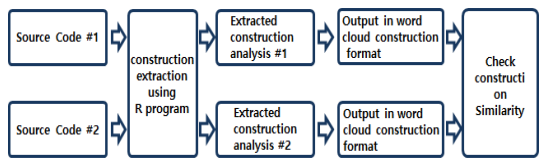

Fig. 4. Program syntax-mode similar classification

Fig. 3과 같이 먼저 2개의 IoT 인터페이스의 소스 프로그램을 R 프로그램을 동작하여 text 을 추출하고 그 다음 결과에 지장을 주지 않는 수식 기호 등을 삭제하여 text 을 정제하여 시각화 하였으며 프로그램 코드의 사용 빈도수에 따라 시각화되어 표현되는데 두 가지의 프로그램의 text 을 서로 비교하여 프로그램의 속성을 알 수 있었다. Fig. 4는 기존에 동일한 소스프로그램의 구문 프로그램 그대로의 형태를 사용 빈도수에 따라 시각화 하여 출력하여 기존에 프로그램과 비교하였다. 또한 하기 프로그램 Fig.5는 IoT 전용 오픈소스프로그램으로 스마트 폰에서 해당 어플로 접속했을 경우에 인터넷 기반 IoT와 연결된 디바이스의 Control 하기 위하여 만들어진 프로그램으로서 기능이 구현되는 원본 프로그램이며 Fig.6은 원본 프로그램과 같은 결과의 기능을 하지만 프로그램의 순서와 소스 코드 변경 그리고 일부 코드 삭제 그리고 코드 추가로 인하여 수정된 프로그램을 나타내고 있습니다.

```
- (void)viewWillAppear:(BOOL)animated
{
    [self updateViewLabels];
    AppDelegate *appDelegate = [[UIApplication sharedApplication] delegate];
    appDelegate.currentView = LOGIN;

    self.authTokenField.secureTextEntry = YES;
    self.authTokenField.delegate = self;
    self.deviceIDField.delegate = self;
    self.organizationField.delegate = self;

    self.organizationField.placeholder = IOTOrgPlaceholder;
    self.deviceIDField.placeholder = IOTDevicePlaceholder;
    self.authTokenField.placeholder = IOTAuthPlaceholder;

    if (![appDelegate.organization isEqualToString:@""])
    {
        self.organizationField.text = appDelegate.organization;
    }
    if (![appDelegate.deviceID isEqualToString:@""])
    {
        self.deviceIDField.text = appDelegate.deviceID;
    }
    if (![appDelegate.authToken isEqualToString:@""])
    {
        self.authTokenField.text = appDelegate.authToken;
```

Fig. 5. Original source program

```
- (void)viewWillAppear:(BOOL)animated
{
    [self updateViewLabels];

    self.aDelegate.currentView = LOGIN;

    self.authToken.secureTextEntry = YES;
    self.authToken.delegate = self;
    self.deviceID.delegate = self;
    self.organization.delegate = self;

    self.organization.placeholder = IOTOrgPlaceholder;
    self.deviceID.placeholder = IOTDevicePlaceholder;
    self.authToken.placeholder = IOTAuthPlaceholder;

    if (![self.aDelegate.organization isEqualToString:@""])
    {
        [self.organization setText:self.aDelegate.organization];
    }
    if (![self.aDelegate.deviceID isEqualToString:@""])
    {
        [self.deviceID setText:self.aDelegate.deviceID];
    }
    if (![self.aDelegate.authToken isEqualToString:@""])
    {
        [self.authToken setText:self.aDelegate.authToken];
```

Fig. 6. Translated source program

Fig. 7. Original text mining using R program

Fig. 8. Conversion text mining with R program

외관 상 서로 다른 소스프로그램에 대한 마이닝을 진행한 원본 프로그램의 결과 값은 Fig. 7과 같이 표현이 되는데 빈도가 높은 코드는 크게 표시되어 시각화되어 표현이 되는데 각각의 프로그램의 코드의 빈도수가 같은 코드의 형태가 표시되고 있음을 보여주며 Fig. 8은 수정 변환된 소스 프로그램을 기준으로 텍스트 마이닝을 한 결과가 표시된다. 따라서 Fig. 7과 Fig. 8을 비교 하였을 때 최초 다른 프로그램으로 처음에 보이지만 결과를 분석해 보면 위 appDelegate 와 aDelegate 는 같은 명령어 소스인데 코드 명을 수정하여 나타냄을 알 수 있고 또한 전체적으로 사용한 언어를 추리하여 UI application 의 프로그램으로 인지할 수 있고 또한 더 나아가서 현 프로그램은 사용자 정보가 필요하여 등록하고 들어가야 되는 프로그램인지를 빠른 시간 내에 전체적인 프로그램의 의미를 파악 할 수 있다. 따라서 프로그램을 복제하여 사용하기 위하여 삭제되고 추가된 언어도 있으나 실질적인 의미상으로는 원본 소스프로그램의 수정된 소스프로그램은 의미상 일치함을 보여주고 있다. 각기 다른 프로그램의 저작자가 명령어을 달리 사용하더라도 결과론적으로 같은 기능의 프로그램으로 파악 할 수 있고 또한 같은 의미의 내용이라도 어떻게 프로그램의 순서를 달리 하던지 아니면 명령어를 달리 사용하는 방식이 있고 그리고 프로그램 작성 시에 스페이스 등 특별한 기능을 추가함으로 실제 원본과 다르게 보이려고 하지만 프로그램의 의미상의 속성을 텍스트 마이닝을 통하여 1차적으로 빠르게 파악하고 분석 할 수 있다.

```
___AppDelegate_appDelegate_=_[[UIApplication_sharedApplication]_delegate];
                                                                        7
____NSLog(@"%s:%d_entered",__func__,__LINE__);
                                                                        3
_________appDelegate.sensorFrequency_=_IOTSensorFreqDefault;
                                                                        2
_clientID_=_[NSString_stringWithFormat:IOTClientID,_appDelegate.organization,_IOTDeviceType,_appDelegate.deviceID];
                                                                        2
_____Messenger_messenger_=_[Messenger_sharedMessenger];
                                                                        2
```

Fig. 9. Original program extracted from R program

```
___AppDelegate_aDelegate_=_[app_delegate];
                                                                        7
___UIApplication_app_=_[UIApplication_sharedApplication];
                                                                        7
____NSLog(@"%s:%d_entered",__func__,__LINE__);
                                                                        3
_________[aDelegate_setSensorFrequency:IOTSensorFreqDefault];
                                                                        2
__clientID_=_[NSString_stringWithFormat:IOTClientID,_aDelegate.organization,_IOTDeviceType,_aDelegate.deviceID];
                                                                        2
_____Messenger_messenger_=_[Messenger_sharedMessenger];
```

Fig. 10. Conversion program extracted from R program

위 결과 값 Fig. 9, Fig. 10에서는 텍스트 마이닝 했을 때 각 프로그램에 사용된 프로그램 구문의 사용 n 수 을 알 수 있는데 실질적으로 Fig. 10에서 확인하여 보면 원본 프로그램과 같은 내용이지만 2개로 분할하여 구문을 작성하였습니다. 그래서 추가로 구문 마이닝을 한 결과는 Fig. 11과 Fig. 12 에 나타나며 결과로 비교할 수 있습니다.

Fig. 11와 Fig. 12의 결과 값은 처음엔 각기 다른 프로그램으로 보이지만 원본 Appdelegate appdelegate =UI Application_SharedApplication) Delegate 와 분리되어 있지만 Appdelegate app delegate 로 시작되는 문장과 UI Applicaiton 으로 시작하는 문장은 분리되어 있지만 원본과 같은 문장으로 알 수 있다. 따라서 소스구문을 분석하여 보았을 때 분리되어 구문이 존재하지만 즉 표현은 다르지만 그 결과 값은 동일하게 나타 낼 수 있으며 따라서 각 소스코드의 사용의 빈도수를 파악 하였으며 이를 통하여 전체적인 구조를 파악 하여 텍스트 타입과 구문 타입을 통하여 프로그램 최대한의 유사도를 알아보았으며 본 연구는 IoT 을 스마트 폰과 연동하여 대상사물과 통신하는 프로그램을 기준으로 작성하였으며 해당 프로그램의 표절을 텍스트 마이닝 기법으로 분석하여 프로그램의 텍스트와 구문을 분석하여 프로그램과 프로그램사이의 특징을 분석할 수 있으며 프로그램을 전체를 비교하여 해당 프로그램의 사용 용도를 빠른 시간 내에 전반적으로 파악 할 수 있다. 또한 다른 향후 연구로는 클래스 간의 관계를 표현하는 n방법에 관한 연구가 필요하며 지금의 제안방법은 클래스의 유사도를 분석하는데 집중되어 있지만 전체 프로그램의 흐름에 대한 객체 간의 관련성도 추가적은 데이터 마이닝을 통하여 고려하여 유사도 분석이 이루어진다면 표절 검사의 정확도는 더 올라갈 것으로 생각된다[55].

Fig. 11. Original Source syntax mining using R program

Fig. 12. Converted original source syntax using R program

본 실험에서는 IoT 을 스마트 폰과 연동하여 대상사물과 통신하는 프로그램을 기준으로 작성하였는데 프로그램의 최종 결과 값은 Fig. 13에 나타나 있으며 각기 다른 프로그램의 형식이지만 결과 값은 동일한 결과 값을 나타내었다. 따라서 소스프로그램에 대한 프로그램의 표절을 텍스트 마이닝으로 분석하여 해당 프로그램의 텍스트 코드의 속성을 파악하여 분류 할 수 있고 추가적으로 프로그램 구문을 기준으로 전체적인 프로그램의 유사도를 알아 볼 수 있으며 프로그램 속성을 파악할 수 있어서 추후 프로그램에서 대한 법적으로 저작권 주장 시에 정

량적인 프로그램은 다르더라도 실질적 유사성 측면에서 참고가 될 것이다.

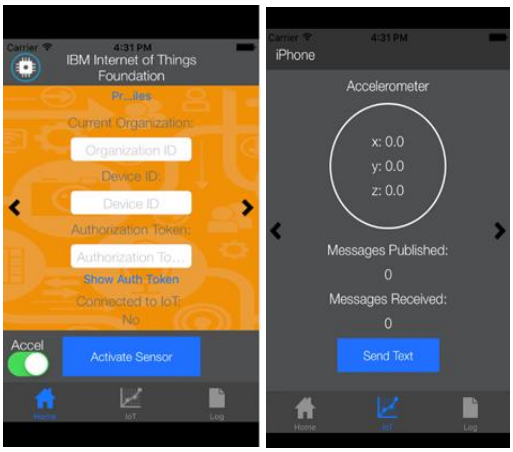

Fig. 13. Smart phone-based IoT interface screen

4.2 IoT 프로그램에 대한 분석절차도입

먼저 전체적인 정량적 유사성을 도출하여 기존에 프로그램을 비교하고 비슷한 표현요소를 메타분석을 통하여 도출하며 프로그램의 실질적 결과 통하여 하여 기존 프로그램과 비교 분석하여서 문제를 해결한다. 이제까지 기존에 프로그램에 대한 유사성 확인 방식은 서로 다른 두 프로그램의 유사도를 측정하는 시스템을 제시하는 구문트리 비교를 통한 프로그램 유형 복제 검사를 채택하고 있는 상태이다. 그러나 이러한 구문트리 기법으로는 정량적 유사도를 통한 기법은 한계가 있는 상태이며 추가적으로 프로그램의 표현요소들에 대한 검토를 종합하여 컴퓨터프로그램을 실질적으로 유사한 정도를 복제하였는지 여부를 판단하는 것이 중요한데 텍스트 마이닝을 통한 프로그램의 소스코드와 구문을 비교하여 의미적인 정확도를 더 올릴 수 있다. 또한 추가적으로 데이터 마이닝을 통한 클래스 구조를 분석하고 또한 그래프 비교와 데이터 마이닝 맵을 통한 프로그램 표절 검사방법이 더욱 더 유사도의 정확도를 올릴 수 있을 것이다. 기존에 채택된 표절 방식에서 Java 프로그램의 표절을 검사하는 방법을 제안 하였는데 멤버 변수와 메소드 간의 참조 관계를 나타내는 그래프를 생성하여 변수 참조 관계는 이분 그래프 형태로 나타나는데 이렇게 생성된 그래프를 대상으로 그래프 동형 검사를 적용하여 프로그램 간의 유사도를 측정 하였는데[56] 이러한 경우 외로 많은 프로그래밍 언어의 표현이 다양하므로 문언적인 외형상 다르게 보이는 소스코드라도 실제로는 동일한 표현 범주에 해당하는 경우가 있을 수 있어 어문저작물에 비하여 판단의 방법이 어려우며 매우 분석적일 수밖에 없는데 이를 해결하기 위하여 프로그램 저작자의 사상과 감정이 들어간 저작물의 형태인 그 표현이 보호 받는데 있어서 텍스트 마이닝 기법을 활용하여 최초 프로그램 저작자가 자신의 프로그램에 대한 결과와 특성을 고려하여 최종 데이터 마이닝 기법으로 도출 한 다음 그 결과가 비정형, 반정형 데이터를 통하여 차이를 발견하는[57] 부분에서는 정확한 결과 값을 보여 주었다. 그에 따라 프로그램의 저작등록의 여부를 알 수 있고 추후 관련 프로그램의 연관성과 예측 할 수 있을 것이다.

5. 결 론

본 연구는 앞서 살펴본 바와 같이 IoT (사물인터넷)의 용용프로그램 형태인 소스프로그램을 대상으로 프로그램의 특허법과 저작권법으로의 법적인 보호 방향을 살펴보았으며 프로그램 표절에 대하여 실질적 유사성 측면에서 법적인 부분이 중요한 바와 같이 실질적 유사성 측면에서 프로그램 연구를 하였으며 실험을 통하여 결과를 알 수 있었는데 본 연구에서는 앞으로 인터넷이 발전하면서 모든 사물과 사물 또는 사물과 인간을 연결하는 사물인터넷 IoT (Internet of Things) 기반의 프로그램을 대상으로 실험을 하였는데 그 이유는 사물인터넷이 발전할수록 소프트웨어를 복제와 표절을 통한 개인정보유출이 심각하게 늘어날 수 있다. 따라서 본 연구에서는 스마트 폰의 소스 코드을 이용하여 IBM에서 제공하는 IoT 오픈 소스프로그램을 응용하여 실험을 진행 하였다. 원본 프로그램과 역변환한 프로그램을 비교하기 위하여 R 프로그램을 사용하여 텍스트 마이닝으로 텍스트와 구문을 통한 유사도를 비교 분석하였다. 이러한 방식은 기존에 정형화된 분석 방식에서 효과를 거두지 못한 의미적인 정확도를 구문 마이닝과 텍스트 마이닝으로 실질적 유사 정확도를 더 올렸는데 구체적으로 불필요한 문장의 변수 걸려 낼 수 있으며 약간의 프로그램의 타입을 변경하여도 발견이 가능하다. 또한 최근 IoT 보안을 위하여 SHA-256 해시 프로세서의 면적 효율적인 설계 방안[58] 그리고 IoT 환경의 네트워크 공격 탐지를 위한 정규 표

현식 매칭 오버헤드 감소방식 [59] 등 다양한 프로그램 보안 방법이 중요하게 대두되고 있다. 따라서 IoT 는 불법 복제와 보안이 중요한 상황으로 본 연구는 복제로 인한 프로그램 창작자의 프로그램의 원천 저작자의 권익을 보호하는데 보탬이 될 것이다. 현재 많은 오픈소스코드 자체는 법적으로 규제가 심하지 않으나 이를 이용하여 수많은 유사 프로그램이 넘쳐나고 있는 것이 현실이다. 물론 2차적 저작물로 인정될 수 도 있으나 원작자의 동일성 유지권이 심하게 회손 당할 수 있는 상황이며 또한 새로운 독립 저작물을 만들기 위해 소스 코드를 다른 문장으로 추가하고 변경하고 타입과 같은 의미 이지만 다른 문장으로 바꿀 시에 유사성이 파악이 힘들고 실질적 유사성을 분석하기에는 많이 힘든 상태이다. 따라서 원천 프로그램 저작자와 2차적 프로그램 저작자 그리고 독립 프로그램 저작자 간의 분쟁이 끊임 없이 일어나고 있는 상황에서 무엇보다 프로그램에 대한 보호가 필요한 상태이다. 최근의 미국의 판례인 오라클과 구글의 프로그램 저작권 보호측면에서 살펴보면 아직 최종 결과가 나오기는 이르지만 소프트웨어 판단함에 있어 아이디어·표현 이분법은 여전히 적용하고 표현된 소스코드 SSO(Sequence, Struc-ture and Organization) 가 아이디어 그 자체인지 아이디어의 표현에 해당하는지의 여부가 판단기준으로 제시되고 있다는 점에서 향후 파기환송 후 법원에서 합체의 원칙, 짧은 어구 원칙, 필수장면이론, 사실상의 표준 등 저작물성 부정의 요건과 침해주장에 대한 항변으로서 공정사용의 원칙 적용기준을 제시할 것으로 예상되며 공정이용과 저작물성의 인정사이에 오라클과 구글의 사이의 논란은 계속 될 것으로 예상된다. 따라서 미국 법원이 소프트웨어의 보호에 있어 특허권으로서의 보호는 제한적으로 해석하는 반면, 저작권으로서의 보호는 전향적으로 그 범위를 확대하는 것으로 판단될 수 있는 상황이다. 따라서 본 연구에서는 프로그래밍 언어의 표현이 다양하므로 문언적인 외형상 다르게 보이는 소스코드라도 실제로는 동일한 표현 범주에 해당하는 경우가 있을 수 있어서 어문저작물에 비하여 판단의 방법이 어려우며 매우 정량적일 수밖에 없는데 소스프로그램의 침해에 대한 분쟁은 특별히 IoT 의 상용화 을 기점으로 늘어날 수 있는 상황이며 개인정보보호와 같이 개인 프라이버시 측면에서 프로그램보호는 필요한 상황이다. 따라서 본 연구에서 새로운 도입한 텍스트 마이닝 기법으로 프로그램의 코드와 문장을 비교 분석하여서 문장은 다르지만 같은 결과를 도출하는 부분에서 좀 더 정확도를 올리는 계기가 되었으며 향후 IoT 프로그램의 보다 나은 발전을 위하여 추가적인 데이터 마이닝을 통하여 프로그램 소스코드의 개체적인 연관성 분석 할 수 있을 것이며 프로그램의 각각 요소 별로 개별 분석하여 우리나라도 실질적 유사성을 좀 더 면밀히 판단하기를 기대한다.

REFERENCES

[1] C. S. Pyo, H, Y. Yong, N. S. Kim & H. C. Bang. (2013). IoT(M2M) Technology Trends and Prospects. *Journal of the Korean Institute of Communication Sciences (Information and Communication), 30(8),* 3-10.

[2] J. H. Jeon, H. G. Hong, W. S. Lee & H. J. Kim. (2015). Open Source Things Internet (OSIoT) Trends and Forecasts. *Journal of the Korean Institute of Communication Sciences (Information and Communication), 32 (5),* 23-30.

[3] H. S. Jang, H. J. Kim, & T. S. Son. (2015). A Study on Cyber Security Issues in Industrial IoT Environment. *Journal of Information Security, 25(5)*, 12-17.

[4] H. M. Lee & D. K. Shin. (2015). Open Software Platform Design for the Internet. *Proceedings of the Korea Information Science Society Conference,* 1492-1494.

[5] H. K. Lee, M. H. Kim & H. C. Bang. (2014). Feature: Internet: Things Internet technology trends and development direction. *Journal of Korea Institute of Information Scientists and Engineers 21-2:* 14-21.

[6] Y. G. Hong, M. K Shin & H. J. Kim. (2013). Standardization trend of internet (IoT / M2M). *OSIA Standards & Technology Review, 26 (2),* 8-17.

[7] D. H. Kim, S. W. Yoon & Y. P. Lee. (2013). Security or IoT services. *Journal of the Korean Institute of Communication Sciences (Information and Communication), 30(8),* 53-59.

[8] J. H. Seo, Y. S. Min & Y K. Park. (2015). Technical analysis for software protection. *Journal of Information Technology, 13(1),* 33-39.

[9] S. M. You. (2015). Government Policy Direction for Software Copyright Protection. *Journal of Information Technology, 13(1),* 41-47.

[10] J. S, Lee, S. K. Chang & I. H. Cho. (2014). Hardware based security for Internet of Things (IoT) security. *Proceedings of the Fall Conference of the Korean Society for Internet Information,* 241-242.

[11] J. S. Choi & H. W. Kim. (2013). *IoT security technology trend.*

[12] *ITU Internet Reports.* (2005). Internet of Things, ITU

[13] ITU-T Y.2060, (2012). *Overview of the Internet of Things.*

[14] A. Wright. (2009). *Cyber security for the power grid: cyber security issues & Securing control systems,* ACM CCS.

[15] J. S. Choi & H. W. Kim. (2013). *IoT security technology trend.* Table 1, 2814

[16] BILL WEINBERG, *Open Source and the Internet of Things: Roles, Reach and Rationale for Deploying* OSS, http://rtcmagazine.com/articles/ view/105734

[17] B. I. Kim. (2009). Open source license violations and copyright infringement. Quarterly copyright.

[18] H. R. Reddy. (2009). Jacobsen v. Katzer: the Federal Circuit weighs in on the enforceability of free and open source software licenses. Berkeley Tech. LJ, 24, 299.

[19] Y. C. Kim, S. C. Hwang & J. Y. Choi. (2005). Program similarity evaluation algorithm." *Journal of the Internet Information Science Society.* 51-64.

[20] N. J. Park & G. J. Ahn. (2010). Privacy Protection in Smart Grid. *Journal of information security, 20(3),* 62-78.

[21] H. J. Seo, D. K. Kim, J. H. Kim, J. S. Choi & H. W. Kim. (2014). Feature: Internet of Things: Security and Privacy Protection on the Internet. *Journal of Information Processing, 21(2),* 48-60.

[22] S. W. Cha. (2014). *Big Data Protection of the environment and privacy. IT and Law Studies*, 8, 193-259.

[23] Y. S. Son & J. H. Park. (2015). Home IoT technology status and development direction. *Journal of the Korean Institute of Communication Sciences (Information and Communication), 32(4),* 23-28.

[24] S. H. Lee & Y. Y. Cho. (2015). A Study on the Strengthening of National Cyber Security in the Internet Age. *Political Information Research, 18 (2),* 1-30.

[25] H. S. Chang, H. J. Kim & T. S. Son. (2015). A Study on Cyber Security Issues in Industrial IoT Environment." *Journal of information security* 25.5: 12-17.

[26] http://www.ddaily.co.kr/news/article.html?no=137711 Digital Daily, (2016)

[27] Y. L. Lee & J. S. Kim. (2014). Personal Information Protection Framework in IoT Environment. *Korea Contents Association 2014 Fall Conference,* 277-278.

[28] S. S. Kim. (1996). Introduction to Computer Science, *Hongneung Science Publishing.*: 183-216.

[29] Oracle America, Inc. v. Google Inc., 2014 WL 1855277 (Fed. Cir. May 9, 2014).

[30] Y. E. Kim, Y. J. Lee & W. G.2013). Program Plagiarism Checking Method by Class Structure Graph Comparison. *Journal of the Korea Contents Association,* 13 (11), 37-47.

[31] C. Roy & J. Cordy. (2007). A Survey on Software Clone Detection Research, *Technical Report* 541, Queen's University at Kingston.

[32] S. Narayanan & S. Simi. (2012). Source Code Plagiarism Detection and Performance Analysis Using Fingerprint Based Distance Measure Method. *2012 7th International Conference on Computer Science & Education,* 1065-1068.

[33] J. Ji, G. Woo & H. Cho. (2007). A Source Code Linearization Technique for Detecting Plagiarized Programs. *ACM SIGCSE Bulletin 39(3),*, 73-77.

[34] C. Liu, C. Chen, J. Han, & P. Yu, (2006). GPLAG: Detection of Software Plagiarism by Program Dependence Graph Analysis," *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining,* .872-881.

[35] S. O. Yoon. (2013). A Study on Risk Classification of Big Data. *Journal of the Korean Association of Local Information, 16(2),* 93-122.

[36] J. J. Kim. (1997). Computer General, Hong Jin Publishing Co., Ltd pp196 & S.Y. Yuk.(2003) Patentability of computer programs. *Comparative Judgment* 1.2:481-511.

[37] Supreme Court precedent 1996.08.23, 95DO2785

[38] Supreme Court precedent 2001. 05. 15, 98DO732

[39] Seoul High Court case 1999.04.07, 98. 23616

[40] Seoul case law support for East1998. 4.NA 13. Sentence 96KAHAP 1921 verdict

[41] Seoul case law support for East 1997.11.28, 95GAHAP 11403

[42] Supreme Court precedent 2002.06.28, 2001DO2900

[43] Seoul High Court 1996.10.11, 96NA1353

[44] Supreme Court precedent 2002.12.26, 2000DA.13757

[45] Seoul District Court Copy 1997.01.10. 96NO7508 / Supreme Court 1997.07.22. 97DO764/Supreme Court 1997.05.30. 97DO766 /Supreme Court 1997.05.23. 97DO767 /Supreme Court 1997.06.13. 97DO768 Computer program protection law

[46] Supreme Court precedent 1998.03.13, 97DA45259

[47] Supreme Court precedent 2000.11.10, 98DA60590

[48] Seoul High Court case 1993.06.18, 92NA64646 / Seoul District Court 1993.04.22, 92GAHAP31298

[49] J. Holler, V. Tsiatsis, C. Mulligan, S. Avesand, S.

Karnouskos & D. Boyle. (2014). From Machine-to-machine to the Internet of Things: *Introduction to a New Age of Intelligence. Academic Press.*

[50] J. E. Cohen & M. A. Lemley. (2001). Patent scope and innovation in the software industry. *California Law Review,* 1-57.

[51] T. H. Goo. (2005). Technical Characteristics of Computer Programs and Patent and Copyright Protection. 13.

[52] J. Bessen & E. Maskin. (2009). Sequential innovation, patents, and imitation. The RAND *Journal of Economics, 40(4),* 611-635.

[53] S. J. Oh & H. W. Lee. (2013). Copyright Law 3rd Edition, Park Young-sa, 873

[54] http://it.donga.com/23161/IT DONGHA S.W Lee 2015.12.03.

[55] Y. C. Kim, S. C. Hwang & J. Y. Choi. (2005). Program similarity evaluation algorithm. *Journal of the Internet Information Science Society, 6(1),* 51-64.

[56] Y. E. Kim, Y. J. Lee & W. G.(2013). Program Plagiarism Checking Method by Class Structure Graph Comparison. *Journal of the Korea Contents Association, 13(11),* 37-47.

[57] B. Y. Lee, J. T. Lim & J. S. Yoo. (2013). Utilization of social media analysis technique using *big data. Journal of the Korea Contents Association, 13(2),* 211-219.

[58] S. H. Lee & K. W. Shin. (2018). Area-efficient design of SHA-256 hash processors for IoT security. *The Journal of the Korea Information and Communications Society, 22(1),* 109-116.

[59] H. E. Moon, J. Y, Sung, H. S. Lee, K .I, Jang, K. Y. Kwak & S. T. Woo. (2018). Detection of attack group using malware and packer detection. *Journal of the Korean Institute of Information Scientists and Engineers, 45(2),* 106-112.

[60] H. S. Yang. (2017). A Study on the Security Quality Evaluation Method of Smart Healthcare System. *Digital fusion research, 15(11),* 251-259.

[61] Y. C. Ahn, Y. H. Shin, G. H. Lee. (2015). A Study on the countermeasure against BLE based ZEP system attack technique using the fusion of Big Data platform and monitoring system. *Digital fusion research, 13(8),* 331-336.

이 종 식(Jong-Sik Lee) [정회원]

- 2005년 7월 : 한양대 전자공학 (공학석사)
- 2016년 2월 : 연세대지식재산권법 (법학석사)
- 2014년 2월 : 성균관대 인터랙션 사이언스 (HCI박사)
- 2017년 9월~현재 : 성균관대인터랙션사이언스 겸임교수
- 2014년 2월~현재 : 성균관대인터랙션사이언스연구소 선임연구원
- 2017년 2월~현재 : 대한신학대학원대학교 겸임교수
- 관심분야 : Research area: HCI, HRI, UI, Cognitive Science, Emotion Recognition, Text Mining, IPRs, Media of mission
- E-mail : jongsic@skku.edu