논문 2013-1-6

동적 연결 라이브러리 기반의 MS 윈도우 애플리케이션 분류 기법

한용만*, 황윤하**, 김동진*, 최종천***, 조성제+† , 유해영+, 우진운+

A Categorization Technique for MS Windows Applications Based on Dynamic-link Libraries

Yongman Han*, Younha Hwang**, Dongjin Kim*, Jongcheon Choi***,
Seong-je Cho+† , Heayoung Yoo+, Jinwoon Woo+

요 약

대표적인 소프트웨어에는 텍스트 편집기, 동영상 재생기, 이미지 뷰어, 네트워크 프로그램, 게임 프로그램, 메신저 프로그램 등이 있는데, 소프트웨어 가치가 높아지면서 다수의 프로그램들이 전체 또는 일부 도용되고 있다. 소프트웨어 도용을 효율적으로 탐지하기 위해서는 프로그램을 특징에 따라 분류하는 것이 필요하다. 본 논문에서는 MS 윈도우 시스템에서 동적 연결 라이브러리에 근거하여 윈도우 애플리케이션을 효과적으로 분류하는 기법을 제안한다. 제안된 기법은 잘 알려지지 않은 프로그램을 분석하는데 도움을 줄 수 있다.

Abstract

Well-known software includes text editors, media players, image viewers, networking programs, game programs, messenger programs, etc. As software is getting more valuable, a malicious programmer illegally copies all or part of a valuable program and resells the copied one. Therefore, we need to categorize programs according to their characteristics in order to efficiently detect software theft. In this paper, we propose a technique that effectively classifies Microsoft Windows applications based on dynamic-link libraries. The proposed technique can be used to analyze suspicious programs even when their functions are not known in advance.

한글키워드 : 윈도우 애플리케이션, 소프트웨어 분류, 동적 연결 라이브러리

※ 이 논문은 2013 제18회 한국SW감정평가학회 춘계 학술대회에서 발표된 '유사한 MS 윈도우 애플리케이션 탐지를 위한 소프트웨어 분류에 관한 연구' 논문을 확장한 것임.

* 단국대학교 컴퓨터학과
** 단국대학교 컴퓨터학과 소프트웨어보안전공
*** 단국대학교 컴퓨터과학과
+ 단국대학교 소프트웨어학과

※ 본 연구는 문화체육관광부 및 한국저작권위원회의 2013년도 저작권기술개발사업의 연구 결과, 그리고 미래과학창조부 및 한국인터넷진흥원의 "고용계약형 지식정보보안 석사과정 지원사업"의 연구 결과로 수행되었음 (과제번호 H2101-13-1001)

† 교신저자:조성제(Email: sjcho@dankook.ac.kr)

접수일자: 2013.5.25 수정완료: 2013.6.21.

1. 서 론

소프트웨어 시장은 자동차, 조선, 항공, 의료 등 다양한 산업분야에서 융합되어 사용하고 있다. 다양한 애플리케이션들이 개발되면서 소프트웨어 산업의 범위는 점차 확대되고 있으며, 국내 시장에서 공개 소프트웨어에 대한 시장 규모는 그림 1과 같이 매년 증가하고 있다[1].

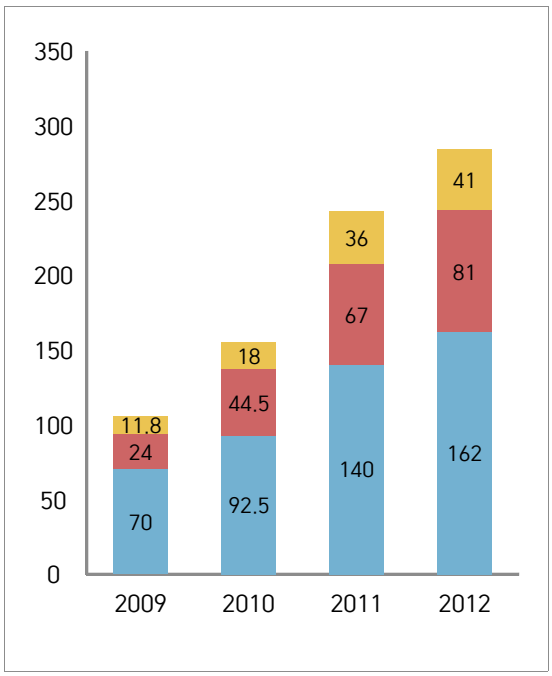

그림 1. 국내 공개SW 시장 규모 상세(억원)

한편, 최근 소프트웨어 가치가 높아지고 오픈소스 소프트웨어(OSS)를 활용하는 사례가 많아지면서, 소프트웨어 저작권을 위반하는 사례가 증가하고 소프트웨어를 불법으로 유통하는 문제가 야기되고 있다. 이러한 문제의 대책으로 소프트웨어의 도용을 탐지하기 위한 소프트웨어 필터링[2][3]과 유사도[4][5][6]에 대한 연구가 진행되고 있다.

대표적으로 소프트웨어 몽타주(Software montage) 기반으로 유사 프로그램 탐지 대상을 필터링하거나 안드로이드 애플리케이션을 분류하는 연구가 수행되었다[3][6]. 소프트웨어 몽타주는 쉽고 빠르게 추출할 수 있는 소프트웨어 특성으로, 박희완 등은 API (Application Programming Interface), 문자열, URL 정보 등을 사용하였다[3][6].

소프트웨어 분류에서는, 수많은 프로그램들로부터 비슷한 프로그램들을 1차적으로 빠르게 수행하는 것이 필요하다. 즉, 대상 프로그램의 특성을 빨리 추출하여 비교하는 것이 중요하다[6].

본 논문에서는 MS 윈도우 시스템에서 애플리케이션이 가지고 있는 DLL 정보를 정적으로 분석하는 윈도우 애플리케이션을 분류하는 기법을 제안한다. 윈도우 애플리케이션에서 DLL 정보는 API 정보보다 쉽고 간단하게 추출할 수 있다. 제안 기법을 검증하기 위해서, 많이 사용되는 애플리케이션들을 선정하고 그들로부터 DLL 정보를 분석한 후, 유사 애플리케이션을 탐지하는 실험을 수행하였다.

이러한 소프트웨어 분류 기법은, 범죄에 사용된 소프트웨어의 특성을 분석하는 디지털 포렌식(Digital forensic)에 사용될 수 있다. 또한, 도용이 의심되지만 원본 소프트웨어가 분명하게 주어지지 않은 상황에서, 소프트웨어 유사성을 비교할 대상 집합의 크기를 줄이는데 사용될 수 있다. 또한, 악성 소프트웨어의 변종을 탐지하는데도 활용 가능하다.

본 논문의 구성은 다음과 같다. 2장에서는 관련연구에 대한 내용을 살펴보고, 3장에서는 DLL 파일을 소프트웨어 군별로 추출하고, 이를 기반으로 소프트웨어 분류연구를 진행하였다. 마지막으로 4장에서는 결론 및 향후연구를 정리하였다.

2. 관련 연구

2.1 윈도우 실행코드

윈도우 운영체제에서 실행 가능한 파일형식으로 PE (Portable Executable) 파일 포맷을 지원한다. PE파일 포맷은 링크를 위한 DLL (Dynamic-link library)과 API (Application programming interface)의 Import 및 Export 테이블 등을 포함하고 있으며 프로그램이 실행되면 운영체제 로더가 이를 분석하여 필요한 DLL 파일을 메모리에 함께 로드하게 된다. DLL 파일에 포함된 API함수 등에 대한 연구와 함께 다양한 정보를 내재하고 있다는 점에서 소프트웨어별 유사성에 대한 관계를 확인 할 수 있었다.

본 논문에서는 DLL 파일의 정보만을 가지고 유사 소프트웨어 및 소프트웨어 분류 연구를 진행하였다.

2.2 소프트웨어 몽타주 기법

소프트웨어 몽타주(Montage)는 대상 프로그램에 내재되어 있는 특성으로 그 프로그램으로부터 빠르게 추출 가능한 정보를 말한다. 일반적인 몽타주처럼, 소프트웨어 몽타주도 대상 프로그램에 대해 겉으로 드러난 특징이나 기능을 의미하며, 정밀한 정적 분석을 거치지 않고 빠르게 추출할 수 있다. 수많은 프로그램들로부터 유사한 프로그램을 1차적으로 효율적으로 분류하기 위해서는, 빠른 추출 속도 및 비교가 중요한데, 소프트웨어 몽타주가 그 용도로 적합하다[3][4][5][6].

박희완 등이 제안한 소프트웨어 몽타주[3]는 API 함수 호출과 문자열에 대한 빈도를 점수화하여 순위를 정하고 그에 따라 유사도를 측정하는 것에 비해, 본 논문에서는 DLL 파일 정보를 통해서 소프트웨어가 갖고 있는 특징을 확인하고 이를 다수의 소프트웨어 군들을 정리하여 소프트웨어를 분류한다.

기존의 몽타주 기법은 문자열의 경우 위변조가 쉽고, API함수 호출의 경우 수많은 API 함수를 분석하고 사용빈도를 추출함으로써 많은 노력과 시간이 드는 단점이 존재하였다.

따라서 본 연구는 MS의 윈도우 기반의 애플리케이션 실행파일이 갖고 있는 DLL 정보에 대한 분석을 진행하고, 이를 토대로 소프트웨어 분류 연구를 진행하였다.

3. DLL 파일 추출을 통한 SW군 분류

MS 윈도우 애플리케이션에 대한 PE 파일 포맷에서 DLL들에 대한 정보를 추출하고, 이를 활용하여 애플리케이션들을 분류하는 기법을 연구하였다. 기존 연구[3][6]에서는 API 함수 정보 등을 활용하였다면, 본 논문에서는 MS 윈도우 시스템에서 참조되는 DLL 정보와 특정 소프트웨어 군에서만 사용되는 DLL 정보를 분석하고, 가중치를 두어 점수를 계산하였다. 본 연구는 지난 5월 한국소프트웨어감정평가학회 춘계학술대회에서 발표한 내용을 수정 보완하였다[7].

본 논문에서 고려한 소프트웨어 군은 FTP 클라이언트(FTP client), 텍스트 편집기(Text editor), 미디어 플레이어(Media player) 등의 그룹으로, 각 군에 속한 애플리케이션들의 최신 버전을 사용하여 DLL 파일을 추출하였다. 표 1은 소프트웨어 분류 작업을 위해 사전조사를 실시한 각 제품군별 5개씩 15개의 프로그램 목록이다.

표 1. 실험 소프트웨어 목록

분류	프로그램명	버전	DLL 수
FTP	ALFTP	5.3.2	13
	FileZilla	3.6.0	17
	WinSCP	5.1.4	21
	ALDrive	1.0	20

	CuteFTP	9.0	19
Text	AkelPad	4.8.2	11
	edtitplus	3.51	21
	editplus	3.51	21
	wordpad	5.7	22
	HWP	2010	30
	notepad	2010	14
Media	GomPlayer	2.2.5	23
	KMPlayer	3.6.0	20
	Mplayer	6.4.9	21
	JetAudio	8.0.1	20
	Winamp	5.6.3	17

각 프로그램들은 가장 많이 활용되는 프로그램을 우선적으로 선별하였으며, MS-word와 같이 패킹기술이 적용된 애플리케이션이나 적은 수의 DLL 파일을 포함한 프로그램은 제외하였다.

3.1 분류를 위한 DLL 추출

소프트웨어의 분류를 위해 사용할 DLL파일을 소프트웨어별로 추출하였다. 소프트웨어군 별로 사용빈도가 높은 DLL 파일을 확인하고, 이를 100%, 80%, 60%의 사용 빈도로 구분하였다.

3.1.1 사용 빈도 100%의 프로그램 분류

소프트웨어별 분류를 위해 선택한 각 애플리케이션들을 기반으로 DLL을 추출하여 각 소프트웨어 실험군에서 모두 사용되는 DLL을 추출하고, 표 2와 같이 정리하였다.

표 2. SW 분류별 사용 DLL - 100%

DLL 파일명	FTP	Text	Media
ADVAPI32	o	o	o
COMCTL32	o	o	
GDI32	o	o	o
KERNEL32	o	o	o
OLE32	o	o	o
OLEAUT32	o	o	o
OLEDLG			
SHELL32	o	o	o
SHLWAPI			o
USER32	o	o	o
VERSION		o	
WININET			o

소프트웨어별로 분류할 수 있는 DLL 파일에 대해서 각 애플리케이션 별로 모두 쓰이는 DLL 파일과 특징을 가질 수 있는 DLL파일의 구분이 가능하였다. COMCT32.DLL 파일의 경우 FTP군과 Text 군에서 쓰이지만 Media 군에서는 사용되지 않거나 SHLWAPI.DLL 파일과 WININET.DLL 파일은 Media군에서 유일하게 보이는 등 그 차이를 구분 할 수 있었다. 사용 빈도율 계산은 식 1과 같이 각 소프트웨어 군에서 사용되는 DLL 파일에 대해 몇 개의 애플리케이션에서 사용하는지를 확인하였다.

< *식 1* - DLL 사용빈도율 계산하기 >

$$DLL\text{사용빈도율} = \frac{SW\text{군에서 } DLL\text{을 사용하는 프로그램 수}}{SW\text{군에 포함된 총 프로그램 수}} \times 100$$

< 예시 1 >

- FTP 군에서 COMCTL32 DLL파일에 대한 사용 빈도율 계산

1) FTP군의 소프트웨어 개수는 5개,

2) 그중 COMCTL32 파일을 사용하는 소프트웨어 개수는 5개
=> COMCTL32 DLL의 사용빈도율은

$$\frac{5}{5} \times 100 = 100(\%)$$

3.1.2 사용 빈도 80%의 소프트웨어 분류

두 번째로 소프트웨어 분류를 위한 특징이 될 수 있는 DLL 파일을 확인하기 위해 DLL 사용 빈도를 80%까지 낮추었다. 각 실험군별로 중복되는 DLL 파일의 수가 늘어났지만, Text editor 군의 경우 WINMM.DLL 등과 같이 분류 가능한 DLL 파일이 추가 되었다. 사용 빈도가 80%인 DLL 파일에 대한 정리를 표 3과 같이 하였다.

표 3. SW 분류별 사용 DLL - 사용 빈도 80%

DLL 파일명	FTP	Text	Media
ADVAPI32	o	o	o
COMCTL32	o	o	o
COMDLG32	o	o	o
GDI32	o	o	o
KERNEL32	o	o	o
OLE32	o	o	o
OLEAUT32	o	o	o
SHELL32	o	o	o
SHLWAPI		o	o
USER32	o	o	o
VERSION	o	o	o
WININET			o
WINMM			o

3.1.3 사용 빈도 60%의 소프트웨어 분류

세 번째는 소프트웨어 군별 5개의 제품에서 3개의 DLL 파일의 사용 빈도를 확인하였다. ADVAPI32.DLL 파일과 같이 전체 소프트웨어 군에서 사용되는 DLL파일이 있고, WS2_32.DLL 파일과 같이 FTP군에서만 사용되는 DLL파일이 확인 되었다. 이와 같이 각 소프트웨어 제품군별로 특정 DLL 파일들이 확인 할 수 있었다.

표 4. SW 분류별 사용 DLL - 사용 빈도 60%

DLL 파일명	FTP	Text	Media
ADVAPI32	o	o	o
COMCTL32	o	o	o
COMDLG32	o	o	o
CRYPT32	o		
GDI32	o	o	o
GDIPLUS		o	o
KERNEL32	o	o	o
OLE32	o	o	o
OLEAUT32	o	o	o
OLEDLG		o	
SHELL32	o	o	o
SHLWAPI		o	o
USER32	o	o	o
VERSION	o	o	o
WININET			o
WINMM	o	o	o
WINSPOOL		o	o
WS2_32	o		
WSOCK32	o		

3.2 DLL 파일 사용 빈도수에 따른 소프트웨어별 점수 계산

DLL파일의 사용 빈도수를 활용하여 각 소프트웨어 군별 점수를 계산하였다. FTP, Text, Media 세 제품군에서 각 5개씩 총 15개의 제품들에 대한 DLL 파일 연구를 진행하였다. 또한 이를 바탕으로 애플리케이션을 추가하여 분류가 가능한지 실험하였다.

표 5. 각 DLL파일 기본점수 부여(FTP 예시)

구분	FTP		
	100	80	60
구간별 DLL 파일 개수	8	10	14
DLL당 점수	12.5	10.0	7.1
구간별 총점	100	100	100

실험식 계산은 표 5와 같이 각 DLL 파일 점수를 추출하였다. 구간별 합을 100점으로 세 구간의 합을 300점으로 하여, 각 DLL 파일에 대해 사용 빈도별로 나누어 기본 점수를 부여하였다. 이를 바탕으로 애플리케이션이 갖고 있는 DLL 파일의 사용유무를 확인하여 총점을 매기고 가장 높은 점수가 나오는 소프트웨어를 확인하였다.

표 6과 같이 각 DLL 파일에 대한 점수를 기준으로 소프트웨어 분류가 가능한지 확인하였다.

표 6. filezilla에 대한 SW분류 예시

DLL 파일	FTP	TEXT	MEDIA
ADVAPI32	29.64	26.87	25.47
COMCTL32	29.64	26.87	14.36
COMDLG32	17.14	15.76	14.36
CRYPT32	7.14	2.86	2.70
GDI32	29.64	26.87	25.47
GDIPLUS	-	-	-
KERNEL32	29.64	26.87	25.47
OLE32	29.64	26.87	25.47
OLEAUT32	29.64	26.87	25.47
OLEDLG	-	-	-
SHELL32	29.64	26.87	25.47
SHLWAPI	-	-	-
USER32	29.64	26.87	25.47
VERSION	-	-	-
WININET	-	-	-
WINMM	7.14	6.67	25.47
WINSPOOL	-	-	-
WS2_32	7.14	2.86	2.70
WSOCK32	7.14	2.86	2.70
합계	282.86	245.95	240.59
분류확인	FTP	-	-

3.3 SW별 분류 확인

다음은 각 소프트웨어 군별 애플리케이션 15개 모두를 확인한 결과이다.

표 7. 실험 소프트웨어 분류목록

분류	프로그램명	버전	분류
FTP	ALFTP	5.3.2	O
	FileZilla	3.6.0	O
	WinSCP	5.1.4	O
	ALDrive	1.0	O
	CuteFTP	9.0	O
Text	AkelPad	4.8.2	X
	edtitplus	3.51	O
	wordpad	5.7	X
	HWP	2010	O
	notepad	2010	O
Media	GomPlayer	2.2.5	O

	KMPlayer	3.6.0	O
	Mplayer	6.4.9	O
	JetAudio	8.0.1	O
	Winamp	5.6.3	X

분류 결과 FTP는 소프트웨어 분류가 모두 성공하였으며, Text는 Aelpad와 wordpad를 Media는 winamp를 분류하지 못했다. Text의 wordpad의 경우 DLL파일의 구성이 중복되는 DLL파일을 모두 포함하고 있어 점수의 차이가 크게 나지 않는 이유도 있었다.

3.4 추가된 소프트웨어에 대한 분류

분석된 DLL파일을 기준으로 새로운 소프트웨어들을 대상으로 분류실험을 진행하였다. 추가된 프로그램은 FTP Cilent군에서 uTorrent, Putty, Ndrive, emule이 있고, Text editor군은 cutePDF를 Media군은 Tokplayer, NaverMedaPlyaer, ChocoPlayer 총 8개의 애플리케이션이다.

표 8. 추가 소프트웨어 목록

분류	프로그램명	버전	검출
FTP	uTorrent	3.4.0	O
	Putty	0.62	O
	Ndrive	1.4.2	O
	emule	0.50	X
Text	cutePDF	3.7	O
Media	TokPlayer	1.0.0.2	O
	NaverMediaPlayer	1.0.6.4	O
	ChocoPlayer	1.1.13.0	X

8개의 애플리케이션에 대해 분류 작업을 진행한 결과 FTP와 Media군에서 각 한 개씩을 제외한 6개의 프로그램의 분류를 성공하였다.

3.5 소프트웨어 분류 결과

기존의 15개의 제품들을 기준으로 추출된 DLL 파일을 대상으로 분류가능한 점수표를 만들고, 이를 통해 실제로 각 제품군별 분류실험을 진행하였다. 또한, 기존의 데이터를 기준으로 새로운 추가 소프트웨어를 추가 한 실험을 진행하였다.

그 결과 그림 2과 같이 각각 80%와 75%의 분류 성공률을 보였다.

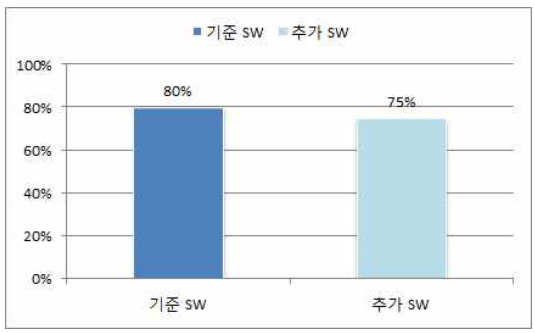

그림 2. 소프트웨어 분류 성공률

4. 결 론

본 논문에서는 MS 윈도우 환경에서 실행파일들에 대한 DLL 정보를 기준으로 유사한 소프트웨어를 분류하는 기법을 제안하였다. 그 결과 15개의 기본 실험군의 경우 12개의 분류가 성공하였고, 추가로 8개의 프로그램을 분류하는데 6개의 분류가 성공하였다. 이는 DLL 기반의 애플리케이션 특성이 각 소프트웨어를 분류하는데 사용될 수 있음을 보여 준다.

본 논문은 다양한 애플리케이션들을 대상으로 소프트웨어를 분류하는 방법을 제시하였으며, 향후 프로그램들 간의 유사성 분석에도 적용할 수

있다. 일부 애플리케이션의 경우 참조하는 DLL 개수가 소수이거나 또는 배포 버전이 패킹되어 있어 DLL 정보의 검출이 어려워 본 논문에서 제안한 기법을 적용할 수 없었다.

향후, 패킹이나 난독화가 적용된 소프트웨어도 분류하는 방법에 대해서도 연구할 예정이며, 실험 대상 소프트웨어 군도 압축 프로그램 그룹, 이미지 뷰어 그룹, 백신 그룹, 메신저 그룹 등으로 확장할 예정이다. 또한, 소프트웨어 분류의 성공률을 높이기 위한 효과적인 특징 정보 추출에 대한 연구도 진행할 계획이다.

참 고 문 헌

[1] 한국소프트웨어진흥원 “2012 공개소프트웨어 백서” 2013

[2] 최종천, 한용만, 조성제, 유해영, “윈도우용 이진실행 프로그램 식별을 위한 정적 버스마크”, 한국 소프트웨어 감정 평가학회 논문지 제8권, 제2호, 2012. 12.

[3] 박희완, 한태숙, “소프트웨어 몽타주: 디지털 포렌식 수사를 위한 유사 소프트웨어 탐지 대상의 필터링”, 정보과학회논문지: 컴퓨팅의 실제 및 레터, 제16권 제4호, 2010. 04.

[4] Haruaki Tamada, Keiji Okamoto, Masahide Nakamura, Akito Monden, and Ken-ichi Matsumoto, “Dynamic Software Birthmarks to Detect the Theft of Windows Applications”, In Proc. International Symposium on Future Software Technology 2004 (ISFST 2004), October 2004.

[5] Haruaki Tamada, Masahide Nakamura, Akito Monden, Ken-ichi Matsumoto, "Java birthmark-detecting the software theft," IEICE Trans. on Info. & Syst, vol.E88-D, no.9, 2148- 2158. Sept. 2005.

[6] 최성하, 박희완, “소프트웨어 몽타주 기반의 안드로이드 애플리케이션 자동 분류 기법”, 정보과학회논문지: 컴퓨팅의 실제 및 레터, 제18권 제11호, 756-761, 2012. 11

[7] 황윤하, 한용만, 김동진, 조성제, 유해영, 우진운, “유사한 MS 윈도우 애플리케이션 탐지를 위한 소프트웨어 분류에 관한 연구”, 제18회 한국SW감정평가학회 춘계학술대회, 2013. 05.

저 자 소 개

한 용 만

2011년 단국대학교 컴퓨터과학과 학사

2012년 단국대학교 컴퓨터학과 공학석사

2012년 9월～현재 : 단국대학교 컴퓨터학과 컴퓨터과학전공 박사과정

<관심분야 : 컴퓨터보안, 소프트웨어 분류, 소프트웨어 필터링, 소프트웨어 공학, 소프트웨어 품질, 모바일 소프트웨어>

황 윤 하

2013년 단국대학교 컴퓨터공학과 학사

2013년 3월～현재 단국대학교 컴퓨터학과 소프트웨어보안전공 석사과정

<관심분야 : 컴퓨터보안, 소프트웨어 공학, 소프트웨어 품질>

저 자 소 개

김 동 진

2009년 단국대학교 컴퓨터과학과 학사
2011년 단국대학교 컴퓨터학과 공학석사
2011년 9월～현재 : 단국대학교 컴퓨터학과 컴퓨터과학전공 박사과정
<관심분야 : 컴퓨터보안, 소프트웨어 보증, 보안 테스팅, 시스템소프트웨어 등>

최 종 천

1995년 강릉대학교 통계학과 학사
2005년 단국대학교 정보컴퓨터과학과 이학석사
2005년 3월～현재 : 단국대학교 정보컴퓨터과학과 박사과정
<관심분야 : 컴퓨터보안, 시스템소프트웨어, 소프트웨어보호, 임베디드시스템 등>

조 성 제

1989년 서울대학교 컴퓨터공학과 공학사
1991년 서울대학교 컴퓨터공학과 공학석사
1996년 서울대학교 컴퓨터공학과 공학박사
2001년 미국 University of California, Irvine 객원연구원
2009년 미국 University of Cincinnati 객원연구원
1997년 3월～현재 : 단국대학교 소프트웨어학과/컴퓨터학과 교수
<관심분야 : 컴퓨터보안, 소프트웨어 감정, 소프트웨어 보증, 시스템소프트웨어, 실시간스케줄링, 임베디드 소프트웨어 등>

유 해 영

1979년 단국대학교 수학과 학사
1981년 단국대학교 수학과 이학석사
1994년 아주대학교 컴퓨터공학과 공학박사
1983년 3월～현재 : 단국대학교 소프트웨어학과 교수
<관심분야 : 소프트웨어 보증, 소프트웨어 공학, 웹공학, 서비스 및 콘텐츠 정책 등>

우 진 운

1980년 서울대학교 수학교육과 학사
1989년 미국 University of Minnesota 전산학과 박사
1989년 3월～현재 단국대학교 소프트웨어학과 교수
<관심분야 : 소프트웨어 보증, 소프트웨어 감정, 알고리즘 등>