J. Internet Comput. Serv.
ISSN 1598-0170 (Print) / ISSN 2287-1136 (Online)
http://www.jics.or.kr

은닉형 Vault 안티포렌식 앱 탐색을 위한 XML 기반 특징점 추출 방법론 연구

A Study on the Feature Point Extraction Methodology based on XML for Searching Hidden Vault Anti-Forensics Apps

김 대 규[1] 김 창 수[1*]
Dae-gyu Kim Chang-soo Kim

요 약

스마트폰 앱을 사용하는 일반 사용자들은 개인이 소유하고 있는 사진, 동영상 등 개인정보를 보호하기 위해 Vault 앱을 많이 사용하고 있다. 그러나 범죄자들은 불법 영상들을 은닉하기 위해 Vault 앱 기능을 안티포렌식 용도로 악용하는 사례가 증가하고 있다. 이러한 앱들은 구글 플레이에 정상적으로 등록된 매우 많은 앱들 중 하나이다. 본 연구는 범죄자들이 이용하고 있는 Vault 앱들을 탐색하기 위해 XML 기반의 핵심어 빈도 분석을 통해 특징점을 추출하는 방법론을 제안하며, 특징점 추출을 위해서는 텍스트마이닝 기법을 적용한다. 본 연구에서는 은닉형 Vault 안티포렌식 앱과 비은닉형 Vault 앱 각각 15개를 대상으로 앱에 포함된 strings.xml 파일을 활용하여 XML 구문을 비교 분석하였다. 은닉형 Vault 안티포렌식 앱에서는 불용어처리를 1차, 2차 거듭할수록 더 많은 은닉 관련 단어가 높은 빈도로 발견된다. 본 연구는 공학 기술적인 관점에서 APK 파일을 정적 분석하는 대부분의 기존 방식과는 다르게 인문사회학적인 관점에서 접근하여 안티포렌식 앱을 분류해내는 특징점을 찾아내었다는 것에 의의가 있다. 결론적으로 XML 구문 분석을 통해 텍스트마이닝 기법을 적용하면 은닉형 Vault 안티포렌식 앱을 탐색하기 위한 기초 자료로 활용할 수 있다.

☞ 주제어 : 은닉형 Vault 앱, 안티포렌식, XML, 텍스트마이닝, 핵심어 빈도 분석

ABSTRACT

General users who use smartphone apps often use the Vault app to protect personal information such as photos and videos owned by individuals. However, there are increasing cases of criminals using the Vault app function for anti-forensic purposes to hide illegal videos. These apps are one of the apps registered on Google Play. This paper proposes a methodology for extracting feature points through XML-based keyword frequency analysis to explore Vault apps used by criminals, and text mining techniques are applied to extract feature points. In this paper, XML syntax was compared and analyzed using strings.xml files included in the app for 15 hidden Vault anti-forensics apps and non-hidden Vault apps, respectively. In hidden Vault anti-forensics apps, more hidden-related words are found at a higher frequency in the first and second rounds of terminology processing. Unlike most conventional methods of static analysis of APK files from an engineering point of view, this paper is meaningful in that it approached from a humanities and sociological point of view to find a feature of classifying anti-forensics apps. In conclusion, applying text mining techniques through XML parsing can be used as basic data for exploring hidden Vault anti-forensics apps.

☞ keyword : Hidden Vault Application, Anti-forensics, XML, Text Mining, Keyword Frequency Analysis

1. 서 론

스마트폰에는 대부분의 개인 사생활 관련 정보가 저장되어있다고 해도 과언이 아닐 정도로 수많은 개인정보가 저장되어있다. 이에 따라 구글 플레이에는 개인정보를 보호할 목적으로 개발된 앱이 많다. 이러한 앱이 개발 목적에 맞게 사용되면 문제없지만 불법을 저지르는 범죄자에게는 유용한 은닉형 Vault 안티포렌식 앱이 된다. 예를 들어 지하철에서 불법촬영을 하다가 현행범으로 체포되어 압수한 스마트폰에서 불법촬영 사진과 동영상을 압수하고자 할 때 범인이 은닉형 안티포렌식 앱으로 증거 파일을 숨기거나 암호화해놓으면 증거 확보를 하기 어렵다[1].

수사는 신속성이 중요하기 때문에 해당 앱에 대한 분석이 사전에 되어있지 않다면 수사가 계속 진행되기 어려운 상황에 봉착하게 된다. 이를 해결하기 위해 사전에

1 Department of IT Convergence and Application Engineering, Pukyong National University, Busan City, 48513, Korea.
* Corresponding author (cskim@pknu.ac.kr)
[Received 5 January 2022, Reviewed 13 January 2022(R2 8 February 2022), Accepted 10 February 2022]

은닉형 안티포렌식 앱으로 활용될 수 있는 앱을 선별하여 해당 앱의 특성을 분석하고 숨김 해제나 복호화 기법을 개발해놓는 것은 중요한 수사기법으로 활용될 수 있을 것이다[2]. 이를 위해 본 논문에서는 '은닉형 Vault 안티포렌식 앱 탐색을 위한 XML 기반 특징점 추출 방법론 연구'를 제안하고자 한다. 대상 앱은 구글 플레이스토어에 등록되어있는 앱들이며, 은닉형 Vault 안티포렌식 앱을 분석하여 특징점을 파악하고, 하루에도 수없이 개발 및 배포되는 앱들 중, 은닉 기능을 가지고 있는 앱을 사전 탐지하여 범죄에 이용되었을 경우에 수사기관에서는 신속하게 대응할 수 있도록 정보를 제공하는데 기여하고자 한다.

본 논문의 구성은 다음과 같다. 2장에서는 구글 플레이 Vault 앱의 특징과 은닉형 안티포렌식 기존 연구를 소개한다. 3장에서는 은닉형 Vault 안티포렌식 앱 탐색을 위한 특징점 추출 방법을 설명한다. 4장에서는 논문의 의의, 한계점, 향후 연구 방향을 제시하고자 한다.

2. 안티포렌식 앱 탐색 관련 연구

2.1 구글 플레이 Vault 앱의 특징

구글 플레이(Google Play)는 구글에서 운영 중인 안드로이드 앱, 음악, 동영상 등을 제공하는 디지털 콘텐츠 서비스이다. 구글 플레이에는 매일 수천 명의 개발자가 개발한 수백만 개의 응용 프로그램 앱을 업로드한다.

안드로이드 OS가 설치된 스마트폰 사용자는 스마트폰에 저장된 사생활 관련 사진, 동영상 파일 등을 숨기거나 보호하기 위해 개발된 Vault 앱을 유료 또는 무료로 구글 플레이에서 합법적으로 다운로드 받아 설치할 수 있다.

Vault 앱은 사용자의 개인정보보호를 목적으로 개발되었기 때문에 기본적으로 사진, 동영상, 파일 등을 다른 폴더로 이동하여 숨기거나 숨긴 파일을 볼 수 없도록 암호화하는 기능을 갖고 있다. 숨긴 파일의 저장 장소는 스마트폰이 아닌 클라우드에 저장하는 방식도 있다. 사용자 로그인 없이 사용할 수 있는 앱도 있지만 대부분 이메일로 사용자 인증을 받아 이용자를 구분하는 앱이 많다.

2.2 은닉형 안티포렌식 기존 연구

기존 안티포렌식 앱 관련 연구는 디지털포렌식 관점으로 안티포렌식 앱(APK; Android Package)[3]의 은닉 및 암호화 기능 분석을 통한 안티포렌식 대응기법에 관한 논문이 대부분이다.

Soojin Kang et al.[4]은 디지털 포렌식 관점에서 사진 잠금, 사진/비디오 숨기기, Safe GalleryCamera 앱을 분석하여 은닉 기법을 분석, 해제 방법 제시 및 암호화된 데이터를 복호화하는 방법을 연구하였다. Xiaolu Zhanga et al.[5]은 구글 플레이에 있는 은닉(Vault) 앱 18개를 대상으로 은닉된 사진과 동영상을 복구하는 방법, 앱에 우회 접근하는 방법, 은닉 파일의 암호화 방식 등을 연구하였다.

텍스트마이닝 기법은 주로 문헌정보나 경영정보 등 인문학이나 인문사회학 분야에 널리 사용되는 기법이며, 비정형데이터인 텍스트 기반 빅데이터를 바탕으로 새로운 기준으로 카테고리를 만들어내고 의미를 부여하는 방식으로 연구자의 주관적인 판단이 중요한 역할을 한다.

Jiyong Park et al.의 연구[6]에서는 2020년 미국 대통령 선거 후보가 대한민국 언론에 끼치는 영향을 텍스트 마이닝 기법을 통해 분석하였다. Young-Man Kim and Jin Gu Lee의 연구[7]에서는 은 텍스트 마이닝 기법을 활용하여 유리천장 관련 학위논문 및 학술지논문의 연구 동향을 분석하였다.

Do-Hee Kim and Min-Jeong Kim의 연구[8]에서는 텍스트마이닝 기법을 사용하여 1990년부터 2019년까지 5,141건의 신문기사에서 '허위·과장광고' 용어의 트렌드를 분석하였다. Jufri and Musdalifa Thamrin[9]은 소셜미디어(SNS)가 정치에 미치는 영향에 대해 자료를 수집하고 텍스트 마이닝 기법으로 분석하였다. 이처럼 텍스트 마이닝은 다양한 분야의 뉴스나 문헌, SNS를 연구하는데 많이 활용되고 있다.

APK 파일을 분류하는 연구에는 APK에서 추출된 디컴파일 된 클래스는 dalvik Android 가상머신(VM) 용 바이트 코드의 간단한 어셈블리 표현인 smali 코드 형식을 이용한 연구가 있다. .smali 파일은 읽기 쉽고 분석하기 쉬운 형식으로 되어 있어 다양한 연구에서 많이 사용된다.

Junseok Cheon et al. 연구[10]에서는 .smali 코드 파일 간의 유사도 점수를 계산하는 방식을 이용하여 APK 파일을 분류하였다. Konstantinos F. Xylogiannopoulos et al. 연구[11]에서는 APK 파일의 smali 코드 기반으로 소스 코드에 주입(injected)되어 있는 악성코드 구문을 텍스트마이닝 기법을 적용하여 악성코드 분류하는 방법을 연구하였다. 여기서 smali 코드는 간단한 어셈블리 언어이기 때문에 인간의 언어와는 다른 방식으로 텍스트마이닝 기법이 적용된다. 단어와 구를 사용하는 대신에 다양한 기호와 매우 긴 단어가 포함된 명령어 또는 프로그램 구조를 이용한다. 본 연구와 유사한 관점을 갖고 있는 연구이지만 인문사회학적으로 접근한 본 연구와는 차이가 있다.

Daegyu Kim and Changsoo Kim의 연구[12]에서는 5개의 은닉형 Vault 안티포렌식 앱에 대해 strings.xml 파일에서 단순히 'lock', 'vault' 등과 같이 직관적인 방식으로 공통된 은닉 관련 단어를 발견했을 뿐이므로, 좀 더 많은 수의 은닉형 기능을 가진 앱들과 더 많은 연관키워드 산출 방법에 대해 연구할 필요가 있다. 그리고 비은닉형 앱과의 비교를 통해 분류 기준으로서 의미가 있는지에 대해서도 추가 연구가 필요하다. 본 논문은 이러한 한계점을 보완한 XML 기반 특징점 추출 방법을 제안한다.

3. 은닉형 Vault 안티포렌식 앱 탐색을 위한 특징점 추출 방법

3.1 은닉형 Vault 안티포렌식 앱의 XML 특성 파악을 위한 검색 방법

구글 플레이에 등록되어 있는 은닉형 Vault 안티포렌식 앱을 스마트폰이나 가상 에뮬레이터(Nox, Bluestacks 등)에 다운로드 받아 APK 파일을 추출한다. 추출한 APK 파일은 디컴파일(decompile)하여 XML 파일을 추출하고, 태그 내에 있는 문장, 단어 등을 추출한다. 이렇게 추출한 텍스트를 텍스트 마이닝 기법을 적용하여 특징점을 범위를 결정한다.

3.1.1 안드로이드 앱(APK)

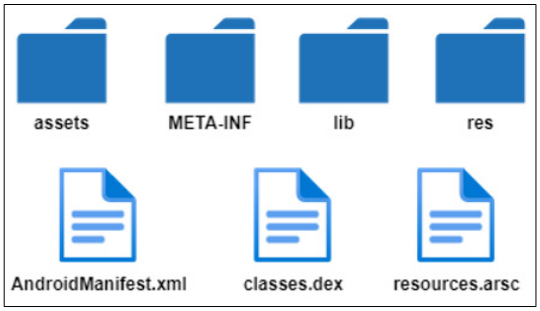
안드로이드(Android)는 주로 스마트폰 및 태블릿과 같은 터치스크린 모바일 장치용으로 설계된 Linux 커널 및 기타 오픈 소스 소프트웨어의 수정된 버전을 기반으로 하는 모바일 운영 체제이다. Android는 Open Handset Alliance로 알려진 개발자 컨소시엄에서 개발하고 Google이 상업적으로 후원하고 있으며 2007년 11월에 공개되었다[13].

APK는 Android 운영 체제 및 모바일 앱, 모바일 게임 및 미들웨어의 배포 및 설치를 위한 기타 여러 Android 기반 운영 체제에서 사용하는 Android 애플리케이션 패키지 파일 형식이다[14].

APK 파일은 일반적으로 파일과 디렉토리를 포함하는 아카이브이다. 파일 및 디렉토리 구성은 다음 그림 1과 같다.

assets 디렉토리는 AssetManager를 통해 검색 할 수 있는 응용 프로그램의 정보를 포함하는 디렉토리이다.

META-INF 디렉토리는 MANIFEST.MF 매니페스트 파일(Manifest file), CERT.RSA 전자 서명 파일, CERT.SF 파일을



(그림 1) APK 파일 구성
(Figure 1) Structure of APK file

포함하고 있는 디렉토리이고, lib 디렉토리는 프로세서의 소프트웨어 레이어와 관련된 컴파일된 코드를 포함하는 디렉토리이다.

res 디렉토리는 resources.arse로 컴파일되지 않는 리소스를 포함하는 디렉토리이다.

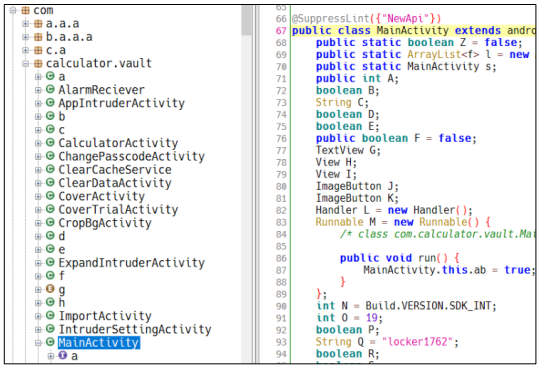
Androidmanifest.xml 파일은 핵심 안드로이드 매니페스트 파일(manifest file)이 포함되고, 이 파일에는 앱의 이름, 버전, 액세스 권한 및 참조된 라이브러리 파일이 나열된다.

classes.dex 파일은 Dalvik/ART 가상 머신에서 인식하는 DEX 파일 형식으로 컴파일된 클래스를 포함한다.

resources.arsc 파일은 컴파일된 리소스를 포함한다. 이 파일에는 res/values/ 폴더에 있는 모든 구성의 XML 콘텐츠가 포함된다. 이 콘텐츠에는 언어 문자열 및 스타일뿐만 아니라 레이아웃 파일처럼 resources.arsc 파일에 직접 포함되지 않은 콘텐츠 파일의 경로가 포함된다[15].

3.1.2 APK 디컴파일(decompile)

APK 파일을 디컴파일하면, JAVA 클래스 파일로 변환시켜 JAVA 소스코드를 그대로 볼 수 있다. 그래서 개발자들은 소스코드 노출을 방지하기 위해 난독화 기술을 적용한다. APK에 적용하는 난독화 도구는 안드로이드 개발도구인 Android Studio에서 기본적으로 제공하는 프로가드(Proguard)가 주로 사용된다[16]. 코드 난독화의 목적은 앱 클래스, 메서드 및 필드의 이름을 단축하여 앱 크기를 줄이는 것이다. 물론 역공학(reverse engineering)에 의해 바이너리나 소스코드가 분석되는 것을 어렵게 하기 위한 목적도 있다[17]. 다음 그림 2는 난독화 된 JAVA 소스코드 예시이다.



(그림 2) 난독화 된 JAVA 소스코드 예시
(Figure 2) An example of obfuscated JAVA source code

이처럼 난독화를 적용하면 JAVA 클래스명과 JAVA 소스코드는 대부분 난독화 되어 'a', 'b', 'c', 'd'와 같이 알아볼 수 없도록 되어있는 경우가 많기 때문에 유의미한 정보를 얻어내기 위해서는 개발자가 난독화 시키기 어려운 부분을 찾아야 한다.

3.1.3 XML 구문 분석

XML(eXtensible Markup Language)은 확장성 있는 마크업 언어로서, 데이터의 정의 및 수정이 용이하다는 장점이 있다. APK 파일은 XML 파일을 포함하고 있는데, XML 파일은 JAVA 소스코드와 다르게 대부분 난독화가 되어있지 않고 리소스(resource)에 포함되어 있다. 리소스는 JAVA 코드에서 사용하는 추가 파일과 정적인 컨텐츠를 말한다. 리소스 디렉토리(res/)는 'animator/', 'anim/', 'color/', 'drawable/', 'mipmap/', 'layout/', 'menu/', 'raw/', 'values/', 'xml/', 'font/' 등 모든 리소스를 하위 디렉토리에 포함하고 있다[18].

이 중에 'values/' 디렉토리에 있는 strings.xml은 사용자에게 문구를 보여줄 때 이용해야하는 구문들이 저장되어 있는 파일로서 기능적인 특성상 난독화 되어있지 않다.

strings.xml 파일은 보통 'res/values/strings.xml'에 선언해서 R.string.name처럼 사용한다. 개발자가 특별히 문자열을 java 소스코드에 직접 하드코딩을 하지 않는 한, strings.xml 파일에 애플리케이션 또는 다른 리소스에서 참조할 수 있는 단일 문자열을 선언해놓고 이를 불러와 사용한다.

본 논문에서는 위와 같은 strings.xml이 은닉 앱 분류를 위한 주요 특징점으로서 활용할 수 있는지에 대해 연구한다.

3.1.4 은닉형 Vault 안티포렌식 앱

본 논문의 연구 대상 은닉형 Vault 안티포렌식 앱은 다음 표 1과 같다. 대상 앱 15개는 모두 다 개인정보보호를 목적으로 제작된 앱으로서 주요 기능에는 파일, 사진, 동영상, 파일 등을 잠금 및 잠금해제하는 기능이 있다. (이하 은닉형 Vault 안티포렌식 앱의 No. 1은 VA1으로, No. 2는 VA2와 같이 표기한다.)

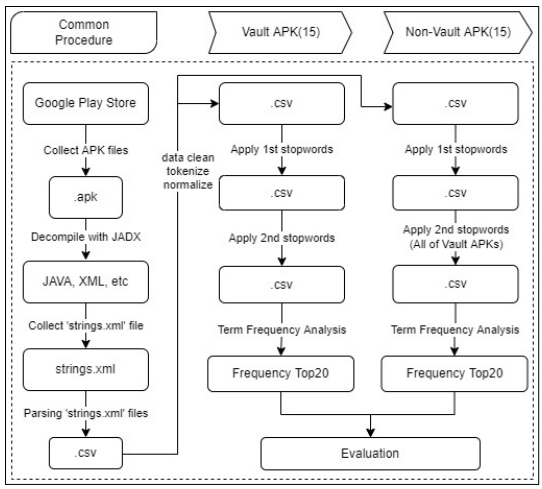
(표 1) 은닉형 Vault 안티포렌식 앱 목록
(Table 1) Vault Anti-Forensic Application List

No	Application	Version
1	AppLock Lock Apps	3.2
2	AppLock	2.8.10
3	Calculator Vault Gallery Lock	15.0
4	Gallery Lock Hide pictures	5.0.4
5	Gallery Vault Hide Pictures And Videos	3.14.22
6	Hide Photos in Photo Locker	2.1.0
7	Hide Photos Video and App Lock Hide it Pro	8.0.5
8	Hide Pictures Videos Vaulty	4.8.17
9	HidePhoto	1.21
10	Keeper Password Manager	14.2.4.2
11	Keepsafe Photo Vault Hide Private Photos Videos	9.36.0
12	LOCX Applock Lock Apps Photo	2.3.3
13	Pic Lock Hide Photos Videos	3.1
14	Private Photo Vault	2.3.1
15	Private Zone AppLock Video Photo Vault	5.9.6

3.2 특징점 추출을 위한 텍스트마이닝 알고리즘

텍스트마이닝이란 텍스트 형태의 비정형 데이터로부터 의미있는 정보를 찾는 것을 말한다. 텍스트마이닝은 1) 텍스트 수집(Collection), 2) 텍스트 정제(Natural Language Processing), 3) 텍스트 분석(Analysis), 4) 평가(Evaluation) 단계로 진행된다. 국가별 언어에 따라 문법적으로 서로 다른 특징을 갖고 있기 때문에 텍스트에 대한 수집, 정제, 분석 방법이 다를 수 있다. 형태소(morpheme) 분석의 경우 영어는 공백 단위로 쉽게 구분할 수 있지만[19], 한국어는 영어와 달리 형태소가 어절을 이루는 형태이기 때문에 형태소 구분 방법이 복잡하다[20]. 본 논문에서는 글로벌 언어인 영어만을 대상으로 연구를 진행한다.

분석 절차는 다음 그림 3과 같다.



(그림 3) 분석 절차
(Figure 3) Analysis process

텍스트 수집은 연구 대상 APK 파일을 JADX(Dex to Java Decompiler) 도구로 디컴파일하여 'res/values/' 폴더에 있는 strings.xml을 추출한다[21].

본 논문에서는 텍스트 정제 단계를 1) 사전 정제(data cleaning), 2) 토근화(tokenization), 3) 정규화(normalization) 4) 불용어(stopwords) 제거 총 4단계로 나누어 진행한다.

1) 사전 정제 단계

텍스트 데이터는 비정형 데이터이므로 불필요한 사항을 제거해서 의미 있는 데이터만 추출하는 사전 정제 단계를 거쳐야한다.

예를 들어, 'What if you forgot your Calculator password &amp \n ;'와 같은 텍스트 데이터가 있을 경우 '&, \, ;'는 별다른 의미를 갖지 않는 기호이다. 문장의 의미를 이해하는데 필요 없는 기호나 문구는 제거해야 데이터를 효과적으로 처리할 수 있다. 영어 텍스트의 경우에는 보통 대문자와 소문자를 통일시켜서 처리하는데 본 논문에서는 소문자로 통일시켜 처리한다. 데이터를 정제하기 위해서는 정규 표현식(Regular expression, Regex)를 많이 사용한다. 본 논문에서는 파이썬(Python)의 re 패키지를 이용한다. 다음으로 JADX를 이용하여 strings.xml 파일을 추출하고, 추출한 텍스트 데이터를 파이썬의 xml.etree 패키지(xml.etree.ElementTree)를 활용하여 XML 구문을 분석한다[22].

2) 토큰화 단계

토큰화는 텍스트 데이터를 분석하기 좋은 형태로 나누는 과정이다. 즉, 문장을 구성 요소로 나누는 것을 말한다. 토큰화를 진행하기 위해서는 사전 정제 단계를 거쳐야 한다. 토큰은 '하나의 유용한 의미적 단위로 함께 모여 있는 일련의 문자열'이다[23, pp.21]. 토큰은 형태소일 수도 있고, 하나의 단어일 수도 있고, 두 단어 이상이 결합된 복합어일 수도 있다. 형태소는 더 이상의 분석이 불가능한, 의미의 최소 단위이다. 토큰화 작업은 nltk 패키지의 word_tokenize를 이용하였다[24].

3) 정규화 단계

정규화 단계는 형태소 분석을 하고 불규칙한 변화를 통일된 형태로 추출하는 과정이다. 형태소 분석은 "단어를 구성하는 각각의 형태소를 인식하고 용언의 활용, 불규칙 활용이나 축약 탈락현상이 일어난 형태소는 원형으로 복원하는 과정"이다[23, pp.48].

다음은 어간 추출(stemming, 형태소 분석)로써 영어와 같이 단어가 다양한 형태로 변형되는 경우 기본 형태로 바꿔줄 수 있다. 하지만 예를 들어 'application'이라는 단어의 어간을 추출하면 'applic'으로 변환되어 의미를 갖지 않는 단어가 추출될 수 있으므로 본 연구에서는 사용하지 않는다. 위와 같은 사항을 극복하기 위해 어간 추출 대신에 nltk 패키지의 WordNetLemmatizer를 이용하여 표제어를 추출(lemmatization)한다. 표제어 추출은 형태소 분석을 통해 더 정확한 단어수준 분석을 수행하여 품사정보가 보존된 형태의 기본형으로 변환하는 방법이다[25, pp.146].

4) 불용어 제거 단계

문장에서 실질적인 의미를 가지고 있지 않은 기능어(an, an, the 등; 관사, 전치사)을 불용어(stopword)로 제거하는 단계이다.

본 논문에서는 불용어 제거를 위해 먼저 nltk의 english stopwords로 1차로 불용어 제거하고, 단어 길이가 3 이상인 단어만으로 필터링하였다. 단어 길이가 1이면 의미를 부여할 수 없고, 2인 단어는 'go', 'to' 등 은닉을 표현하는 단어로서는 의미가 없으므로 제외하였다. 그리고 '%1$s', '%2$d' 등은 strings.xml에 정의된 문자열 리소스에서 매핑 시켜주기 위한 변수이므로 의미를 부여하기 어렵다. 따라서 re(Regular Expression)을 이용해 영문자를 제외한 나머지 의미 없는 문자는 모두 제외하였다.

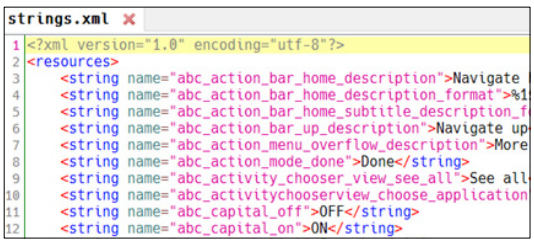
텍스트 분석 단계에는 핵심어 빈도분석, 단어 빈도분석, 군집분석, 토픽 모델링, 감성 분석, 연관어 분석 등 다양한 방법들이 있는데, 본 논문에서는 텍스트마이닝의 핵심어 빈도분석을 통한 특징점 추출 방법을 연구한다.

평가 단계에서는 은닉형 Vault 안티포렌식 앱과 비은닉형 안티포렌식 앱에 대한 핵심어 빈도분석 결과를 비교하여 특징점으로서 활용 가능한지에 대해 평가한다.

3.3 특징점 추출을 위한 XML 구문 분석

본 논문에서는 은닉형 Vault 안티포렌식 앱 15개 내에 존재하는 strings.xml 구문 분석을 수행한다.

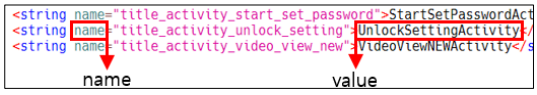
strings.xml 파일에는 각 <name> 요소에 문자열 값(value)이 지정되어있고, 단어수준(word level)으로 구분하는 방식을 적용하여 구문을 분석한다. 그림 4는 APK 디컴파일러 JADX를 이용하여 살펴본 strings.xml의 첫 부분이다. 문자열 값은 'Navgate up', 'Done', 'See all', 'OFF', 'ON' 등과 같이 난독화 되어있지 않은 것을 확인할 수 있다.



(그림 4) JADX로 디컴파일한 strings.xml 예시
(Figure 4) Example strings.xml decompiled with JADX

<resources> 태그는 최상위 루트 요소(element, tag)이고, <string> 태그 안에 스타일 지정 태그를 포함할 수 있다.

다음 그림 5와 같이 <name> 요소마다 자주 사용하고자 하는 문자열을 값(value)으로 지정해서 사용하는 것이 일반적이다.



(그림 5) strings.xml 〈name〉, 문자열 값
(Figure 5) strings.xml 〈name〉, string value

본 논문에서는 <name> 요소의 문자열 값을 추출하여 은닉형 기능 앱의 특징점을 찾아내고자 한다.

텍스트마이닝 기법인 핵심어 빈도분석이란 불용어 제거와 형태소 분석, 어간 추출 등의 자연어 처리를 시행한 후 텍스트에서 많이 등장하는 단어의 등장 빈도를 분석함으로써 핵심어를 추출하는 것이다. 이는 특정 텍스트에 많이 나타나는 단어가 그 텍스트 주제를 표출할 가능성이 높다는 가정에 기초한다. 그리고 핵심어(keyword)란 텍스트 자료의 중요한 내용을 함축적으로 표현하는 단어나 문구를 말한다[25, pp. 167].

핵심어 빈도분석은 텍스트마이닝의 가장 기본적인 분석방법으로써 핵심어 빈도분석 방법에는 단순 빈도를 제시하는 방법, 워드 클라우드(word cloud)로 빈도 분석 결과를 제시하는 방법이 있다[25, pp. 168].

본 논문에서는 은닉형 기능 앱과 비은닉형 기능 앱에 대한 핵심어 빈도분석을 통해 은닉형 기능 앱에서 주로 사용되는 단어(토큰)를 확인하고자 한다.

3.3.1 은닉형 Vault 앱 XML 구문 분석

은닉형 기능 앱 빈도수 분석을 진행하기 전에 핵심어와는 관련성을 가질 수 없는 '&, \, ;' 같은 특수 기호와 nltk 패키지에서 제공하는 영어(english) 불용어를 1차적으로 제거한 후, VA 15개 중에 3개(VA1 ~ VA3)만 선정하여 빈도분석(빈도 높은 순 top 20)을 한 결과는 다음 표 2와 같다.

(표 2) 1차 불용어 처리 후, Top 20 빈도분석 결과
(Table 2) Top 20 frequency analysis results after primary stopword processing

VA1		VA2		VA3	
Word	Count	Word	Count	Word	Count
lock	45	please	45	app	26
mode	35	lock	41	file	15
please	25	applock	34	calculator	14
privacy	22	password	33	pattern	13
password	21	security	31	serif	12
new	20	google	24	sans	12
- 중략 -					
font	10	apps	16	card	5

여기까지는 1차 불용어 처리만 한 것이므로 XML 구문에서 자주 사용되는 단어나 sans-serif 글자체 등 문장의 의미를 이해하는데 필요 없는 단어는 제거되지 않은 상태이다. 빈도수가 높은 단어(토큰)에는 'applock', 'gallery', 'lock', 'password', 'pattern', 'permission', 'photo', 'security', 'unlock', 'protection', 'vault', 'video', 'data', 'privacy', 'file', 'folder', 'locker', 'pic', 'hide', 'stealth', 'storage', 'galleryvault', 'pin'과 같이 은닉과 관련성 높은 단어가 존재한다.

다음으로 특수 기호나 nltk 패키지에서 제공하는 영어(english) 불용어 외에도 은닉과 관련 없는 단어를 2차 불용어로 추가한 후, 다시 빈도분석을 진행한다. 결과(빈도 높은 순 top 20)는 다음 표 3과 같다.

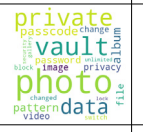
(표 3) 2차 불용어 처리 후, Top 20 빈도분석 결과
(Table 3) Top 20 frequency analysis results after secondary stopword processing

VA1		VA2		VA3	
Word	**Count**	**Word**	**Count**	**Word**	**Count**
lock	45	lock	41	file	15
privacy	22	applock	34	pattern	13
password	21	password	33	password	9
pattern	10	security	31	data	7
change	9	unlock	22	permission	7
protection	8	protection	22	lock	6
- 중략 -					
unlocked	2	locked	6	unhide	1

2차 불용어를 추가로 제거한 후에는 기존에 발견된 단어(토큰) 외에도 'access', 'album', 'alternating', 'change', 'hidden', 'image', 'keep', 'limit', 'limitation', 'locked', 'locking', 'passcode', 'picture', 'switch', 'private', 'prevent', 'unhided', 'unhide', 'hint', 'gallerylock', 'key', 'disguise', 'encrypted', 'camera', 'hiding', 'changed', 'recovery'와 같이 좀 더 다양한 은닉 관련 단어가 발견된다.

앞서 분석한 2개를 포함한 은닉형 Vault 안티포렌식 앱 15개 내에 존재하는 strings.xml을 1, 2차 불용어 처리한 뒤, 빈도분석 한 결과를 다음 표 4와 같이 워드 클라우드(wordcloud) 형태로 나타내보았다.

(표 4) 2차 불용어 처리 후, Top 20 워드클라우드
(Table 4) Top 20 Word Cloud results after 2nd stopword processing

VA 1	VA 2	VA 3	VA 4
			
VA 5	**VA 6**	**VA 7**	**VA 8**
			
VA 9	**VA 10**	**VA 11**	**VA 12**
			
VA 13	**VA 14**	**VA 15**	

3.3.2 비은닉형 Vault 앱 XML 구문 분석

비은닉형 앱(NVA; Non-Vault Application) 15개를 대상으로 은닉형 Vault 앱과 동일한 환경에서 빈도분석을 해보았다. 비은닉형 앱 목록은 다음 표 5와 같다. (이하 비은닉형 앱의 No. 1은 NVA1으로, No. 2는 NVA2와 같이 표기한다.)

(표 5) 비은닉형 앱(Non-Vault App) 목록
(Table 5) Non-Vault Application List

No	Application	Version
1	Duolingo Learn Languages Free	5.0.2
2	Google Translate	6.17.1.04.359877260
3	Cookpad Create your own Recipes	2.194.1.0
4	Seamless Restaurant Takeout Food Delivery App	7.155
5	Untappd Discover Beer	3.5.3
6	Calorie Counter MyFitnessPal	20.19.1
7	Spotify Listen to podcasts find music you love	8.6.10.581
8	Snapseed	2.19.1.303051424
9	Doodle Easy Scheduling	4.15.3
10	Khan Academy	6.8.0
11	ESPN	6.40.0
12	Feedly Smarter News Reader	77.0.2
13	Amazon Kindle	8.40.0.100
14	Libby by OverDrive	4.2.1
15	rif is fun for Reddit	5.0.7

다음으로 NVA 15개 내에 존재하는 strings.xml을 1차 불용어와 VA 15개 관련 2차 불용어를 모두 불용어 처리한 뒤 빈도분석을 한다. NVA 15개 중에 3개(NVA1 ~ NVA3)만 선정하여 빈도분석(빈도 높은 순 top 20)을 한 결과는 다음 표 6과 같다.

(표 6) 1, 2차 불용어 처리 후, Top 20 빈도분석 결과
(Table 6) Top 20 frequency analysis results after 1st, 2nd stopword processing

NVA1		NVA2		NVA3	
Word	Count	Word	Count	Word	Count
duolingo	60	translation	73	recipe	161
streak	50	speak	26	cookpad	76
lesson	46	camera	22	change	28
learning	40	chinese	10	cooksnap	23
heart	37	image	9	access	17
skill	36	transcribe	9	camera	16
- 중략 -					
span	12	change	5	challenge	7
crown	11	pair	5	proven	7

빈도분석한 결과, 빈도 높은 순 top20 단어(토큰)은 'duolingo', 'streak', 'lesson', 'learning', 'heart', 'skill', 'translation', 'speak', 'camera', 'chinese', 'image', 'transcribe', 'recipe', 'cookpad', 'change', 'cooksnap', 'access', 'camera' 등이 확인된다.

3.3.3 은닉형 Vault 앱과 비은닉형 Vault 앱 XML 구문 분석 결과 비교

은닉형 Vault 앱에서는 1차 불용어 처리를 했을 때보다 2차 불용어 처리를 했을 때 더 많은 은닉 관련 단어가 발견되었고, 비은닉형 Vault 앱에서는 1, 2차 불용어 처리를 한 결과 은닉형 Vault 앱에서 나온 단어보다는 적은 빈도수이지만 'keep', 'password', 'unlimited', 'image', 'camera', 'switch', 'hide', 'change', 'access', 'permission', 'video', 'photo', 'data', 'folder', 'changed', 'pin', 'security' 등 단어도 발견된다.

이는 해당 단어들은 은닉 관련 단어가 될 수도 있지만 은닉 기능을 갖지 않은 앱에서도 사용하는 단어임을 알 수 있다. 따라서 이러한 단어들은 3차 불용어 처리하여 은닉형 Vault 안티포렌식 앱에서만 사용되는 단어만을 확인할 수 있도록 범위를 줄여줄 수 있다.

본 논문에서 적용한 2차 불용어 처리까지만을 볼 때 연구 대상 은닉형 Vault 안티포렌식 앱 15개에서 높은 빈도수를 가지는 단어 중 은닉형 앱에서만 사용된 은닉 관련 단어는 'album', 'alternating', 'applock', 'block', 'disguise', 'encrypted', 'fake', 'file', 'gallery', 'gallerylock', 'galleryvault', 'hidden', 'hidephoto', 'hiding', 'hint', 'indicating', 'keeper', 'keepersecurity', 'keepsafe', 'key', 'limit', 'limitation', 'lock', 'locked', 'locker', 'locking', 'passcode', 'pattern', 'piclock', 'picture', 'prevent', 'privacy', 'private', 'protect', 'protection', 'recover', 'recovered', 'recovery', 'restore', 'restricted', 'secret', 'secure', 'stealth', 'storage', 'unhide', 'unhided', 'unlock', 'vault' 'vaulty' 이다.

결론적으로 본 연구를 통해 strings.xml 파일을 활용해 텍스트마이닝 기법을 적용하면 은닉형 Vault 안티포렌식 앱의 특징점으로 활용할 수 있음을 알 수 있다.

4. 결 론

본 논문은 공학 기술적인 관점에서 APK 파일을 정적 분석하는 대부분의 기존 방법과는 다르게 문헌 자료를 분석하는 용도로 사용되는 텍스트마이닝 기법을 활용하여 안티포렌식 앱을 분류해내는 특징점을 찾아내었다는 데 의의가 있다. APK 파일은 결국 사람이 사용하는 안드로이드 스마트폰에서 작동하도록 개발된 것이므로 사람이 읽을 수 있는 형태의 텍스트가 포함되지 않을 수 없다. 그리고 사람이 읽어야 할 텍스트를 난독화 할 필요는 없기 때문에 해당 텍스트를 추출하여 텍스트마이닝 기법을 적용할 수 있는 것이다.

특히, 본 연구 대상 앱들은 '구글 플레이'에서 다운로드 가능한 앱들이고 일반 사용자들의 개인정보를 보호할 목적으로 개발된 것이기 때문에 사용자의 편의성을 위해 기능에 알맞은 용어를 사용한다는 점도 분석에 이점을 준다.

본 연구의 방법론을 은닉형 Vault 안티포렌식 앱을 탐색하는 기법에 적용하게 되면, APK 파일을 단순히 디컴파일하는 것만으로도 쉽게 특징점을 찾아낼 수 있다는 점, 난독화 되어 있는 소스코드를 공학적으로 분석하여 탐색하는 것보다 더 쉽게 난독화를 우회하여 접근할 수 있는 점 등이 탐색의 효율성을 높일 수 있는 주요 포인트가 된다.

본 연구는 텍스트마이닝의 기초적인 분석방법인 빈도분석만을 통해 특징점을 찾아내었고, 더 많은 앱을 대상으로 연구하지 못했다는 한계점이 있다. 향후에는 크롤링 기법을 활용하여 더 많은 앱과 신규 앱들을 수집하고, 머신러닝을 이용하여 좀 더 정확하고 자동화된 은닉형 안티포렌식 앱 분류 기법을 연구할 예정이다.

참고문헌(Reference)

[1] "[Police Team 24/7] 'Trickery' that hides the shooting screen… 'Wits war' using big data", [Online]. Available: https://www.sedaily.com/NewsVIew/1VMZQY59CT

[2] Jongman Kim, Sangjin Lee, "Digital forensic framework for illegal footage -Focused On Android Smartphone-", Journal of Digital Forensics, Vol.12, No.3, pp.39-54, 2018. http://doi.org/10.22798/kdfs.2018.12.3.39

[3] "Application Fundamentals", [Online]. Available: https://developer.android.com/guide/components/fundamentals

[4] Soojin Kang, Sumin Shin, Giyoon Kim, Jongsung Kim, "Forensic Analysis of Locking Pictures, Hiding Photos/Video and Safe Gallery/Camera Applications", Journal of Digital Forensics Vol 14, No. 2, pp.125-137, 2020. http://doi.org/10.22798/KDFS.2020.14.2.125

[5] Xiaolu Zhanga, Ibrahim Baggilia, Frank Breitingera, "Breaking into the vault: Privacy, security and forensic analysis of Android vault applications", Computers & Security, Vol 70, pp. 516-531, 2017. http://dx.doi.org/10.1016/j.cose.2017.07.011

[6] Jiyong Park, Hyunmoo Lee, Giseop Noh, "Analysis of the impact of the US presidential candidate coverage of the US media on the Korean media: focusing on text mining", Journal of Korean Institute of Intelligent Systems Vol 31, No. 6, pp.510-518, 2021. http://doi.org/10.5391/JKIIS.2021.31.6.510

[7] Young-Man Kim and Jin Gu Lee, "Analysis of the Study Trend of Glass Ceiling by Period Using Text Mining", Journal of the Korea Contents Association, Vol 21, No. 8, pp.376-387, 2021. https://doi.org/10.5392/JKCA.2021.21.08.376

[8] Do-Hee Kim and Min-Jeong Kim, "Analyzing the Trend of False·Exaggerated Advertisement Keywords Using Text-mining Methodology (1990-2019)", Journal of the Korea Contents Association, Vol 21, No. 4, pp.38-49, 2021. https://doi.org/10.5392/JKCA.2021.21.04.038

[9] Jufri and Musdalifa Thamrin, "Political Influence Analysis Social Media Text Mining for Public Opinion: Case Study Makassar City", 2021 3rd International Conference on Cybernetics and Intelligent System (ICORIS), IEEE Xplore, 2021. https://doi.org/10.1109/ICORIS52787.2021.9649592

[10] Junseok Cheon, Yeoneo Kim, Xiao Liu, Gyun Woo, "Clear: An Efficient Auto App Collector and Lightweight Classification Tool for Android", 2018 IEEE International Conference on Applied System Invention (ICASI), IEEE Xplore, 2021. https://doi.org/10.1109/ICASI.2018.8394390

[11] Konstantinos F. Xylogiannopoulos, Panagiotis Karampelas, Reda Alhajj, "Text Mining for Malware Classification Using Multivariate All Repeated Patterns Detection", 2019 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM), IEEE Xplore, 2019. http://doi.org/10.1145/3341161.3350841

[12] Daegyu Kim and Changsoo Kim, "A Study on the APK Classification of Hidden Anti-forensics", 2019 Fall Conference on Korea Society For Internet Information, Vol 20, No. 2, pp.63-64, 2019. https://www.cseric.or.kr/literature/ser_view.php?SnxGubun=INME&mode=total&searchCate=literature&literature=Y&more=Y&research=Y&gu=INME020D9&cmd=qryview&SnxIndxNum=216144&rownum=1&totalCnt=1&q1_t=6rmA64yA6rec&listUrl=L2xpdGVyYXR1cmUvcmVzdWx0LnBocD9TbnhHdWJ1bj1JTk1FJm1vZGU9dG90YWwmc2VhcmNoQ2F0ZT1saXRlcmF0dXJlJmxpdGVyYXR1cmU9WSZxMT0lQjElRTglQjQlRUIlQjElRDQmbW9yZT1ZJmYxPU1OJnJlc2VhcmNoPVk=&f1=MN&q1=%B1%E8%B4%EB%B1%D4

[13] "Android (operating system)", [Online]. Available: https://en.wikipedia.org/wiki/Android_(operating_system)

[14] "Android application package", [Online]. Available: https://en.wikipedia.org/wiki/Android_application_package

[15] Android for Developer, Reduce your app size, [Online]. Available: https://developer.android.google.cn/topic/performance/reduce-apk-size

[16] "Proguard", [Online]. Available: https://www.guardsquare.com/proguard

[17] “Android for Developer, Shrink, obfuscate, and optimize your app”, [Online]. Available: https://developer.android.com/studio/build/shrink-code#groovy

[18] “Android for Developer, App resources overview”, [Online]. Available: https://developer.android.google.cn/guide/topics/resources/providing-resources

[19] Kroeger, Paul R. “Analyzing grammar: An introduction”, Cambridge University Press, pp. 12-14, 2005. https://doi.org/10.1017/CBO9780511801679

[20] Shin Jung-ho, Han Young-seok, Park Young-chan, Choi Key-Sun, “An HMM Part-of-Speech Tagger for Korean Based on Wordphrase”, Annual Conference on Human and Language Technology, Human and Language Technology, pp.389-394, 1994. https://doi.org/10.1075/cilt.136.37shi

[21] “JADX”, [Online]. Available: https://github.com/skylot/jadx

[22] “Python 3.10.1 documentation, The ElementTree XML API”, [Online]. Available: https://docs.python.org/3/library/xml.etree.elementtree.html

[23] Min Song, “Text Mining”, Chungram, 2017. http://www.kyobobook.co.kr/product/detailViewKor.laf?mallGb=KOR&ejkGb=KOR&barcode=9788959725939

[24] “NLTK Documentation, nltk.tokenize package”, [Online]. Available: https://www.nltk.org/api/nltk.tokenize.html

[25] Taeil Yoon, Suan Lee, “Analyzing text with Python”, neulbom, 2019. http://www.kyobobook.co.kr/product/detailViewKor.laf?ejkGb=KOR&mallGb=KOR&barcode=9788965550792&orderClick=LAG&Kc=

◐ 저 자 소 개 ◑

김 대 규(Dae-gyu Kim)
2007년 부경대학교 컴퓨터멀티미디어공학과(공학사)
2014년 부경대학교 대학원 정보시스템협동과정(공학석사)
2016년~현재 부경대학교 대학원 정보시스템협동과정 박사과정
관심분야 : 디지털포렌식, 사이버범죄, 정보보호
E-mail : dgkim16@pukyong.ac.kr

김 창 수(Chang-soo Kim)
1991년 중앙대학교 컴퓨터공학과(공학박사)
2013년~2014년 미국 콜로라도대학 방문교수
2020년~현재 한국인터넷정보학회 이사
1992년~현재 부경대학교 IT융합응용공학과 교수
관심분야 : 재난안전관리, 빅데이터, 인공지능, 정보시스템 등
E-mail : cskim@pknu.ac.kr