불법 복제 애플리케이션 탐지 방법 및 장치

Keyword	어플리케이션, 앱, APP, 불법 복제 방지, 오덱스 파일, 해쉬 테이블, 모바일		
기술보유 기관	숭실대학교 산학협력단	기술판매형식	기술협력, 라이센스
연구 책임자	최재영	기술 완성단계(TRL)	4단계-연구실 규모 실험 단계

기/술/개/요

오덱스(odex) 파일과 해쉬 테이블(hash table)을 이용하여 불법 복제 애플리케이션(illegally copied application)을 탐지하는 방법 및 장치에 관한 것임

기존 기술의 문제점

1. 안드로이드(Android) 기반의 모바일 기기에서는 불법적으로 모바일 기기에 설치되어 있는 앱에 대한 접근이 쉽게 가능하며, 손쉽게 추출이 가능함
2. 불법으로 복제된 앱을 사용하면서, 사용자 정보가 유출되거나 악성 코드 삽입 등으로 인한 문제 발생 가능성이 증가하고, 이로 인한 2차 피해 가능성 또한 커짐
3. 안드로이드 앱의 불법 복제에 대한 많은 문제들이 발생하고 있지만, 현재까지 안드로이드 플랫폼에서 불법으로 복제된 앱의 설치를 제한하거나 사용하지 못하게 하는 효과적인 방안이 마련되지 않고 있음

기술 내용 및 차별성

기술 내용 차별성

안드로이드 플랫폼에서 불법으로 복제된 애플리케이션(application)의 설치와 사용을 효과적으로 제한할 수 있는 방안에 관한 기술

기술 내용

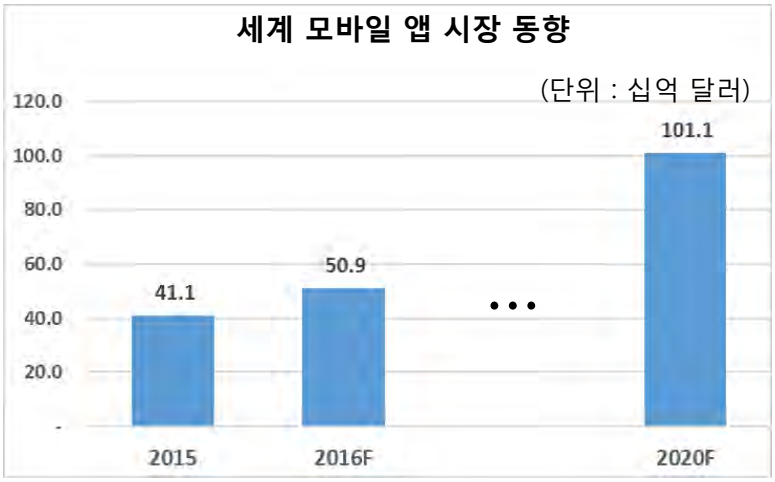
- 본 발명의 일 실시예에 따른 불법 복제 애플리케이션을 탐지하는 방법은 구매 기기의 식별자와 애플리케이션을 설치하는 사용 기기의 식별자를 비교하여 일치 여부에 따라 불법 복제 여부를 판단하는 것을 특징으로 함

기술의 우수성 / 혁신성

- **효율적인 불법 복제 여부 검사 가능**
  - 앱 파일을 압축 해제하는 과정을 거치지 않으므로, 불법 복제 여부 검사의 효율성 향상 가능
  - 스케줄러가 해쉬 테이블을 관리함으로써, 정상 앱에 대한 중복 검사를 방지하고 검사 대상인 앱의 수를 감소하는 것이 가능
- **은닉성 및 항상 실행 보장**
  - 시스템 서버에서 운용되는 인비저블 매니저(invisible manager)에 의해 관리되어 은닉성을 제공할 수 있으며 항상 실행 보장이 가능함

시장 현황

시장 현황



기술 활용 분야

기술 활용 분야

- 안드로이드 플랫폼을 사용하는 스마트폰, 태블릿 PC 등

권리현황

권리현황

발명의 명칭	문헌번호	등록일자	상태
불법복제 애플리케이션 탐지 방법 및 장치	KR 10-1600178	2016. 02. 26	등록

문 의 처

기술문의	

(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2016년03월14일
(11) 등록번호 10-1600178
(24) 등록일자 2016년02월26일

(51) 국제특허분류(Int. Cl.)
G06F 21/56 (2013.01)
(21) 출원번호 10-2014-0059858
(22) 출원일자 2014년05월19일
심사청구일자 2014년05월19일
(65) 공개번호 10-2015-0133038
(43) 공개일자 2015년11월27일
(56) 선행기술조사문헌
KR1020130027158 A*
보안공학연구논문지 제 10권, 51P 내지 62P(2013.02)
KR1020110085894 A
*는 심사관에 의하여 인용된 문헌
기술이전 희망 : **기술양도**, **실시권허여**, **기술지도**

(73) 특허권자
숭실대학교산학협력단
서울특별시 동작구 상도로 369 (상도동)
(72) 발명자
최재영
서울특별시 동작구 상도로 369, 정보과학관 21307호 (상도동)
김은희
서울특별시 동작구 상도로 369, 정보과학관 21307호 (상도동)
조득연
서울시 동작구 상도로 369 , 정보과학관 21307 (상도동)
(74) 대리인
송인호, **민영준**, **최관락**

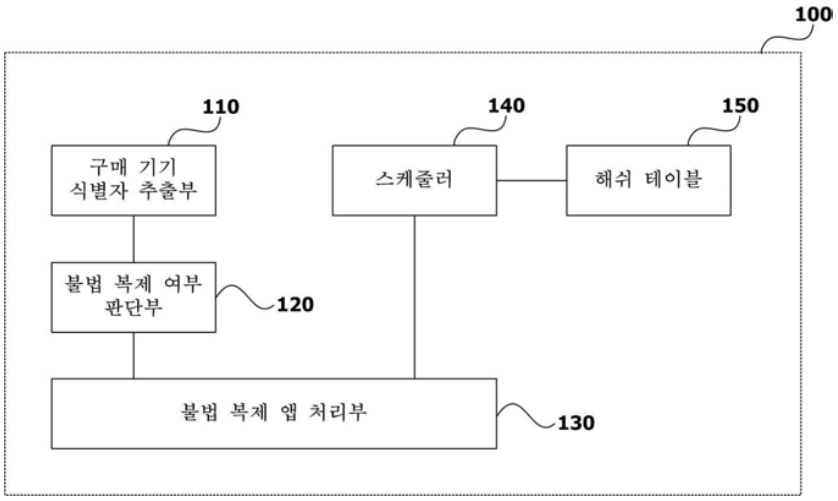
전체 청구항 수 : 총 8 항 심사관 : 구본재

(54) 발명의 명칭 **불법 복제 애플리케이션 탐지 방법 및 장치**

(57) 요 약

불법 복제 애플리케이션을 탐지하는 방법 및 장치가 제공된다. 본 발명의 불법 복제 애플리케이션을 탐지하는 장치는, 애플리케이션의 설치 과정에서 생성되는 오덱스(odex) 파일로부터 구매 기기의 식별자를 추출하는 구매 기기 식별자 추출부 및 상기 추출된 구매 기기의 식별자와 상기 애플리케이션을 설치하는 사용 기기의 식별자를 비교하여 일치 여부에 따라 불법 복제 여부를 판단하는 불법 복제 여부 판단부를 포함하되, 상기 구매 기기 식별자 추출부는 상기 오덱스(odex) 파일의 헤더(odex header)를 제거하고, 상기 헤더가 제거된 덱스(dex) 형태의 파일에서 자바(Java)의 클래스(class) 영역을 제외한 곳에 바이너리 형태로 삽입된 상기 구매 기기의 식별자를 추출하는 것을 특징으로 한다.

대 표 도 - 도1



이 발명을 지원한 국가연구개발사업
과제고유번호 201317940273
부처명 문화체육관광부
연구관리전문기관 저작권위원회
연구사업명 2013년도 저작권기술개발사업
연구과제명 시스템 소프트웨어 기반 모바일 앱 불법 복제 방지 기술 연구 개발
기 여 율 1/1
주관기관 숭실대학교 산학협력단
연구기간 2013.04.01 ~ 2014.03.31

명 세 서

청구범위

청구항 1

불법 복제 애플리케이션을 탐지하는 장치에 있어서,

애플리케이션의 설치 시, 오덱스(odex) 파일로부터 구매 기기의 식별자를 추출하는 구매 기기 식별자 추출부;

상기 애플리케이션의 설치 시, 상기 추출된 구매 기기의 식별자와 상기 애플리케이션을 설치하는 사용 기기의 식별자를 비교하여 일치 여부에 따라 불법 복제 여부를 판단하는 불법 복제 여부 판단부; 및

상기 애플리케이션이 설치된 후, 일정 시간마다 또는 상기 사용 기기의 부팅(booting) 시 상기 사용 기기에 설치 및 저장된 상기 애플리케이션에 대한 불법 복제 여부를 검사하는 스케줄러

를 포함하되,

상기 구매 기기 식별자 추출부는

상기 오덱스(odex) 파일의 헤더(odex header)를 제거하고, 상기 헤더가 제거된 덱스(dex) 형태의 파일에서 자바(Java)의 클래스(class) 영역을 제외한 곳에 바이너리 형태로 삽입된 상기 구매 기기의 식별자를 추출하며,

상기 스케줄러는

상기 애플리케이션의 고유한 해쉬 스트링(hash string)을 생성하고,

상기 생성된 해쉬 스트링과 해쉬 테이블(hash table)에 저장된 정상 애플리케이션의 해쉬 스트링을 비교하여 서로 일치하는 해쉬 스트링이 존재하면 상기 애플리케이션에 대한 불법 복제 여부 검사를 스킵(skip)하며,

상기 비교 결과 서로 일치하는 해쉬 스트링이 존재하지 않으면, 상기 애플리케이션을 압축 해제 하여 덱스(dex) 파일을 추출하고, 상기 추출된 덱스(dex) 파일로부터 상기 구매 기기의 식별자를 추출하여 상기 사용 기기의 식별자와 비교한 후 일치 여부에 따라 불법 복제 여부를 판단하는 것을 특징으로 하는 불법 복제 애플리케이션 탐지 장치.

청구항 2

제 1 항에 있어서,

상기 불법 복제 여부 판단부 또는 스케줄러의 상기 불법 복제 여부 판단 결과, 불법 복제된 것으로 판단되면, 상기 애플리케이션을 삭제하고 리포팅 정보를 화면에 표시하는 불법 복제 애플리케이션 처리부

를 더 포함하는 것을 특징으로 하는 불법 복제 애플리케이션 탐지 장치.

청구항 3

불법 복제 애플리케이션을 탐지하는 장치에 있어서,

불법 복제된 것이 아닌 정상 애플리케이션의 고유한 해쉬 스트링(hash string)을 저장하는 해쉬 테이블(hash table); 및

일정 시간마다 사용 기기에 설치 및 저장된 애플리케이션 파일에 대하여 고유의 해쉬 스트링을 생성하고, 상기 해쉬 테이블에 저장된 정상 애플리케이션의 해쉬 스트링을 참조하여 상기 애플리케이션 파일에 대한 불법 복제 여부를 검사하는 스케줄러

를 포함하되,

상기 스케줄러는,

상기 생성된 해쉬 스트링과 일치하는 해쉬 스트링이 상기 해쉬 테이블에 존재하면, 상기 애플리케이션 파일에 대한 불법 복제 여부 검사를 스킵(skip)하고,

상기 생성된 해쉬 스트링과 일치하는 해쉬 스트링이 상기 해쉬 테이블에 존재하지 않으면, 상기 애플리케이션 파일에 대한 불법 복제 여부를 검사하는 것을 특징으로 하는 불법 복제 애플리케이션 탐지 장치.

청구항 4

제 3 항에 있어서,

상기 스케줄러는,

상기 애플리케이션 파일에 대한 불법 복제 여부를 검사 시, 상기 애플리케이션 파일을 압축 해제 하여 덱스(dex) 파일을 추출하고, 상기 추출된 덱스(dex) 파일로부터 구매 기기의 식별자를 추출한 후, 상기 추출된 구매 기기의 식별자와 상기 사용 기기의 식별자를 비교하여 일치 여부에 따라서 불법 복제 여부를 판단하는 것을 특징으로 하는 불법 복제 애플리케이션 탐지 장치.

청구항 5

제 4 항에 있어서,

상기 스케줄러는,

상기 사용 기기의 부팅(booting) 시, 상기 애플리케이션 파일에 대한 불법 복제 여부를 검사하는 것을 특징으로 하는 불법 복제 애플리케이션 탐지 장치.

청구항 6

제 4 항에 있어서,

상기 스케줄러는,

상기 불법 복제 여부 판단 결과, 불법 복제된 것이 아닌 정상 애플리케이션으로 판단되면, 상기 애플리케이션 파일의 고유한 해쉬 스트링을 상기 해쉬 테이블에 저장하는 것을 특징으로 하는 불법 복제 애플리케이션 탐지 장치.

청구항 7

제 4 항에 있어서,

상기 불법 복제 여부 판단 결과, 불법 복제된 것으로 판단되면, 상기 애플리케이션 파일을 삭제하고 리포팅 정보를 화면에 표시하는 불법 복제 애플리케이션 처리부

를 더 포함하는 것을 특징으로 하는 불법 복제 애플리케이션 탐지 장치.

청구항 8

불법 복제 애플리케이션을 탐지하는 장치가 불법 복제 애플리케이션을 탐지하는 방법에 있어서,

(a) 애플리케이션의 설치 시, 오덱스(odex) 파일로부터 구매 기기의 식별자를 추출하는 단계;

(b) 상기 애플리케이션의 설치 시, 상기 추출된 구매 기기의 식별자와 상기 애플리케이션을 설치하는 사용 기기의 식별자를 비교하여 일치 여부에 따라 불법 복제 여부를 판단하는 단계; 및

(c) 상기 애플리케이션의 설치 후, 일정 시간마다 또는 상기 사용 기기의 부팅(booting) 시 상기 사용 기기에

설치 및 저장된 상기 애플리케이션에 대한 불법 복제 여부를 검사하는 단계

를 포함하되,

상기 (a) 단계는,

상기 오덱스(odex) 파일의 헤더(odex header)를 제거하고, 상기 헤더가 제거된 덱스(dex) 형태의 파일에서 자바(Java)의 클래스(class) 영역을 제외한 곳에 바이너리 형태로 삽입된 상기 구매 기기의 식별자를 추출하며,

상기 (c) 단계는,

상기 애플리케이션의 파일의 고유한 해쉬 스트링(hash string)을 생성하는 단계;

상기 생성된 해쉬 스트링과 해쉬 테이블(hash table)에 저장된 정상 애플리케이션의 해쉬 스트링을 비교하는 단계; 및

상기 비교 결과 서로 일치하는 해쉬 스트링이 존재하면, 상기 애플리케이션에 대한 불법 복제 여부 검사를 스킵(skip)하고, 상기 비교 결과 서로 일치하는 해쉬 스트링이 존재하지 않으면, 상기 애플리케이션을 압축 해제 하여 덱스(dex) 파일을 추출하고, 상기 추출된 덱스(dex) 파일로부터 상기 구매 기기의 식별자를 추출하여 상기 사용 기기의 식별자와 비교한 후 일치 여부에 따라 불법 복제 여부를 판단하는 단계

를 포함하는 것을 특징으로 하는 불법 복제 애플리케이션 탐지 방법.

청구항 9

삭제

발명의 설명

기 술 분 야

[0001] 본 발명은 불법 복제 애플리케이션(illegally copied application)을 탐지하는 방법 및 장치에 관한 것으로, 더욱 상세하게는 오덱스(odex) 파일과 해쉬 테이블(hash table)을 이용하여 불법 복제 애플리케이션을 탐지하는 방법 및 장치에 관한 것이다.

배 경 기 술

[0002] 음성과 문자 전송을 목적으로 하던 기존의 모바일 환경은 통신 환경이 발전하고 새로운 모바일 기기들이 등장하면서 스마트 모바일 환경으로 변화하고 있으며, 모바일 기기에서 실행되는 다양한 기능의 애플리케이션(application;이하 '앱'이라 칭함) 사용이 크게 증가하면서 불법으로 복제된 앱도 많이 사용되고 있다.

[0003] 특히 안드로이드(Android) 기반의 모바일 기기에서는 내부 저장소의 데이터를 보호하기 위하여 외부로부터의 불법적인 접근을 방지하고 있으나, 사용자가 루팅(rooting)과 같은 불법적인 접근 권한 획득 또는 ADB(Android Debug Bridge)와 같은 개발 도구를 이용하여 자신의 모바일 기기에 설치되어 있는 앱을 손쉽게 추출할 수 있다.

[0004] 이와 같이 안드로이드 기반에서는 앱에 대한 보호가 매우 취약하여, 안드로이드 앱 개발을 통한 수익성은 iOS의 24%밖에 되지 않을 정도로 매우 낮은 편이다.

[0005] 불법으로 복제된 앱의 사용 증가는, 앱에 대한 저작권을 보호하지 못하여 수익성이 떨어지고 앱 개발자의 참여를 저하시키며, 결국 안드로이드 앱 마켓의 신뢰성과 투명성을 떨어뜨린다.

[0006] 또한 불법으로 복제된 앱을 사용하면서, 사용자 정보가 유출되거나 악성 코드 삽입 등으로 인한 문제도 발생하는 등, 사용자들에게 2차적인 피해를 줄 수 있다.

[0007] 이처럼 안드로이드 앱의 불법 복제에 대한 많은 문제들이 발생하고 있지만, 현재까지 안드로이드 플랫폼에서 불법으로 복제된 앱의 설치를 제한하거나 사용하지 못하게 하는 효과적인 방안이 마련되지 않고 있는 실정이다.

발명의 내용

해결하려는 과제

[0008] 본 발명은 전술한 종래 기술의 문제점을 해결하기 위한 것으로, 안드로이드 플랫폼에서 불법으로 복제된 애플리케이션(application)의 설치와 사용을 효과적으로 제한할 수 있는 방안을 제공하고자 한다.

과제의 해결 수단

[0009] 상기와 같은 목적을 달성하기 위해, 본 발명의 일 실시예에 따른 불법 복제 애플리케이션을 탐지하는 장치는, 애플리케이션의 설치 과정에서 생성되는 오덱스(odex) 파일로부터 구매 기기의 식별자를 추출하는 구매 기기 식별자 추출부 및 상기 추출된 구매 기기의 식별자와 상기 애플리케이션을 설치하는 사용 기기의 식별자를 비교하여 일치 여부에 따라 불법 복제 여부를 판단하는 불법 복제 여부 판단부를 포함하되, 상기 구매 기기 식별자 추출부는 상기 오덱스(odex) 파일의 헤더(odex header)를 제거하고, 상기 헤더가 제거된 덱스(dex) 형태의 파일에서 자바(Java)의 클래스(class) 영역을 제외한 곳에 바이너리 형태로 삽입된 상기 구매 기기의 식별자를 추출하는 것을 특징으로 한다.

[0010] 본 발명의 일 측면에서, 상기 불법 복제 애플리케이션 탐지 장치는, 상기 불법 복제 여부의 판단 결과, 불법 복제된 것으로 판단되면, 상기 애플리케이션을 삭제하고 리포팅 정보를 화면에 표시하는 불법 복제 애플리케이션 처리부를 더 포함하는 것을 특징으로 한다.

[0011] 상기와 같은 목적을 달성하기 위해, 본 발명의 다른 실시예에 따른 불법 복제 애플리케이션을 탐지하는 장치는, 불법 복제된 것이 아닌 정상 애플리케이션의 고유한 해쉬 스트링(hash string)을 저장하는 해쉬 테이블(hash table) 및 일정 시간마다 사용 기기에 설치 및 저장된 애플리케이션 파일에 대하여 고유의 해쉬 스트링을 생성하고, 상기 해쉬 테이블에 저장된 정상 애플리케이션의 해쉬 스트링을 참조하여 상기 애플리케이션 파일에 대한 불법 복제 여부를 검사하는 스케줄러를 포함하되, 상기 스케줄러는 상기 생성된 해쉬 스트링과 일치하는 해쉬 스트링이 상기 해쉬 테이블에 존재하면, 상기 애플리케이션 파일에 대한 불법 복제 여부 검사를 스킵(skip)하고, 상기 생성된 해쉬 스트링과 일치하는 해쉬 스트링이 상기 해쉬 테이블에 존재하지 않으면, 상기 애플리케이션 파일에 대한 불법 복제 여부를 검사하는 것을 특징으로 한다.

[0012] 본 발명의 다른 측면에서, 상기 스케줄러는 상기 애플리케이션 파일에 대한 불법 복제 여부를 검사 시, 상기 애플리케이션 파일을 압축 해제 하여 덱스(dex) 파일을 추출하고, 상기 추출된 덱스(dex) 파일로부터 구매 기기의 식별자를 추출한 후, 상기 추출된 구매 기기의 식별자와 상기 사용 기기의 식별자를 비교하여 일치 여부에 따라서 불법 복제 여부를 판단하는 것을 특징으로 한다.

[0013] 또한, 본 발명의 다른 측면에서, 상기 스케줄러는 상기 사용 기기의 부팅(booting) 시, 상기 애플리케이션 파일에 대한 불법 복제 여부를 검사하는 것을 특징으로 한다.

[0014] 또한, 본 발명의 다른 측면에서, 상기 스케줄러는 상기 불법 복제 여부 판단 결과, 불법 복제된 것이 아닌 정상 애플리케이션으로 판단되면, 상기 애플리케이션 파일의 고유한 해쉬 스트링을 상기 해쉬 테이블에 저장하는 것을 특징으로 한다.

[0015] 또한, 본 발명의 다른 측면에서, 상기 불법 복제 애플리케이션 탐지 장치는 상기 불법 복제 여부 판단 결과, 불법 복제된 것으로 판단되면, 상기 애플리케이션 파일을 삭제하고 리포팅 정보를 화면에 표시하는 불법 복제 애플리케이션 처리부를 더 포함하는 것을 특징으로 한다.

[0016] 상기와 같은 목적을 달성하기 위해, 본 발명의 일 실시예에 따른 불법 복제 애플리케이션을 탐지하는 장치가 불법 복제 애플리케이션을 탐지하는 방법은 (a) 애플리케이션의 설치 과정에서 생성되는 오덱스(odex) 파일로부터 구매 기기의 식별자를 추출하는 단계 및 (b) 상기 추출된 구매 기기의 식별자와 상기 애플리케이션을 설치하는 사용 기기의 식별자를 비교하여 일치 여부에 따라 불법 복제 여부를 판단하는 단계를 포함하되, 상기 (a) 단계는 상기 오덱스(odex) 파일의 헤더(odex header)를 제거하고, 상기 헤더가 제거된 덱스(dex) 형태의 파일에서 자바(Java)의 클래스(class) 영역을 제외한 곳에 바이너리 형태로 삽입된 상기 구매 기기의 식별자를 추출하는 것을 특징으로 한다.

[0017] 본 발명의 일 측면에서, 상기 불법 복제 애플리케이션 탐지 방법은 (c) 일정 시간마다 또는 상기 사용 기기의 부팅(booting) 시 상기 사용 기기에 설치 및 저장된 애플리케이션 파일에 대한 불법 복제 여부를 검사하는 단계를 더 포함하되, 상기 일정 시간마다 불법 복제 여부를 검사하는 단계는, 상기 애플리케이션 파일의 고유한 해

쉬 스트링(hash string)을 생성하는 단계, 상기 생성된 해쉬 스트링과 해쉬 테이블(hash table)에 저장된 정상 애플리케이션 파일의 해쉬 스트링을 비교하는 단계 및 상기 비교 결과 서로 일치하는 해쉬 스트링이 존재하면, 상기 애플리케이션 파일에 대한 불법 복제 여부 검사를 스킵(skip)하고, 상기 비교 결과 서로 일치하는 해쉬 스트링이 존재하지 않으면, 상기 애플리케이션 파일을 압축 해제 하여 덱스(dex) 파일을 추출하고, 상기 추출된 덱스(dex) 파일로부터 상기 구매 기기의 식별자를 추출하여 상기 사용 기기의 식별자와 비교한 후 일치 여부에 따라 불법 복제 여부를 판단하는 단계를 포함하는 것을 특징으로 한다.

발명의 효과

[0018] 본 발명의 일 실시예에 따르면, 안드로이드 플랫폼에서 불법으로 복제된 애플리케이션(application)의 설치와 사용을 효과적으로 제한할 수 있다.

[0019] 본 발명의 효과는 상기한 효과로 한정되는 것은 아니며, 본 발명의 상세한 설명 또는 특허청구범위에 기재된 발명의 구성으로부터 추론 가능한 모든 효과를 포함하는 것으로 이해되어야 한다.

도면의 간단한 설명

[0020] 도 1은 본 발명의 일 실시예에 따른 불법 복제 애플리케이션을 탐지하는 장치의 구성을 도시한 블록도이다.

도 2는 본 발명의 일 실시예에 따른 불법 복제 애플리케이션을 탐지하는 과정을 도시한 흐름도이다.

도 3은 본 발명의 다른 실시예에 따른 불법 복제 애플리케이션을 탐지하는 과정을 도시한 흐름도이다.

도 4는 본 발명의 일 실시예에 따른 불법 복제 애플리케이션을 탐지하는 성능 평가 결과를 도시한 그래프이다.

도 5는 본 발명의 일 실시예에 따른 불법 복제 애플리케이션의 탐지 결과를 도시한 화면이다.

발명을 실시하기 위한 구체적인 내용

[0021] 이하에서는 첨부한 도면을 참조하여 본 발명을 설명하기로 한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며, 따라서 여기에서 설명하는 실시예로 한정되는 것은 아니다.

[0022] 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0023] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 부재를 사이에 두고 "간접적으로 연결"되어 있는 경우도 포함한다.

[0024] 또한 어떤 부분이 어떤 구성 요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성 요소를 제외하는 것이 아니라 다른 구성 요소를 더 구비할 수 있다는 것을 의미한다.

[0025] 이하 첨부된 도면을 참고하여 본 발명의 실시예를 상세히 설명하기로 한다.

[0026] 도 1은 본 발명의 일 실시예에 따른 불법 복제 애플리케이션을 탐지하는 장치의 구성을 도시한 블록도이다.

[0027] 본 발명의 일 실시예에 따른 불법 복제 애플리케이션(application; 이하, '앱'이라 칭함)을 탐지하는 장치(이하, '불법 복제 앱 탐지 장치'라 칭함)(100)는 크게 두 가지 경우에 불법 복제 앱을 탐지할 수 있다.

[0028] 첫 번째는, 앱 설치 이벤트 발생 시, 설치 과정에서 생성되는 Optimized Dalvik Executable 파일, 즉 odex 파일을 이용하여 불법 복제 앱을 탐지하는 경우이고, 두 번째는 이미 기기에 설치되어 저장된 앱에 대하여 스케줄러에 의해 일정 시간마다 불법 복제 앱의 여부를 확인하는 경우이다.

[0029] 이를 위해 불법 복제 앱 탐지 장치(100)는 구매 기기 식별자 추출부(110), 불법 복제 여부 판단부(120), 불법 복제 앱 처리부(130), 스케줄러(140) 및 해쉬 테이블(150)를 포함할 수 있다.

[0030] 각 구성 요소를 설명하면, 구매 기기 식별자 추출부(110)는 앱 설치 과정에서 생성되는 odex 파일로부터 odex header를 제거하여, 구매 시점에서 해당 앱에 포렌식 마크(Forensic watermark)로 삽입된 구매 기기의 식별자, 예를 들어, IMSI(International Mobile Subscriber Identity)를 추출할 수 있다.

[0031] 참고로, 구매 시점에 앱에 삽입되는 구매 기기의 식별자는 다른 식별자(예를 들어, 구매 기기에 종속되지 않는 구매자의 식별자 등)로 대체될 수도 있고, 반드시 포렌식 마크로 삽입되어야 하는 것은 아니며 상기 식별자를 보호할 수 있는 다양한 기법이 사용될 수 있다.

[0032] 구매 기기 식별자 추출부(110)가 odex 파일로부터 구매 기기의 식별자를 추출하는 상기 동작을 뒷받침하기 위해, odex 파일의 생성 과정을 간략히 설명하도록 한다.

[0033] 불법 복제 앱 탐지 장치(100)는 패키지 매니저(package manager)(미도시)를 포함할 수도 있고, 패키지 매니저 서비스를 제공하는 외부 서버(미도시)와 연동될 수도 있다.

[0034] 여기서, 패키지 매니저(미도시)/패키지 매니저 서비스는 앱을 설치하는 과정에서, 앱이 실행될 때 생성되거나 필요한 정보들을 저장하는 디렉토리, 설치 파일인 apk 파일의 복사본, 그리고 앱 실행 코드 영역인 dex 파일의 검증과 최적화 단계를 거친 odex 파일을 생성할 수 있다.

[0035] 앱의 설치 과정에서 생성되는 파일과 디렉토리를 설명하면, 패키지 매니저(미도시)는 앱 실행 중에 생성되거나 사용할 수 있는 데이터를 저장 및 관리하기 위해, '/data/data/ 디렉토리' 하위에 패키지 이름의 디렉토리를 생성할 수 있다.

[0036] 또한, 패키지 매니저(미도시)는 앱 실행에 필요한 리소스 정보가 들어있는 apk 파일을 복사하여 '/data/app/ 디렉토리'에 저장할 수 있다.

[0037] 만일, 앱 개발자가 앱의 강제 추출 방지 옵션을 설정한 경우, 패키지 매니저(미도시)는 apk 파일을 '/data/app-private/ 디렉토리'에 저장하고 '/data/app/ 디렉토리'에는 apk 파일의 심볼릭 링크를 저장할 수 있다.

[0038] 그리고 앱의 실행코드 영역인 dex 파일을 검증 및 최적화한 odex 파일을 생성하여 '/data/dalvik-cache/ 디렉토리'에 파일로 저장할 수 있다.

[0039] dex 파일에 대한 검증과 최적화 단계를 거친 odex 파일의 구조는, dex 파일과 비교할 때, 변형된 dex 영역에 40 byte의 odex header가 추가된 구조이며, 불법 복제 여부를 판단하기 위해 dex 내부에 삽입된 포렌식 마크는 dex 영역 중 실행을 관여하는 자바의 class 영역을 제외한 곳에 바이너리 형태로 삽입되어, dex 파일이 변형된 형태인 odex 파일에서도 동일한 위치에 삽입되어 있다.

[0040] 따라서odex 파일의 header 영역을 제거한 dex 파일로도 포렌식 워터마크 라이브러리를 이용해 구매 기기의 고유 식별자 추출이 가능하다.

[0041] 이러한 원리를 이용하여, 구매 기기 식별자 추출부(110)는 앱 설치 과정에서 생성되는 odex 파일에서 40 byte의 odex header 영역을 제거하여 dex 파일 형태로 만든 뒤, 구매 시점에서 해당 앱에 포렌식 마크로 삽입된 구매 기기의 고유 번호를 추출할 수 있다.

[0042] 한편, 불법 복제 여부 판단부(120)는 구매 기기 식별자 추출부(110)에 의해 추출된 구매 기기의 식별자와, 앱을 사용하는(설치하는) 단말기(이하, '사용 기기'라 칭함)의 식별자를 비교하고, 일치 여부에 따라 해당 앱의 불법 복제 여부를 판단할 수 있다.

[0043] 만일, 불법 복제된 것으로 판단되면, 불법 복제 여부 판단부(120)는 해당 앱 파일이 사용 기기에서 삭제되고 이와 관련된 내용이 사용자에게 보고될 수 있도록 불법 복제 앱 처리부(130)로 관련 정보를 제공할 수 있다.

[0044] 한편, 불법 복제 앱 처리부(130)는 불법 복제된 앱 파일을 사용 기기에서 삭제하고, 이에 대한 리포팅 정보를 화면에 표시할 수 있다.

[0045] 참고로, 불법 복제 앱이 삭제되어 화면에 표시되는 리포팅 정보는 불법 복제 앱의 이름, 탐지 시간 등의 정보를 포함할 수 있다.

[0046] 한편, 스케줄러(140)는 사용 기기가 부팅(booting)될 때 사용 기기에 설치 및 저장된 전체 앱 파일에 대하여 1차 불법 복제 여부를 검사할 수 있다.

[0047] 이를 위해 스케줄러(140)는 각 앱 파일(apk 파일)을 압축 해제하여 dex 파일을 추출하고, 추출된 dex 파일로부터 포렌식 마크 형태로 삽입된 구매 기기의 식별자를 추출할 수 있다.

[0048] 이후, 스케줄러(140)는 추출된 구매 기기의 식별자와 사용 기기의 식별자를 비교하여 해당 앱 파일의 불법 복제 여부를 판단할 수 있다.

[0049] 판단 결과, 불법 복제되지 않은 것으로 판단되면, 스케줄러(140)는 해당 앱 파일에 대하여 고유한 해쉬 스트링(hash string)을 생성하고, 생성된 해쉬 스트링과 해당 앱 파일의 정보를 해쉬 테이블(hash table)(150)에 저장할 수 있다

[0050] 만일, 불법 복제된 것으로 판단되면, 스케줄러(140)는 해당 앱 파일이 사용 기기에서 삭제되고 사용자에게 보고될 수 있도록 불법 복제 앱 처리부(130)로 관련 정보를 제공할 수 있다.

[0051] 참고로, 해쉬 테이블(150)은 메모리에 생성될 수 있으며, 사용 기기가 종료될 때까지 유지될 수 있다.

[0052] 따라서 사용 기기가 종료된 후 부팅되면 스케줄러(140)는 메모리에 해쉬 테이블(150)을 생성하고, 전술한 1차 불법 복제 여부 검사를 수행할 수 있다.

[0053] 또한 스케줄러(140)는 일정 시간마다 사용 기기에 설치 및 저장된 앱 파일에 대하여 2차 불법 복제 여부를 검사할 수 있다.

[0054] 이를 위해 스케줄러(140)는 사용 기기에 설치 및 저장된 모든 앱 파일에 대하여 고유의 해쉬 스트링을 추출하고, 해쉬 테이블(150)에 저장된 정상 앱 파일의 해쉬 스트링과 비교하여 동일한 해쉬 스트링이 존재하면, 해당 앱 파일에 대해서는 불법 복제 여부 검사를 수행하지 않을 수 있다.

[0055] 만일, 해쉬 테이블(150)에 동일한 해쉬 스트링이 존재하지 않으면, 스케줄러(140)는 해당 앱 파일을 압축 해제하여 dex 파일을 추출하고, 추출된 dex 파일로부터 포렌식 마크 형태로 삽입된 구매 기기의 식별자를 추출한 후, 추출된 구매 기기의 식별자와 사용 기기의 식별자를 비교하여 해당 앱 파일의 불법 복제 여부를 판단할 수 있다.

[0056] 판단 결과, 불법 복제되지 않은 것으로 판단되면, 스케줄러(140)는 해당 앱 파일의 고유한 해쉬 스티링을 해쉬 테이블(150)에 저장할 수 있으며, 불법 복제된 것으로 판단되면 해당 앱 파일이 사용 기기에서 삭제되고 사용자에게 보고될 수 있도록 불법 복제 앱 처리부(130)로 관련 정보를 제공할 수 있다.

[0057] 참고로, 스케줄러(140)가 사용 기기에 설치 및 저장된 앱 파일에 대한 불법 복제 여부 검사 시 해당 앱 파일을 압축 해제하여 dex 파일을 추출하고, 추출된 dex 파일로부터 포렌식 마크 형태로 삽입된 구매 기기의 식별자를 추출하는 것으로 설명하였지만, 전술한 odex 파일을 이용하여 구매 기기의 식별자를 추출하는 방식이 사용될 수도 있다.

[0058] 한편, 해쉬 테이블(150)은 스케줄러(140)에 의해 불법 복제가 아닌 정상으로 확인된 앱 파일의 고유한 해쉬 스트링과 앱 파일의 정보를 저장할 수 있으며, 일정 시간마다 갱신될 수 있다.

[0059] 해쉬 테이블(150)은 사용 기기의 메모리에 생성될 수 있으며, 사용 기기가 종료될 때까지 유지될 수 있다.

[0060] 참고로, 도 1에서는 앱을 구매 후 설치하는 시점에 앱 파일에 대한 불법 복제 여부를 검사하는 구매 기기 식별자 추출부(110) 및 불법 복제 여부 판단부(120)와, 사용 기기의 부팅 시 그리고 일정 시간마다, 앱 파일에 대한 불법 복제 여부를 검사하는 스케줄러(140) 및 해쉬 테이블(150)이 하나의 불법 복제 앱 탐지 장치(100)에 포함되는 것으로 설명하였지만, 실시예에 따라서 서로 독립된 불법 복제 앱 탐지 장치로 각각 존재할 수도 있다.

[0061] 이 경우, 불법 복제 앱 처리부(130)는 각각의 불법 복제 앱 탐지 장치에 포함될 수 있다.

[0062] 전술한 바와 같이, 본 발명의 불법 복제 앱 탐지 장치(100)는 앱을 구매 후 설치하는 시점, 사용 기기의 부팅 시 그리고 일정 시간마다, 앱 파일에 대한 불법 복제 여부를 검사할 수 있으며, 검사 후 불법 복제가 아닌 것으로 확인된 앱 파일에 대해서는 해쉬 테이블을 이용하여 이후 불법 복제 여부 검사 시 중복 검사를 하지 않도록 할 수 있다.

[0063] 특히, 앱을 구매 후 사용 기기에 설치하는 시점에서 앱 파일의 불법 복제 여부를 검사 시, 앱 파일(apk 파일)을 압축 해제하여 dex 파일을 추출한 후 구매 기기의 식별자를 추출하지 않고, 설치 시점에서 생성되는 odex 파일로부터 odex header 영역을 제거하여 dex 파일 형태로 만든 뒤, 구매 시점에서 포렌식 마크로 삽입된 구매 기기의 고유 번호를 추출할 수 있다.

[0064] 따라서, 본 발명의 불법 복제 앱 탐지 장치(100)는 앱 설치 시점에서 불법 복제 여부를 검사하기 위해 앱 파일을 압축 해제하는 과정을 거치지 않으므로, 불법 복제 여부 검사 시 소요됐던 시간을 단축시켜 불법 복제 여부 검사의 효율성을 향상시킬 수 있다.

[0065] 뿐만 아니라, 일정 시간 간격으로 수행되는 불법 복제 여부 검사 시, 스케줄러(140)가 해쉬 테이블(150)을 관리

함으로써, 정상 앱에 대한 중복 검사를 방지하고 검사 대상인 앱의 수를 감소시킬 수 있으므로, 불법 복제 여부 검사의 효율성을 또한 향상시킬 수 있다.

[0066] 참고로, 전술한 불법 복제 앱 탐지 장치(100)는 시스템 서버에서 운용되는 인비저블 매니저(invisible manager)에 의해 관리되어 은닉성을 제공할 수 있으며 항상 실행을 보장할 수 있다.

[0067] 또한, 앱 파일에 삽입된 구매 정보를 이용하여 앱의 불법 복제 여부를 판별하기 때문에 네트워크에 연결되지 않아도 불법 복제 파일의 탐지가 가능하며, 안드로이드 플랫폼에 독립적인 모듈 형태로 구성되어 다양한 안드로이드 버전에서 적용할 수 있다.

[0068] 도 2는 본 발명의 일 실시예에 따른 불법 복제 앱을 탐지하는 과정을 도시한 흐름도이다.

[0069] 도 2는 불법 복제 앱 탐지 장치(100)가 앱 마켓에서 구매된 앱을 설치 시, 해당 앱에 대한 불법 복제 여부를 검사하는 과정이다.

[0070] 앱 설치 이벤트가 발생되면, 불법 복제 앱 탐지 장치(100)는 앱 설치 과정에서 생성되는 odex 파일로부터 odex header를 제거한다(S201).

[0071] S201 후, 불법 복제 앱 탐지 장치(100)는 odex 파일로부터 odex header가 제거된 dex 파일 형태에서 포렌식 마크로 삽입된 구매 기기의 식별자를 추출한다(S202).

[0072] 여기서, odex 파일과 dex 파일의 구조를 비교하면, odex 파일은 변형된 dex 영역에 40 byte의 odex header가 추가된 구조이며, 불법 복제 여부를 판단하기 위해 dex 내부에 포렌식 마크로 삽입된 구매 기기의 식별자는 dex 영역 중 실행을 관여하는 자바의 class 영역을 제외한 곳에 바이너리 형태로 삽입되어, dex 파일이 변형된 형태인 odex 파일에서도 동일한 위치에 삽입되어 있다.

[0073] 따라서odex 파일의 header 영역을 제거한 dex 파일로도 포렌식 워터마크 라이브러리를 이용해 구매 기기의 고유 식별자 추출이 가능하다.

[0074] S202 후, 불법 복제 앱 탐지 장치(100)는 S202에서 추출된 구매 기기의 식별자와 앱을 설치하는 사용 기기의 식별자를 비교하여 불법 복제 여부를 확인한다(S203).

[0075] 만일, 서로 일치하지 않는 경우, 불법 복제 앱 탐지 장치(100)는 해당 앱을 불법 복제 앱으로 판단하고, 사용 기기에서 해당 앱을 삭제한 후 리포팅 정보를 화면에 표시한다(S204).

[0076] 여기서, 리포팅 정보는 불법 복제 앱으로 확인된 앱의 정보와 검사 시간 등의 정보를 포함할 수 있다.

[0077] 참고로, S203의 결과, 불법 복제가 아닌 정상적인 앱으로 확인된 경우, 해당 앱은, 일정 시간마다 사용 기기에 설치 및 저장된 앱 파일에 대하여 2차 불법 복제 여부를 검사하는 스케줄러(140)에 의해서 정상 앱으로 관리, 즉, 해당 앱의 고유한 해쉬 스트링이 해쉬 테이블에 저장되어 관리될 수 있다.

[0078] 도 3은 본 발명의 다른 실시예에 따른 불법 복제 앱을 탐지하는 과정을 도시한 흐름도이다.

[0079] 도 3은 불법 복제 앱 탐지 장치(100)가 해쉬 테이블을 이용하여 불법 복제 여부를 검사하는 과정이다.

[0080] 참고로, 불법 복제 앱 탐지 장치(100)가 해쉬 테이블을 이용하여 불법 복제 여부를 검사하는 시점은 사용 기기가 부팅될 때, 그리고 일정 시간마다 일 수 있다.

[0081] 도 3에서는 사용 기기가 부팅되어 사용 기기에 설치 및 저장된 모든 앱 파일에 대한 불법 복제 여부 검사가 1차적으로 수행된 상태이다.

[0082] 따라서, 메모리에는 해쉬 테이블이 생성되었으며, 해쉬 테이블에는 상기 1차 불법 복제 여부 검사를 통해 정상으로 확인된 앱 파일의 고유한 해쉬 스트링과 해당 앱 파일의 정보가 저장되어 있다.

[0083] 미리 정해진 시간(일정 시간)이 도래하면, 불법 복제 앱 탐지 장치(100)는 앱 파일들이 저장된 저장소를 참고하여 앱 파일 목록을 생성한다(S301).

[0084] 여기서 앱 파일 목록에는 사용 기기에 설치 및 저장된 모든 앱 파일이 포함될 수 있다.

[0085] S301 후, 불법 복제 앱 탐지 장치(100)는 앱 파일 목록의 모든 앱 파일에 대하여 고유한 해쉬 스트링을 생성한다(S302).

[0086] S302 후, 불법 복제 앱 탐지 장치(100)는 S302에서 생성된 각 앱 파일의 해쉬 스트링과 해쉬 테이블에 저장된

정상으로 확인된 앱 파일의 해쉬 스트링을 비교한다(S303).

[0087] S303 결과, 해쉬 스트링이 서로 일치하는 경우, 불법 복제 앱 탐지 장치(100)는 해당 앱 파일을 불법 복제 여부 검사에서 제외시켜 중복 검사를 방지한다(S304).

[0088] S303 결과, 서로 일치하는 해쉬 스트링이 존재하지 않으면, 불법 복제 앱 탐지 장치(100)는 해당 앱 파일에 대한 불법 복제 여부를 검사한다(S305).

[0089] 여기서, 불법 복제 앱 탐지 장치(100)는 해당 앱 파일(apk 파일)을 압축 해제하여 dex 파일을 추출하고, 추출된 dex 파일로부터 포렌식 마크 형태로 삽입된 구매 기기의 식별자를 추출한 후, 추출된 구매 기기의 식별자와 사용 기기의 식별자를 비교하여 해당 앱 파일의 불법 복제 여부를 판단할 수 있다.

[0090] S305 결과, 불법 복제가 아닌 정상 앱으로 판단되면, 불법 복제 앱 탐지 장치(100)는 해당 앱 파일의 고유한 해쉬 스트링과 앱 파일의 정보를 해쉬 테이블에 저장한다(S306).

[0091] 만일, S305 결과, 불법 복제된 앱으로 판단되면, 불법 복제 앱 탐지 장치(100)는 해당 앱 파일을 사용 기기에서 삭제하고, 해당 앱의 정보 및 불법 복제 앱이 탐지되어 삭제되었음을 알리는 메시지가 포함된 리포팅 정보를 화면에 표시한다(S307).

[0092] 도 4는 본 발명의 일 실시예에 따른 불법 복제 앱을 탐지하는 성능 평가 결과를 도시한 그래프이다.

[0093] 도 4는 설치 시점에서 실시간으로 불법 복제 앱을 탐지하고 리포팅하는 시간(이하, '탐지 시간'이라 칭함)을 측정한 결과로서, 총 71개의 앱을 대상으로 하였으며, 각각의 앱 크기는 최소 100KB부터 최대 50MB이고, 포렌식 워터마킹 기법을 사용하여 임의로 테스트 기기와 다른 식별자를 삽입하였다.

[0094] 도 4의 테스트에 사용한 테스트 기기는 안드로이드 플랫폼 개발을 위한 디버깅 보드 중 하나인 OdroidA(hardKernel)이며, 사용한 안드로이드 플랫폼 버전은 진저브레드(Gingerbread)이다.

[0095] 도 4의 A 그래프(410)는 앱 파일을 압축 해제하고 dex 파일을 추출한 후 포렌식 마크로 삽입된 사용 기기의 식별자를 추출하여 불법 복제 여부를 검사한 경우로서, 앱 파일의 크기에 따라 압축 해제에 소요되는 시간이 증가하기 때문에 평균 2062ms의 탐지 시간을 보이고 있음을 알 수 있다.

[0096] 반면, 도 4의 B 그래프(420)는 본 발명의 불법 복제 앱 탐지 장치(100)에 의한 결과로서, 앱의 설치 과정에서 생성되는 odex 파일을 이용하여 압축 해제 과정을 거치지 않기 때문에, 앱 파일의 크기에 상관 없이 일정한 탐지 시간을 보임을 알 수 있다.

[0097] 도 4의 B 그래프(420)에 나타난 평균 탐지 시간은 199ms로서, A 그래프(410)와 비교하여 약 90%의 탐지 시간이 감소되었음을 알 수 있다(즉 약 10배로 성능이 향상됨).

[0098] 도 5는 본 발명의 일 실시예에 따른 불법 복제 앱의 탐지 결과를 도시한 화면이다.

[0099] 도 5는 불법 복제 여부를 검사한 결과, 불법 복제된 앱으로 판단되어 사용 기기로부터 삭제된 후 해당 결과에 대한 알림을 사용자에게 제공하는 리포팅 화면(500)이다.

[0100] 리포팅 화면(500)은 도 5에 도시된 바와 같이, 탐지된 불법 복제 앱의 정보와 탐지 시간 등의 리포팅 정보(510)를 포함할 수 있다.

[0101] 전술한 본 발명의 설명은 예시를 위한 것이며, 본 발명이 속하는 기술분야의 통상의 지식을 가진 자는 본 발명의 기술적 사상이나 필수적인 특징을 변경하지 않고서 다른 구체적인 형태로 쉽게 변형이 가능하다는 것을 이해할 수 있을 것이다.

[0102] 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다.

[0103] 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0104] 본 발명의 범위는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 그 균등 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

부호의 설명

[0105] 100 : 불법 복제 앱을 탐지하는 장치

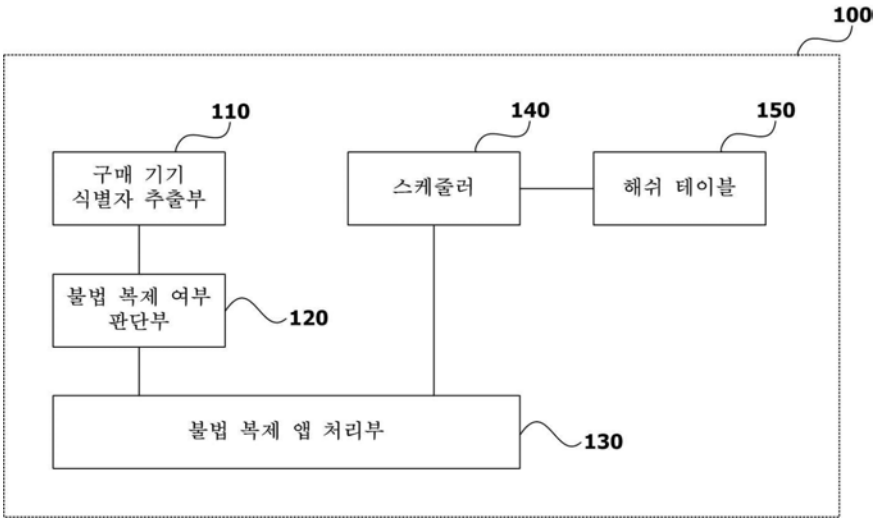
110 : 구매 기기 식별자 추출부, 120 : 불법 복제 여부 판단부

130 : 불법 복제 앱 처리부, 140 : 스케줄러

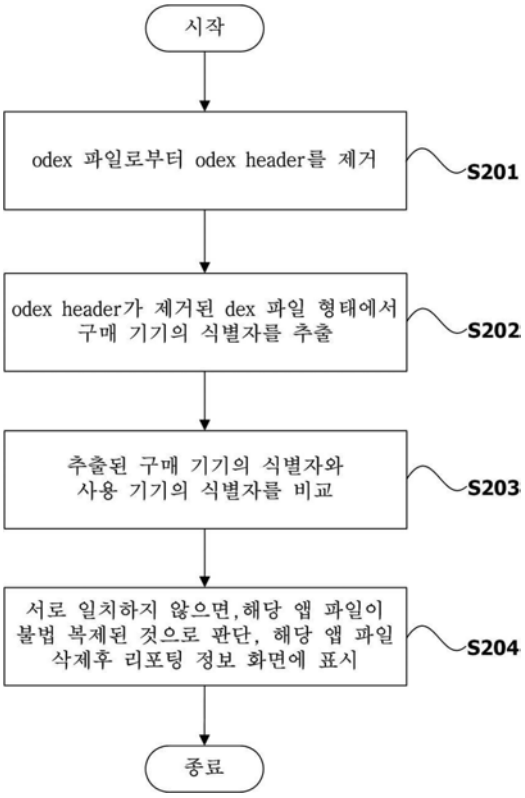
150 : 해쉬 테이블

도면

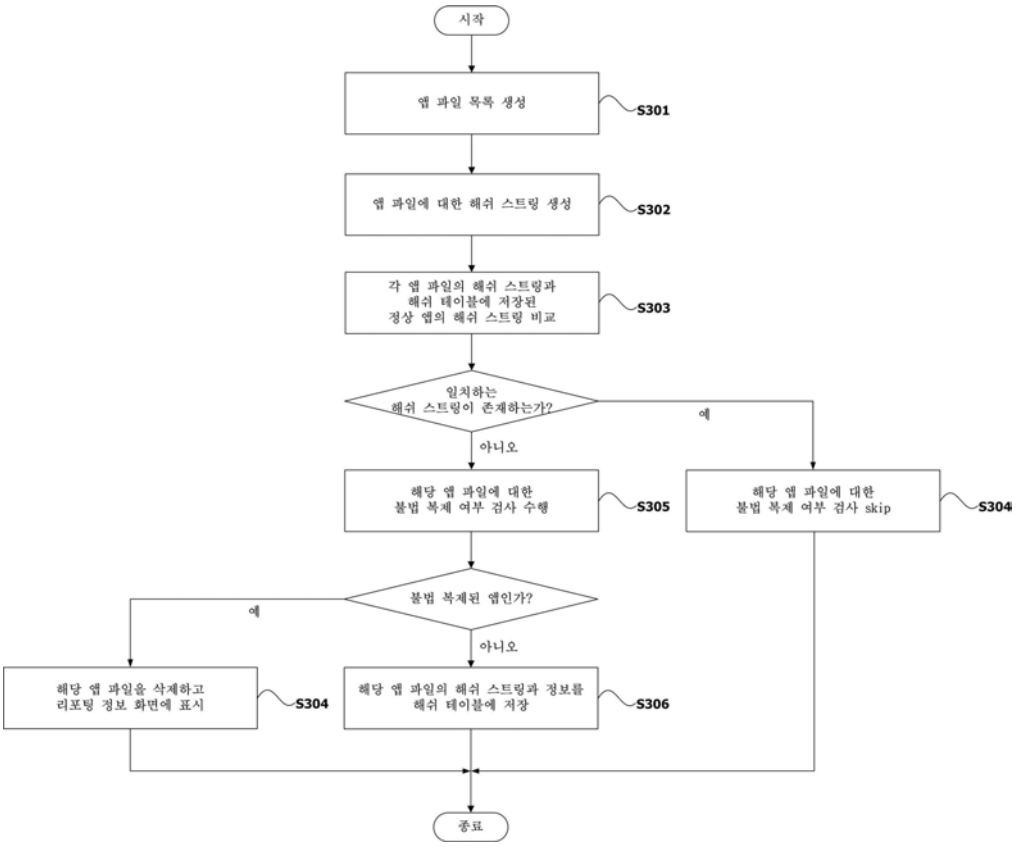
도면1



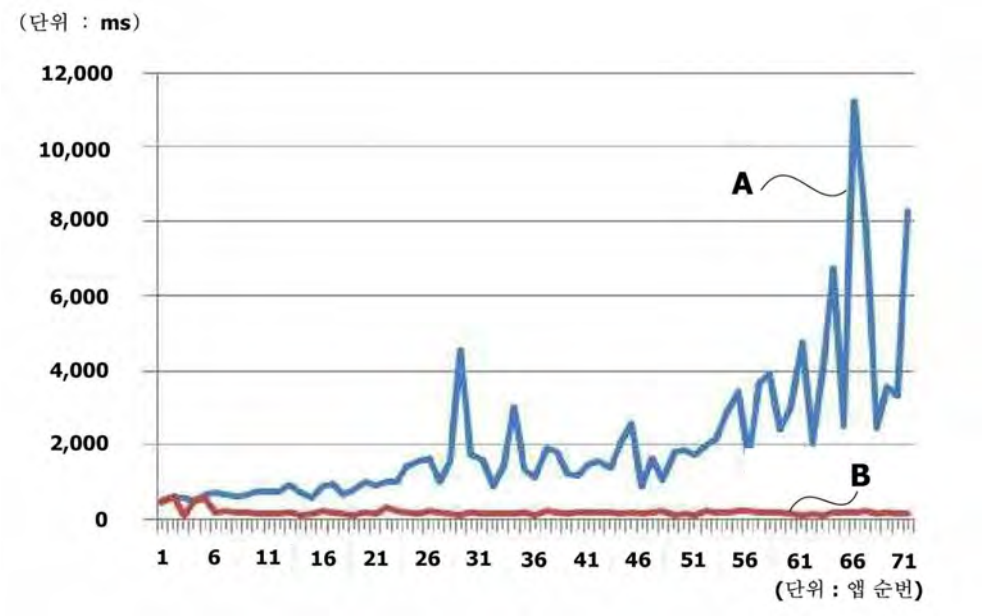
도면2


시작
odex 파일로부터 odex header를 제거
S201
odex header가 제거된 dex 파일 형태에서 구매 기기의 식별자를 추출
S202
추출된 구매 기기의 식별자와 사용 기기의 식별자를 비교
S203
서로 일치하지 않으면,해당 앱 파일이 불법 복제된 것으로 판단, 해당 앱 파일 삭제후 리포팅 정보 화면에 표시
S204
종료

도면3


시작
앱 파일 목록 생성
S301
앱 파일에 대한 해쉬 스트링 생성
S302
각 앱 파일의 해쉬 스트링과 해쉬 테이블에 저장된 정상 앱의 해쉬 스트링 비교
S303
일치하는 해쉬 스트링이 존재하는가?
예
아니오
해당 앱 파일에 대한 불법 복제 여부 검사 수행
S305
해당 앱 파일에 대한 불법 복제 여부 검사 skip
S304
불법 복제된 앱인가?
예
아니오
해당 앱 파일을 삭제하고 리포팅 정보 화면에 표시
S304
해당 앱 파일의 해쉬 스트링과 정보를 해쉬 테이블에 저장
S306
종료

도면4



도면5

【심사관 직권보정사항】

【직권보정 1】

【보정항목】 청구범위

【보정세부항목】 제8항

【변경전】

상기 애플리케이션 파일

【변경후】

상기 애플리케이션의 파일