(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2018년06월28일
(11) 등록번호 10-1872104
(24) 등록일자 2018년06월21일

(51) 국제특허분류(Int. Cl.)
*G06F 21/51* (2013.01) *G06F 21/62* (2013.01)
(52) CPC특허분류
*G06F 21/51* (2013.01)
*G06F 21/565* (2013.01)
(21) 출원번호 10-2016-0110592
(22) 출원일자 2016년08월30일
심사청구일자 2016년08월30일
(65) 공개번호 10-2018-0025380
(43) 공개일자 2018년03월09일
(56) 선행기술조사문헌
KR1020160006925 A*
KR1020150134254 A*
WO2014098387 A1
KR1020140081912 A
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
**한남대학교 산학협력단**
대전광역시 유성구 유성대로 1646 (전민동)
(72) 발명자
**이극**
대전광역시 서구 관저북로 52 109동 1806호 (관저동,대자연마을아파트)
**조지호**
충청남도 태안군 태안읍 군청10길 14 102동 704호 (남문리,진흥더블파크아파트)
**김정민**
대전광역시 대덕구 한남로12번길 33-12 202호 (오정동,아카데미)
(74) 대리인
**특허법인 아이퍼스**

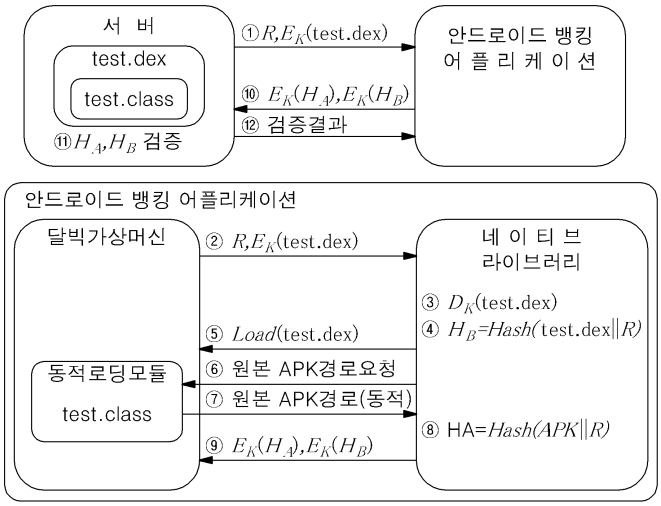
전체 청구항 수 : 총 3 항 심사관 : 구대성

(54) 발명의 명칭 **ＡＰＫ 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템 및 무결성 검증 방법**

## (57) 요 약

본 발명은 본 발명은 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템 및 무결성 검증방법에 대한 것이다. 보다 상세하게는 자바클래스를 DEX 파일로 변환하여 저장하며, 뱅킹 어플리케이션의 요청에 의해 상기 DEX 파일을 뱅킹 어플리케이션에 전송하는 서버; 및 상기 DEX 파일을 기반으로 제1해시값을 계산하고, APK 파일 경로를 동적으로 받아 무결성 검증용 제2해시값을 생성하는 뱅킹 어플리케이션을 포함하고, 상기 서버는 상기 제1해시값과 상기 제2해시값을 기반으로 무결성을 검증하는 것을 특징으로 하는 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템에 관한 것이다.

*대 표 도* - 도7



(52) CPC특허분류
*G06F 21/62* (2013.01)
*G06Q 20/322* (2013.01)
*G06Q 20/382* (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 B00009010
부처명 산업통상자원부
연구관리전문기관 한국산업기술진흥원
연구사업명 지역혁신센터사업
연구과제명 민군겸용보안공학연구센터
기 여 율 1/1
주관기관 한남대학교산학협력단
연구기간 2015.03.01 ~ 2016.02.29

# 명 세 서

## 청구범위

### 청구항 1

뱅킹 어플리케이션 무결성 검증 시스템에 있어서,

자바클래스를 DEX 파일로 변환하여 저장하며, 뱅킹 어플리케이션의 요청에 의해 상기 DEX 파일을 뱅킹 어플리케이션에 전송하는 서버; 및

상기 DEX 파일을 기반으로 제1해시값을 계산하고, APK 파일 경로를 동적으로 받아 무결성 검증용 제2해시값을 생성하는 뱅킹 어플리케이션을 포함하고,

상기 서버는 상기 제1해시값과 상기 제2해시값을 기반으로 무결성을 검증하며,

상기 서버는 상기 DEX 파일을 세션키 K로 암호화한 제1암호값을 난수 R과 함께 상기 뱅킹 어플리케이션으로 전송하고,

상기 뱅킹 어플리케이션은,

상기 제1암호값과 난수 R을 서버로부터 전송받는 달빅가상머신과, 상기 제1암호값과 난수 R을 달빅가상머신으로부터 전송받아 복호화하여 상기 제1해시값을 계산하고 상기 달빅가상머신을 로드하는 네이티브 라이브러리를 포함하며,

상기 달빅가상머신은, 상기 네이티브 라이브러리로부터의 요청에 의해, 원본 APK 경로를 동적으로 상기 네이티브 라이브러리로 전송하는 동적로드모듈을 포함하고,

상기 네이티브 라이브러리는 상기 제1해시값과 상기 제2해시값을 암호화한 제2암호값을 생성하고, 상기 달빅가상머신은 상기 제2암호값을 상기 서버로 전송하며,

상기 서버는 상기 제2암호값을 기반으로, 뱅킹 어플리케이션 및 DEX 파일의 무결성을 검증하고,

상기 서버는 검증결과 데이터를 상기 뱅킹 어플리케이션에 전송하는 것을 특징으로 하는 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템.

### 청구항 2

삭제

### 청구항 3

삭제

### 청구항 4

삭제

### 청구항 5

삭제

### 청구항 6

삭제

### 청구항 7

삭제

**청구항 8**

제 1항에 따른 무결성 검증 시스템을 이용한 뱅킹 어플리케이션 무결성 검증방법에 있어서,

자바클래스를 DEX파일로 변환하여 저장하는 서버가, 뱅킹어플리케이션의 요청에 의해 상기 DEX파일을 상기 뱅킹 어플리케이션에 전송하는 단계;

상기 뱅킹 어플리케이션이 상기 DEX 파일을 기반으로 제1해시값을 계산하고, APK 파일 경로를 동적으로 받아 무결성 검증용 제2해시값을 생성하는 단계;

상기 서버가 상기 제1해시값과 상기 제2해시값을 전송받아, 상기 제1해시값과 상기 제2해시값을 기반으로 무결성을 검증하는 단계; 및

상기 서버가 검증결과데이터를 상기 뱅킹 어플리케이션에 전송하는 단계를 포함하고,

상기 DEX파일을 상기 뱅킹 어플리케이션에 전송하는 단계에서,

상기 서버가 상기 DEX 파일을 세션키 K로 암호화한 제1암호값을 난수 R과 함께 상기 뱅킹 어플리케이션으로 전송하며,

상기 뱅킹 어플리케이션은 달빅가상머신과, 네이티브 라이브러리를 포함하고,

상기 달빅가상머신은, 상기 제1암호값과 난수 R을 서버로부터 전송받고, 상기 네이티브 라이브러리는, 상기 제1암호값과 난수 R을 달빅가상머신으로부터 전송받아 복호화하여 상기 제1해시값을 계산하고 상기 달빅가상머신을 로드하며,

상기 제2해시값을 생성하는 단계는,

상기 달빅가상머신에 포함된 동적로드모듈이, 상기 네이티브 라이브러리로부터의 요청에 의해, 원본 APK 경로를 동적으로 상기 네이티브 라이브러리로 전송하는 단계를 포함하고,

상기 제2해시값을 생성하는 단계에서,

상기 네이티브 라이브러리는 상기 제1해시값과 상기 제2해시값을 암호화한 제2암호값을 생성하고, 상기 달빅가상머신은 상기 제2암호값을 상기 서버로 전송하고,

상기 무결성을 검증하는 단계에서,

상기 서버는 상기 제2암호값을 기반으로, 뱅킹 어플리케이션 및 DEX 파일의 무결성을 검증하는 것을 특징으로 하는 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 방법.

**청구항 9**

삭제

**청구항 10**

삭제

**청구항 11**

삭제

**청구항 12**

컴퓨터에 의해 판독되며,

제 8항에 따른 검증방법을 실행시키는 기록매체.

## *발명의 설명*

### *기 술 분 야*

[0001] 본 발명은 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템 및 무결성 검증방법에 대한 것이다.

## 배 경 기 술

[0002] 통계청에 따르면 2014년 이동전화 가입자는 인구 100명당 113명, 이동전화 가입자 10명 중 7명은 스마트폰사용으로 보도되었다. 유선전화 가입자 수는 1,693만9천명으로 전년 1,762만 명 대비 3.9% 감소한 반면, 이동전화 가입자 수는 5,720만원8천명으로 전년 5,468만 명 대비 4.6% 증가했고, 이동전화 가입자 중 스마트폰 가입자 수는 4,056만 명으로 70.9%를 차지했다. 도 1은 연도별 유·무선 전화가입자 수 및 이동전화 가입률 그래프를 도시한 것이다.

[0003] 2015년 9월말 현재 인터넷 뱅킹서비스(모바일뱅킹 포함) 등록고객수는 1억 1,529만 명으로 전분기말 대비 1.9% 증가했고, 스마트폰 기반 모바일뱅킹의 등록고객수는 6,008만 명으로 전분기말대비 4.4% 증가하여 전체 증가세를 주도하고 있다. 스마트폰 뱅킹 등록고객 증가에 힘입어 전체 등록고객 중 모바일 뱅킹등록 고객이 차지하는 비중이 62.3%로 꾸준히 증가하고 있다.

[0004] 모바일 뱅킹 어플리케이션이 대중화되면서 금융사기에 대한 위험성도 커지고 있다. 구글플레이 어플리케이션을 사칭하는 이러한 악성 어플리케이션은 지금 이 순간에도 설치를 유도하는 스미싱 문자메시지를 발송하고 있을 것이다.

[0005] 구글 플레이 어플리케이션을 사칭한 악성 어플리케이션에 대한 AhnLab V3 Mobile의 진단명은 'Android-Trojan/Bankun'으로, 간략하게 '뱅쿤(Bankun)'이라는 명칭으로도 불린다. 이 어플리케이션은 금융정보를 탈취하기 위해 또 다른 어플리케이션을 추가로 설치하며, 문자메시지나 통화 기록 등 주요 개인정보를 유출하는 기능을 갖고 있다.

[0006] 특히 뱅쿤 어플리케이션은 설치될 때 동일한 아이콘뿐만 아니라 '구글 플레이 스토어(Google Paly Store)'라는 유사한 어플리케이션의 이름을 사용하고 있어 주의 깊게 살펴보지 않으면 예기치 않게 악성 어플리케이션을 설치하기 쉽다.

[0007] 안드로이드 기반 뱅킹 어플리케이션을 APK파일 동적로딩 기법을 이용한 무결성 검증에서 서버로부터 전송하는 동적 로딩 모듈인 DEX 파일을 수시로 변경한다면 동적 분석을 통한 공격도 효과적으로 방어할 수 있지만 APK파일 경로가 유출된다면 뱅킹 어플리케이션은 한 순간에 보안 취약점이 발견된다.

[0008] 안드로이드 특징으로써, 첫 번째로 안드로이드의 운영체제의 핵심이라도 할 수 있는 커널(Kernel)은 공개 운영체제인 리눅스(Linux)에 기반으로 상당히 오랜 기간 개발된 커널이므로 정교한 메모리 관리, 안정적인 멀티 스레드, 보안 등의 기능을 공짜로 사용할 수 있다는 점이고 모바일 환경에 어울리지 않은 크고 무거운 기능은 제거되어있으며 알람, 디버거 등의 기능은 추가되었다.

[0009] 두 번째로 공식적으로 자바 언어를 사용하며 고수준의 언어이므로 생산성이 높으며 하드웨어 추상계층을 제공하므로 전문 지식이 없어도 개발이 가능하다. 다만 고급언어에 해당하다 보니 성능이나 섬세함에서는 다소 불이익도 존재한다. 요즘 C로 개발할 수 있는 NDK가 개발되고 더 많은 부분에 네이티브(Native) 언어를 쓸 수 있도록 개선되었다.

[0010] 세 번째로 검증된 많은 라이브러리들을 대거 포함하고 있어 웬만한 기능은 별도의 외부 라이브러리를 사용할 필요가 없다. 자바가 언어 차원에서 제공하는 라이브러리들 외에도 OpenGL, SQLite, Freettype 등의 써드파티 라이브러리까지 내장되어 있다. 오픈 소스 중에 쓸 만한 것을 모두 집대성 해 놓은 셈이다.

[0011] 네 번째로 플랫폼에 내장된 빌트인 프로그램과 사용자가 만든 프로그램이 동일한 API를 사용하므로 모든 프로그램은 평등하다. 원한다면 기본 제공되는 프로그램을 사용자가 원하는 것으로 교체할 수 있고 플랫폼을 구성하는 요소들을 자유롭게 선택할 수 있다는 면에서 유연성이 뛰어나다. 마지막으로 개방된 환경인 만큼 개발 툴과 관련 문서들이 모두 무료로 제공된다. 심지어 운영체제의 핵심 소스도 대부분 공개되어 별도의 라이센스 비용이 들지 않는다. 도 2는 안드로이드 아키텍쳐를 도시한 것이다.

[0012] 안드로이드 달빅 가상머신은 안드로이드 런타임(Android Runtime)계층이다. 달빅 가상 머신은 적은 메모리 요구 사양에 모바일 환경에 최적화 되어 있다. 밑에 깔린 프로세스 아이솔레이션(process isolation), 메모리 관리, 스레딩 지원 등 운영 체제의 지원에 의존하나, 여러 개의 달빅 VM 인스턴스가 동시에 돌 수 있다. 달빅 가상 머신은 종종 자바 가상 머신으로 혼동하는 경우가 있으나, 달빅 가상 머신은 자바 바이트코드를 사용하지 않

기 때문에 자바 가상 머신과는 다르다. 대신, 안드로이드 SDK에 함께 들어 있는 dx라고 이름 붙은 도구를 이용하면 자바 클래스 파일들을 .dex 포맷으로 바꿀 수 있다.

[0013] 또한, 안드로이드 응용 프로그램 패키지(Android application package, APK)는 안드로이드의 소프트웨어와 미들웨어 배포에 사용되는 패키지 파일이며, '.apk'확장자를 가진다. APK 파일은 우분투 같은 데비안 기반 운영 체제에서 사용하는 뎁 패키지와 마이크로소프트 윈도에서 사용하는 MSI 패키지와 같은 설치 파일과 비슷하다.

[0014] APK 파일을 만들려면, 안드로이드용 프로그램을 먼저 컴파일한 후, 모든 파일들을 하나의 패키지 파일로 모은다. APK 파일은 해당하는 프로그램의 모든 코드를 포함하며, 자원, 정보, 인증서 및 매니페스트 파일(en:Manifest file) 등을 포함한다.

[0015] 안드로이드 응용 프로그램 패키지(APK)의 확장자는 .apk로 ZIP파일 기반인 JAR를 기반으로 하며, 압축 파일의 한 종류이다. MIME 유형은 application/vnd.android.package-archive이다.

[0016] 그리고, 안드로이드 NDK는 안드로이드 어플리케이션의 일부를 C/C++과 같은 네이티브 코드로 구현할 수 있도록 해주는 일종의 toolset이다. 안드로이드 NDK에는 컴파일러 및 링커가 포함된 툴체인, 헤더 파일, 라이브러리, 문서, 예제 프로그램 등 크로스 개발에 필요한 툴이 포함되어 있다.

[0017] 안드로이드 어플리케이션은 보통 JAVA로 개발하는데, 이전 임베디드 시스템에서는 자바를 사용하면 성능이 많이 떨어져서 잘 쓰지 않았다. 안드로이드 역시 JAVA로 구현된 부분이 순수 네이티브 코드로 만들어진 것보다는 느릴 수밖에 없으며, 기존에 사용하던 네이티브 코드를 JAVA로 이식하는 데 사용되는 비용이 너무 크기 때문에 개발자들은 C/C++로 안드로이드 어플리케이션을 개발할 필요가 있었다.

[0018] 이에 구글은 안드로이드 NDK를 배포하여 안드로이드 어플리케이션에서 특정 부분을 네이티브 코드로 구축할 수 있는 방법을 제공하기 시작했다. NDK를 사용하면 안드로이드 어플리케이션 레벨에서 네이티브 코드(CPU가 직접 이해할 수 있는 코드, 기계어)를 이용할 수 있으므로 CPU의 특성에 맞춰 변경하거나 하드웨어를 섬세하게 제어할 수 있다.

[0019] 또한 JAVA레벨에서 동작하는 것보다 더 빠른 실행을 가능하게 해주며 기존에 C/C++로 작성된 코드를 재사용할 수 있는 장점이 있지만 실행파일에 네이티브 코드가 포함되므로 실행파일이 하드웨어(CPU)에 의존하게 되는 가장 큰 단점이 있다. 더불어 코드의 복잡성이 증가하며 JAVA의 디버그 환경과 비교해보면 NDK로 작성한 코드를 디버깅하기가 더 어렵다.

[0020] 안드로이드 어플리케이션은 APK파일로 구성되어있으며 APK파일을 분석하기 위해 디컴 파일 방법 두 가지가 있는데, 첫 번째로는 JAVA코드로 디컴파일하는 방법과 두 번째로는 Smali코드로 디컴파일하는 방법이 있다. JAVA 코드로 분석할 경우에는 가독성은 높지만 일부코드를 정상적으로 디컴파일이 안되는 경우가 있고, Smali코드는 가독성은 떨어지지만 모든 코드를 볼 수 있다는 점이다.

[0021] JAVA코드로 분석하기 전에 dex2jar과 JD-GUD라는 프로그램이 필요하다. dex2jar프로그램은 dex파일을 jar파일로 변환시켜주고, 이 jar파일을 JD-GUI로 파일을 열어보면 JAVA코드를 확인할 수 있다. 도 3은 JD-GUI로 디컴파일한 자바클래스를 도시한 것이다.

[0022] Smali코드 분석은 apktool이라는 프로그램을 사용한다. apktool은 APK파일을 디컴파일하고 리패키징 할 수 있는 툴이다. 우선 자바 소스 파일과 동일한 패키지 구조를 가지고 클래스나 인터페이스 단위로 생성되며, 디컴파일을 하게 되면 Smali코드를 생성해주게 된다. 도 4는 디버그 로그를 삽입한 smali코드를 나타낸 것이다.

[0023] 도 4에 도시된 바와 같이 자바 소스코드보다는 가독성이 떨어지지만 기능 및 제어흐름을 충분히 이해할 수 있는 코드로 작성되어 있다. smali 코드는 텍스트 편집기에서 쉽게 수정할 수 있고, apktool에 의한 리패키징 과정에서 어셈블러에 의해 DEX 형식의 바이트코드로 자동 변환된다. 따라서 smali 코드에서 위변조하고자 하는 코드를 쉽게 찾을 수만 있다면 smali 코드 수정을 통한 어플리케이션 위변조는 매우 쉽다.

[0024] 이하에서는 악성 어플리케이션 피해사례에 대해 설명하도록 한다. A씨는 은행 스마트뱅킹으로 자금이체를 진행하던 중 추가인증이 필요하다며 QR코드가 나타나 메시지에 따라 어플리케이션을 설치했다. QR코드는 격자무늬의 2차원 코드로 스마트폰으로 QR코드를 스캔하면 각종 정보를 제공받을 수 있다. A씨는 어플리케이션 설치 후 보안카드를 비추는 순간 금융사기로 인식돼 동작을 멈췄다. 그러나 통신사에 확인해 보니 이미 게임머니 등으로 35만원이 결제 처리됐다.

[0025] 또한, 지난 3월 서울 송파구에 거주하는 B씨는 은행 스마트뱅킹을 이용하다 '보안프로그램 강화를 위해 앱을

설치하라' 는 메시지를 받았다. 사기범은 B씨에게 인증 등이 필요한 것처럼 속여 QR코드를 통해 악성 어플리케이션을 다운받도록 한 것으로 드러났다.

[0026] B씨는 어플리케이션을 설치한 뒤 보안카드 전. 후면 인식절차를 진행했고 사기범은 B씨가 눈치 못 챈 사이 170만원을 인출해 갔다. 할인 쿠폰을 제공한다며 어플리케이션을 설치하도록 하거나 소액 결제 취소를 미끼로 승인번호 입력을 유도한 뒤 수십만 원씩 결제하도록 하는 피해 사례도 적지 않다. 본인도 모르게 소액결제가 되는 ′ 큐싱′ 사기 피해가 나타난 만큼 스마트폰 대중화에 따른 각별한 주의가 필요함을 알 수 있다. 큐싱은 QR코드와 개인정보나 금융정보를 낚는다는 피싱의 합성어다.

[0027] 최근 악성 어플리케이션을 이용한 금융 사기가 기승하면서 피해금액이 점점 커지면서 뱅킹 어플리케이션을 이용하는 사용자들이 불안감은 점점 늘어나고 있다. 이 외에도 악성 어플리케이션은 사용자가 활동하지 않은 새벽시간 때 많은 피해가 발생 될 것으로 예상된다.

[0028] 따라서, 이러한 악성 어플리케이션의 무결성을 검증할 수 있는 방법 및 시스템이 요구되었다.

## 선행기술문헌

### 특허문헌

[0029] (특허문헌 0001) 등록특허 제1414084호

(특허문헌 0002) 공개특허 제2016-0006925호

(특허문헌 0003) 공개특허 제2014-0081912호

(특허문헌 0004) 등록특허 제1562282호

(특허문헌 0005) 등록특허 제1509585호

## 발명의 내용

### 해결하려는 과제

[0030] 따라서 본 발명은 상기와 같은 종래의 문제점을 해결하기 위하여 안출된 것으로서, 본 발명의 일실시예에 따르면, 무결성 검증의 취약점에 대한 대응방안으로 자바 코드의 동적 로딩을 이용한 무결성 검증 방안으로서 서버에서 전송하는 동적 로딩 모듈인 DEX파일을 수시로 변경하고 네이티브 라이브러리에서 원본 APK 경로를 요청할 때 달빅 가상 머신에서 APK 경로를 동적으로 변경하는 기법을 적용하여 뱅킹 어플리케이션의 무결성을 검증하는 시스템 및 검증방법을 제공하는데 그 목적이 있다.

[0031] 한편, 본 발명에서 이루고자 하는 기술적 과제들은 이상에서 언급한 기술적 과제들로 제한되지 않으며, 언급하지 않은 또 다른 기술적 과제들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

### 과제의 해결 수단

[0032] 본 발명의 목적은, 뱅킹 어플리케이션 무결성 검증 시스템에 있어서, 자바클래스를 DEX 파일로 변환하여 저장하며, 뱅킹 어플리케이션의 요청에 의해 상기 DEX 파일을 뱅킹 어플리케이션에 전송하는 서버; 및 상기 DEX 파일을 기반으로 제1해시값을 계산하고, APK 파일 경로를 동적으로 받아 무결성 검증용 제2해시값을 생성하는 뱅킹 어플리케이션을 포함하고, 상기 서버는 상기 제1해시값과 상기 제2해시값을 기반으로 무결성을 검증하는 것을 특징으로 하는 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템으로서 달성될 수 있다.

[0033] 또한, 상기 서버는 상기 DEX 파일을 세션키 K로 암호화한 제1암호값을 난수 R과 함께 상기 뱅킹 어플리케이션으로 전송하는 것을 특징으로 할 수 있다.

[0034] 그리고, 상기 뱅킹 어플리케이션은, 상기 제1암호값과 난수 R을 서버로부터 전송받는 달빅가상머신과, 상기 제1암호값과 난수 R을 달빅가상머신으로부터 전송받아 복호화하여 상기 제1해시값을 계산하고 상기 달빅가상머신을 로드하는 네이티브 라이브러리를 포함하는 것을 특징으로 할 수 있다.

[0035] 또한, 상기 달빅가상머신은, 상기 네이티브 라이브러리로부터의 요청에 의해, 원본 APK 경로를 동적으로 상기 네이티브 라이브러리로 전송하는 동적로드모듈을 포함하는 것을 특징으로 할 수 있다.

[0036] 그리고, 네이티브 라이브러리는 상기 제1해시값과 상기 제2해시값을 암호화한 제2암호값을 생성하고, 상기 달빅가상머신은 상기 제2암호값을 상기 서버로 전송하는 것을 특징으로 할 수 있다.

[0037] 또한, 상기 서버는 상기 제2암호값을 기반으로, 뱅킹 어플리케이션 및 DEX 파일의 무결성을 검증하는 것을 특징으로 할 수 있다.

[0038] 그리고, 상기 서버는 검증결과 데이터를 상기 뱅킹 어플리케이션에 전송하는 것을 특징으로 할 수 있다.

[0039] 또 다른 카테고리로서 본 발명의 목적은, 뱅킹 어플리케이션 무결성 검증방법에 있어서, 자바클래스를 DEX파일로 변환하여 저장하는 서버가, 뱅킹어플리케이션의 요청에 의해 상기 DEX파일을 상기 뱅킹 어플리케이션에 전송하는 단계; 상기 뱅킹 어플리케이션이 상기 DEX 파일을 기반으로 제1해시값을 계산하고, APK 파일 경로를 동적으로 받아 무결성 검증용 제2해시값을 생성하는 단계; 상기 서버가 상기 제1해시값과 상기 제2해시값을 전송받아, 상기 제1해시값과 상기 제2해시값을 기반으로 무결성을 검증하는 단계; 및 상기 서버가 검증결과데이터를 상기 뱅킹 어플리케이션에 전송하는 단계를 포함하는 것을 특징으로 하는 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 방법으로서 달성될 수 있다.

[0040] 또한, 상기 DEX파일을 상기 뱅킹 어플리케이션에 전송하는 단계에서, 상기 서버가 상기 DEX 파일을 세션키 K로 암호화한 제1암호값을 난수 R과 함께 상기 뱅킹 어플리케이션으로 전송하며, 상기 뱅킹 어플리케이션은 달빅가상머신과, 네이티브 라이브러리를 포함하고, 상기 달빅가상머신은, 상기 제1암호값과 난수 R을 서버로부터 전송받고, 상기 네이티브 라이브러리는, 상기 제1암호값과 난수 R을 달빅가상머신으로부터 전송받아 복호화하여 상기 제1해시값을 계산하고 상기 달빅가상머신을 로드하는 것을 특징으로 할 수 있다.

[0041] 그리고, 상기 제2해시값을 생성하는 단계는, 상기 달빅가상머신에 포함된 동적로드모듈이, 상기 네이티브 라이브러리로부터의 요청에 의해, 원본 APK 경로를 동적으로 상기 네이티브 라이브러리로 전송하는 단계를 더 포함하는 것을 특징으로 할 수 있다.

[0042] 또한, 상기 제2해시값을 생성하는 단계에서, 상기 네이티브 라이브러리는 상기 제1해시값과 상기 제2해시값을 암호화한 제2암호값을 생성하고, 상기 달빅가상머신은 상기 제2암호값을 상기 서버로 전송하고, 상기 무결성을 검증하는 단계에서, 상기 서버는 상기 제2암호값을 기반으로, 뱅킹 어플리케이션 및 DEX 파일의 무결성을 검증하는 것을 특징으로 할 수 있다.

## *발명의 효과*

[0043] 본 발명의 일실시예에 따르면, 무결성 검증의 취약점에 대한 대응방안으로 자바 코드의 동적 로딩을 이용한 무결성 검증 방안으로서 서버에서 전송하는 동적 로딩 모듈인 DEX파일을 수시로 변경하고 네이티브 라이브러리에서 원본 APK 경로를 요청할 때 달빅 가상 머신에서 APK 경로를 동적으로 변경하는 기법을 적용하여 뱅킹 어플리케이션의 무결성을 검증할 수 있는 효과를 갖는다.

[0044] 한편, 본 발명에서 얻을 수 있는 효과는 이상에서 언급한 효과들로 제한되지 않으며, 언급하지 않은 또 다른 효과들은 아래의 기재로부터 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에게 명확하게 이해될 수 있을 것이다.

## *도면의 간단한 설명*

[0045] 본 명세서에 첨부되는 다음의 도면들은 본 발명의 바람직한 일실시예를 예시하는 것이며, 발명의 상세한 설명과 함께 본 발명의 기술적 사상을 더욱 이해시키는 역할을 하는 것이므로, 본 발명은 그러한 도면에 기재된 사항에만 한정되어 해석 되어서는 아니 된다.

도 1은 연도별 유·무선 전화가입자수 및 이동전화 가입률 그래프

도 2는 안드로이드 아키텍처,

도 3은 JD-GUI로 디컴파일한 자바크래스,

도 4는 디버그 로그를 삽입한 smali코드,

도 5는 PC Drozer 실행화면,

도 6은 모바이 기기에서 drozer Agent 실행화면,

도 7은 본 발명의 일실시예에 따른 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템의 블록도,

도 8은 본 발명의 일실시예에 따른 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 방법의 흐름도를 도시한 것이다.

## *발명을 실시하기 위한 구체적인 내용*

[0046] 이상의 본 발명의 목적들, 다른 목적들, 특징들 및 이점들은 첨부된 도면과 관련된 이하의 바람직한 실시예들을 통해서 쉽게 이해될 것이다. 그러나 본 발명은 여기서 설명되는 실시예들에 한정되지 않고 다른 형태로 구체화될 수도 있다. 오히려, 여기서 소개되는 실시예들은 개시된 내용이 철저하고 완전해질 수 있도록 그리고 통상의 기술자에게 본 발명의 사상이 충분히 전달될 수 있도록 하기 위해 제공되는 것이다.

[0047] 본 명세서에서, 어떤 구성요소가 다른 구성요소 상에 있다고 언급되는 경우에 그것은 다른 구성요소 상에 직접 형성될 수 있거나 또는 그들 사이에 제 3의 구성요소가 개재될 수도 있다는 것을 의미한다. 또한 도면들에 있어서, 구성요소들의 두께는 기술적 내용의 효과적인 설명을 위해 과장된 것이다.

[0048] 본 명세서에서 기술하는 실시예들은 본 발명의 이상적인 예시도인 단면도 및/또는 평면도들을 참고하여 설명될 것이다. 도면들에 있어서, 막 및 영역들의 두께는 기술적 내용의 효과적인 설명을 위해 과장된 것이다. 따라서 제조 기술 및/또는 허용 오차 등에 의해 예시도의 형태가 변형될 수 있다. 따라서 본 발명의 실시예들은 도시된 특정 형태로 제한되는 것이 아니라 제조 공정에 따라 생성되는 형태의 변화도 포함하는 것이다. 예를 들면, 직각으로 도시된 영역은 라운드지거나 소정 곡률을 가지는 형태일 수 있다. 따라서 도면에서 예시된 영역들은 속성을 가지며, 도면에서 예시된 영역들의 모양은 소자의 영역의 특정 형태를 예시하기 위한 것이며 발명의 범주를 제한하기 위한 것이 아니다. 본 명세서의 다양한 실시예들에서 제1, 제2 등의 용어가 다양한 구성요소들을 기술하기 위해서 사용되었지만, 이들 구성요소들이 이 같은 용어들에 의해서 한정되어서는 안 된다. 이들 용어들은 단지 어느 구성요소를 다른 구성요소와 구별시키기 위해서 사용되었을 뿐이다. 여기에 설명되고 예시되는 실시예들은 그것의 상보적인 실시예들도 포함한다.

[0049] 본 명세서에서 사용된 용어는 실시예들을 설명하기 위한 것이며 본 발명을 제한하고자 하는 것은 아니다. 본 명세서에서, 단수형은 문구에서 특별히 언급하지 않는 한 복수형도 포함한다. 명세서에서 사용되는 '포함한다(comprises)' 및/또는 '포함하는(comprising)'은 언급된 구성요소는 하나 이상의 다른 구성요소의 존재 또는 추가를 배제하지 않는다.

[0050] 아래의 특정 실시예들을 기술하는데 있어서, 여러 가지의 특정적인 내용들은 발명을 더 구체적으로 설명하고 이해를 돕기 위해 작성되었다. 하지만 본 발명을 이해할 수 있을 정도로 이 분야의 지식을 갖고 있는 독자는 이러한 여러 가지의 특정적인 내용들이 없어도 사용될 수 있다는 것을 인지할 수 있다. 어떤 경우에는, 발명을 기술하는 데 있어서 흔히 알려졌으면서 발명과 크게 관련 없는 부분들은 본 발명을 설명하는데 있어 별 이유 없이 혼돈이 오는 것을 막기 위해 기술하지 않음을 미리 언급해 둔다.

[0052] 이하에서는 본 발명의 일실시예에 따른 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템 및 무결성 검증방법에 대해 설명하도록 한다. 이러한 본 발명의 일실시예에 따라, 무결성 검증의 취약점에 대한 대응방안으로 자바 코드의 동적 로딩을 이용한 무결성 검증 방안에서 서버에서 전송하는 동적 로딩 모듈인 DEX파일을 수시로 변경하고 네이티브 라이브러리에서 원본 APK 경로를 요청할 때 달빅 가상 머신에서 APK 경로를 동적으로 변경하는 기법을 적용하여 뱅킹 어플리케이션의 무결성을 검증할 수 있게 된다.

[0053] 먼저, 자바코드의 동적 로딩에 대해 설명하도록 한다. 동적로딩이란, 배포된 애플리케이션에는 실행코드를 포함시키지 않고 실행 시점에 코드를 동적으로 로딩하여 사용하는 방법을 말한다.

[0054] 즉, 실행코드의 정적분석을 불가능하게 하며 동적 분석 또한 어렵게 함으로써 코드를 보호하는 방법이다. 안드로이드 달빅 가상머신의 DexClassLoader 클래스는 DEX 파일을 현재 실행중인 프로세스에 동적으로 로딩할 수 있도록 지원한다.

[0055] Drozer는 안드로이드 어플리케이션 취약점 도구에 해당한다. PC와 안드로이드 단말기에 설치가 가능하며 PC에는 Drozer(Windows installer)와 단말기에는 Drozer(agent.apk Only)를 다운로드 받아 PC와 단말기에 각각 설치를

해야 한다.

[0056] 도 5는 PC Drozer 실행화면을 도시한 것이고, 도 6은 모바일 기기에서 drozer Agent 실행화면을 도시한 것이다. 두 기기에 설치가 끝나면 첫 번째로 PC에는 adb isntall agent.apk 명령어를 입력하여 adb를 설치하고, 도 6에 도시된 바와 같이 모바일 기기에 drozer Agent 실행 후 하단에 사용 안함을 누르면 사용하기로 변경이 된다. Embedded Server 설정을 상세내용 확인이 가능하다.

[0057] 그리고, 도 5에 도시된 바와 같이, PC는 모바일 기기를 PC에 연결하고 Portforwarding 후, drozer 콘솔로 연결하고, 연결한 후부터는 Drozer 콘솔 명령어를 입력하여 어플리케이션 정보와 확인이 가능하다.

[0059] 이하에서는 보다 구체적으로, 본 발명의 일실시예에 따른 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템 및 무결성 검증방법에 대해 설명하도록 한다. 먼저, 도 7은 본 발명의 일실시예에 따른 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템의 블록도를 도시한 것이다. 또한, 도 8은 본 발명의 일실시예에 따른 APK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증방법의 흐름도를 도시한 것이다.

[0060] 도 7에 도시된 바와 같이, 본 발명의 일실시예에 따른 PK 파일 동적로딩 기법을 이용한 뱅킹 어플리케이션 무결성 검증 시스템은 도 7에 도시된 바와 같이, 서버와 뱅킹 어플리케이션 간에 데이터가 전송되어지면서 검증이 실행되어지게 됨을 알 수 있다. 또한, 뱅킹 어플리케이션은 동적로딩모듈을 갖는 달빅가상머신과 네이티브 라이브러리를 포함하여 구성되게 된다.

[0061] 서버는 자바클래스를 test.dex 파일로 변환하여 저장하며, 뱅킹 어플리케이션의 요청에 의해(S1) test.dex 파일을 뱅킹 어플리케이션에 전송하게 된다. 보다 구체적으로, 서버는 test.dex 파일을 세션키 K로 암호화한 제1암호값($E_K$(test.dex))을 난수 R과 함께 뱅킹 어플리케이션의 달빅가상머신으로 전송하게 된다(S2).

[0062] 세션키 K는 송수신데이터의 암호화 및 복호화를 위해서 뱅킹 어플리케이션과 서버 사이에 대칭키이며, 기존의 뱅킹 어플리케이션들은 모두 송수신 데이터의 암호화를 제공했으므로 세션키 K가 존재하게 된다.

[0063] 그리고, 달빅가상머신은 도 7에 도시된 바와 같이, 수신받은 제1암호값($E_K$(test.dex))을 난수 R을 네이티브 라이브러리로 전송하게 되며, 네이티브 라이브러리는 전송받은 암호화된 test.dex파일인 제1암호값($E_K$(test.dex))을 세션키 K로 복호화($D_K$(test.dex))하고(S3), 이를 기반으로 제1해시값($H_B$)을 계산하게 된다(S4).

[0064] 그리고, 네이티브 라이브러리는 제1해시값($H_B$)을 계산하고, 달빅가상머신을 로드(Load)하고(S5), 달빅가상머신에 구비된 동적로드모듈에 원본 APK 경로를 동적으로 요청하게 된다(S6).

[0065] 그리고, 네이티브 라이브러리는, 동적로딩모듈로부터 APK 파일 경로를 동적으로 전송받아 무결성 검증용 제2해시값($H_A$)을 생성하게 된다(S7).

[0066] 다음으로, 네이티브 라이브러리는 계산한 제1해시값($H_B$)과 생성된 제2해시값($H_A$)을 암호화한 제2암호값($E_K$($H_A$), $E_K$($H_B$))을 달빅가상머신을 통해 서버로 전송하게 된다(S8).

[0067] 서버는 전송받은 제2암호값(EK($H_A$), EK($H_B$)) 기반으로 무결성을 검증하게 된다(S10). 즉, 서버는 해시값을 이용하여 뱅킹 어플리케이션 및 동적 로딩 모듈인 test.dex파일의 무결성 검증을 수행하게 된다.

[0068] 마지막으로 서버는 검증결과 데이터를 뱅킹 어플리케이션으로 전송하게 된다(S10).

[0070] 따라서 본 발명의 일실시예에서는 앞서 언급한 바와 같이, 무결성 검증의 취약점에 대한 대응방안으로 자바 코드의 동적 로딩을 이용한 무결성 검증 방안으로서 서버에서 전송하는 동적 로딩 모듈인 DEX파일을 수시로 변경하고 네이티브 라이브러리에서 원본 APK 경로를 요청할 때 달빅 가상 머신에서 APK 경로를 동적으로 변경하는 기법을 적용하여 뱅킹 어플리케이션의 무결성을 검증할 수 있는 효과를 갖는다.

[0072] 한편, 본 발명은 또한 컴퓨터로 읽을 수 있는 기록매체에 컴퓨터가 읽을 수 있는 코드로서 구현하는 것이 가능하다. 컴퓨터가 읽을 수 있는 기록매체는 컴퓨터 시스템에 의해 읽혀질 수 있는 데이터가 저장되는 모든 종류의 기록장치를 포함한다. 컴퓨터가 읽을 수 있는 기록매체의 예로는 ROM, RAM, CD-ROM, 자기 테이프, 플로피 디스크, 광데이터 저장장치 등이 있으며, 또한 케리어 웨이브(예를 들어 인터넷을 통한 전송)의 형태로 구현되는 것도 포함한다. 또한, 컴퓨터가 읽을 수 있는 기록매체는 네트워크로 연결된 컴퓨터 시스템에 분산되어, 분산방식

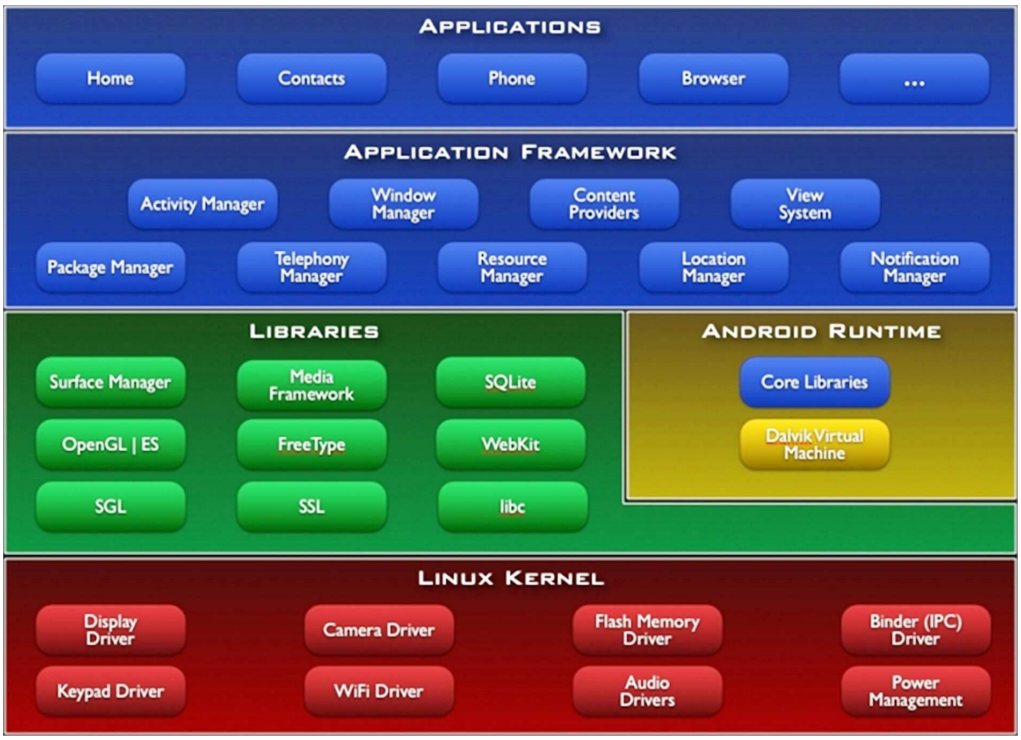
으로 컴퓨터가 읽을 수 있는 코드가 저장되고 실행될 수 있다. 그리고, 본 발명을 구현하기 위한 기능적인(functional) 프로그램, 코드 및 코드 세그먼트들은 본 발명이 속하는 기술분야의 프로그래머들에 의해 용이하게 추론될 수 있다.

[0073] 또한, 상기와 같이 설명된 장치 및 방법은 상기 설명된 실시예들의 구성과 방법이 한정되게 적용될 수 있는 것이 아니라, 상기 실시예들은 다양한 변형이 이루어질 수 있도록 각 실시예들의 전부 또는 일부가 선택적으로 조합되어 구성될 수도 있다.

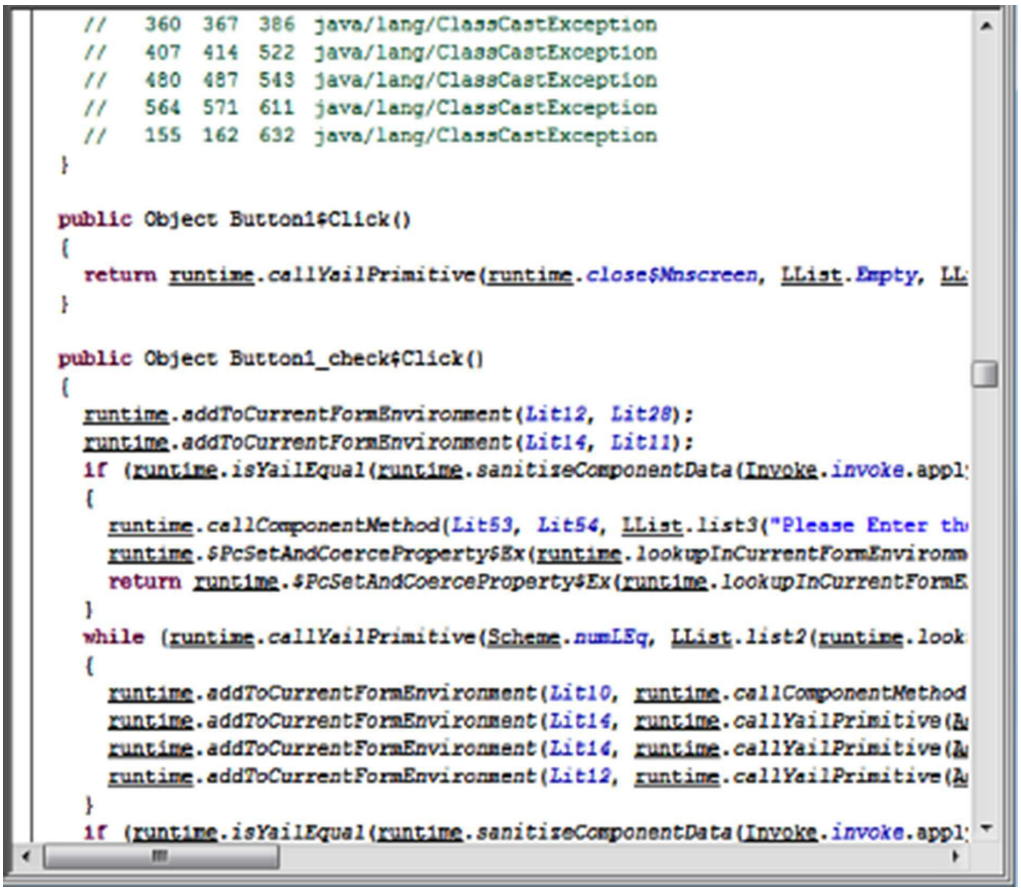
## 도면

### 도면1


### 도면2



도면3



```
    //   360  367  386  java/lang/ClassCastException
    //   407  414  522  java/lang/ClassCastException
    //   480  487  543  java/lang/ClassCastException
    //   564  571  611  java/lang/ClassCastException
    //   155  162  632  java/lang/ClassCastException
  }

  public Object Button1$Click()
  {
    return runtime.callYailPrimitive(runtime.close$Mnscreen, LList.Empty, LL
  }

  public Object Button1_check$Click()
  {
    runtime.addToCurrentFormEnvironment(Lit12, Lit28);
    runtime.addToCurrentFormEnvironment(Lit14, Lit11);
    if (runtime.isYailEqual(runtime.sanitizeComponentData(Invoke.invoke.appl
    {
      runtime.callComponentMethod(Lit53, Lit54, LList.list3("Please Enter th
      runtime.$PcSetAndCoerceProperty$Ex(runtime.lookupInCurrentFormEnvironm
      return runtime.$PcSetAndCoerceProperty$Ex(runtime.lookupInCurrentFormE
    }
    while (runtime.callYailPrimitive(Scheme.numEq, LList.list2(runtime.look
    {
      runtime.addToCurrentFormEnvironment(Lit10, runtime.callComponentMethod
      runtime.addToCurrentFormEnvironment(Lit14, runtime.callYailPrimitive(A
      runtime.addToCurrentFormEnvironment(Lit14, runtime.callYailPrimitive(A
      runtime.addToCurrentFormEnvironment(Lit12, runtime.callYailPrimitive(A
    }
    if (runtime.isYailEqual(runtime.sanitizeComponentData(Invoke.invoke.appl
```

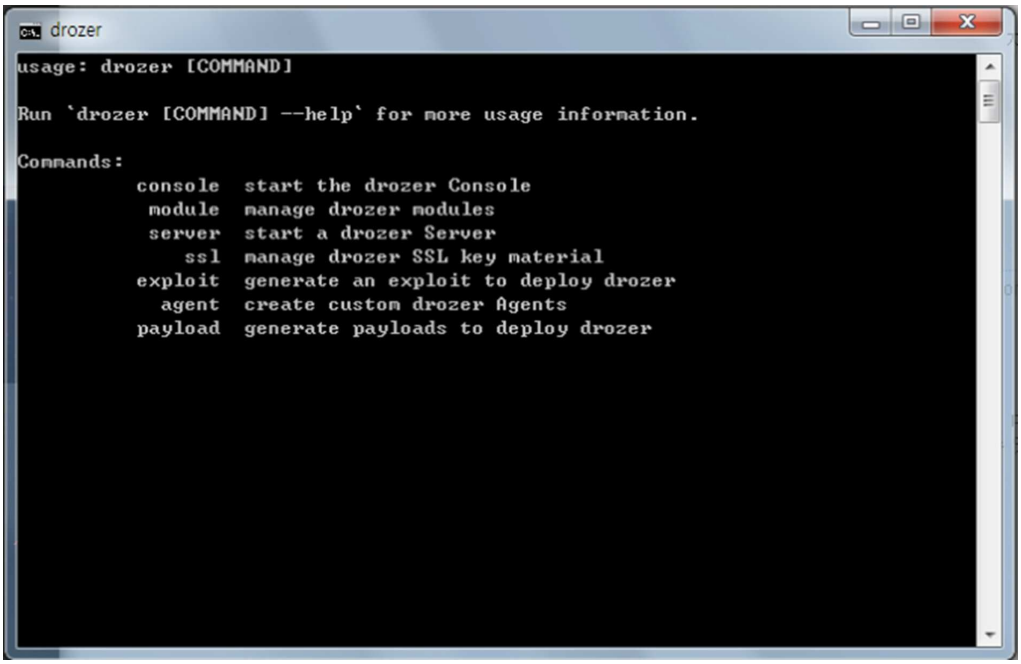
도면4

MainActivity.smali - 메모장

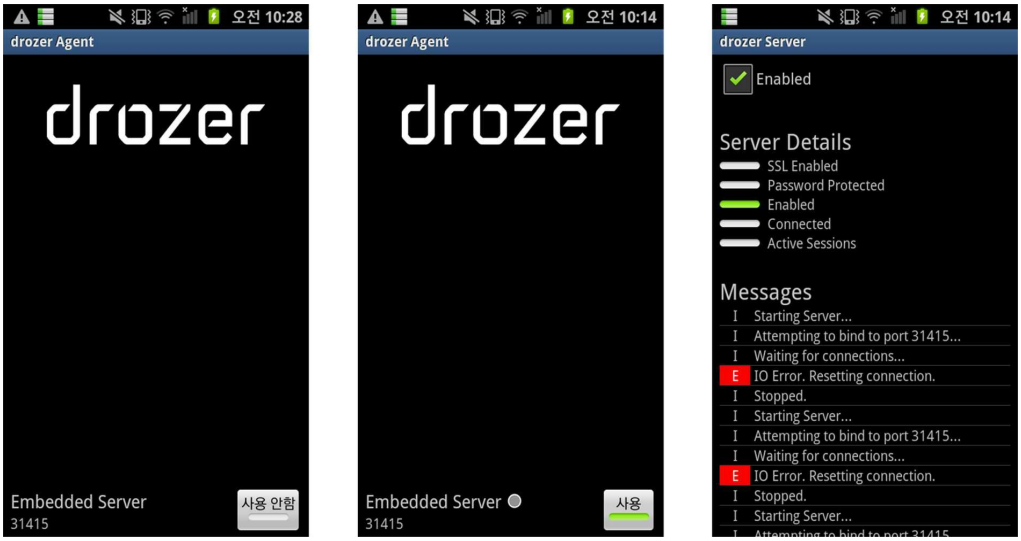
파일(F) 편집(E) 서식(O) 보기(V) 도움말(H)

```
.class public Lkr/pe/sikim/MainActivity;
.super Landroid/app/Activity;
.source "MainActivity.java"

# virtual methods
.method protected onCreate(Landroid/os/Bundle;)V
    .locals 3
    const-string v1, "KSI"

    const-string v0, "kr/pe/sikim/MainActivity : protected onCreate(Landroid/os/Bundle;)V"

    invoke-static {v1, v0}, Landroid/util/Log;->d(Ljava/lang/String;Ljava/lang/String;)I

    .parameter "savedInstanceState"

    .prologue
    .line 12
    invoke-super {p0, p1}, Landroid/app/Activity;->onCreate(Landroid/os/Bundle;)V
```

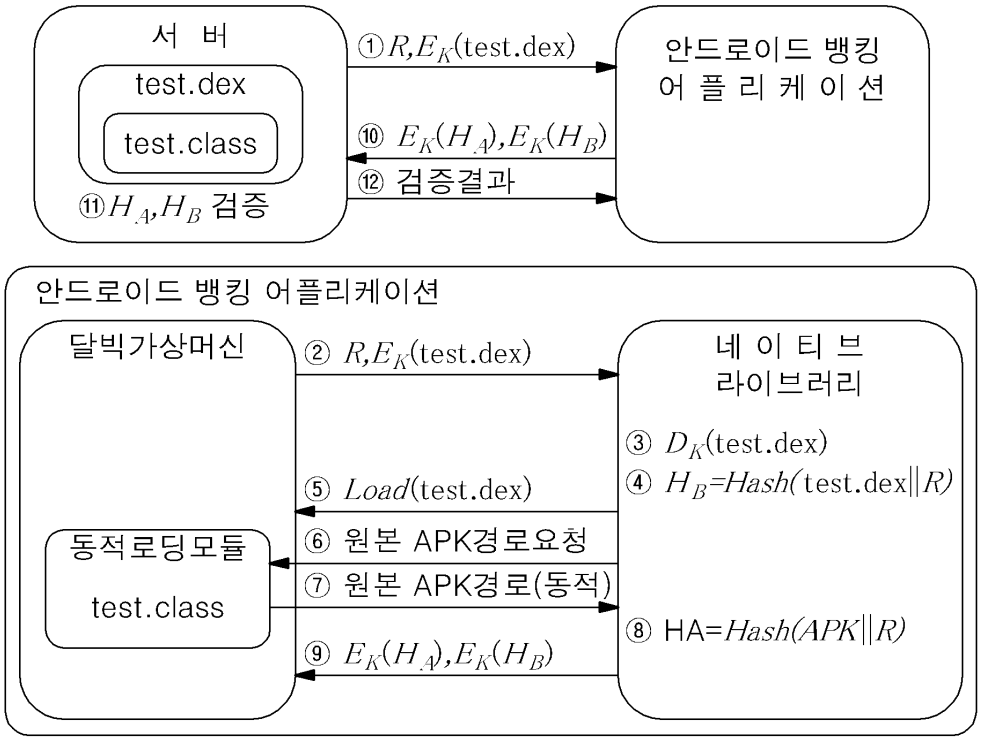
도면5


drozer
usage: drozer [COMMAND]
Run `drozer [COMMAND] --help` for more usage information.
Commands:
console start the drozer Console
module manage drozer modules
server start a drozer Server
ssl manage drozer SSL key material
exploit generate an exploit to deploy drozer
agent create custom drozer Agents
payload generate payloads to deploy drozer

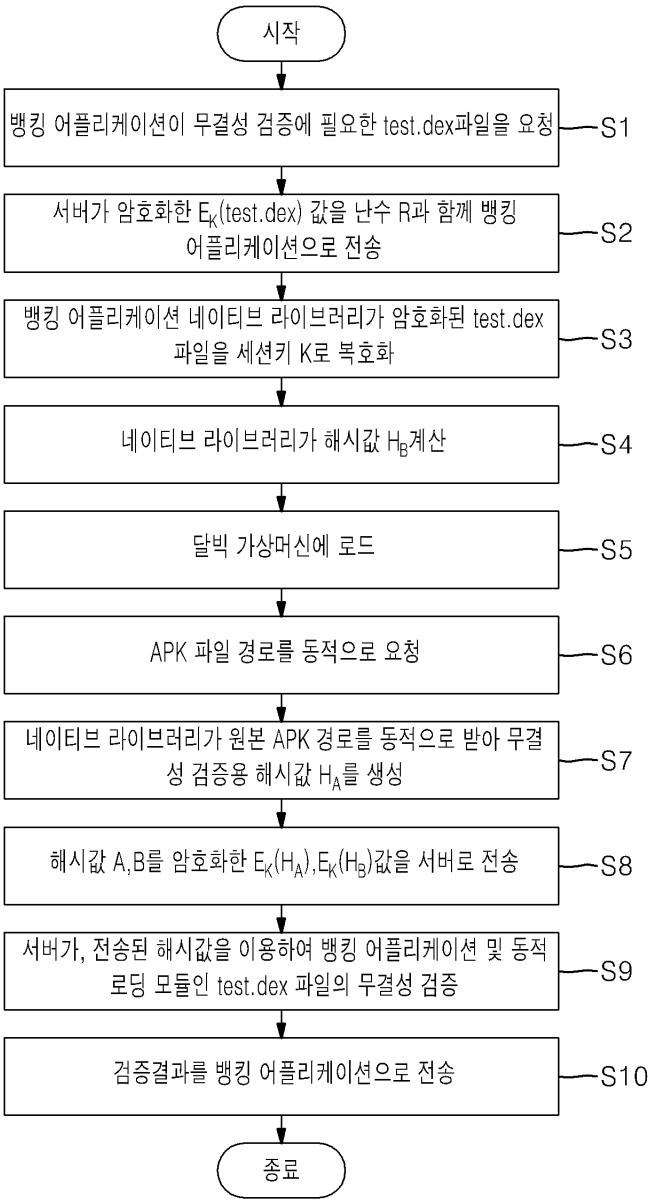
도면6


오전 10:28
drozer Agent
drozer
Embedded Server
31415
사용 안함
오전 10:14
drozer Agent
drozer
Embedded Server
31415
사용
오전 10:14
drozer Server
Enabled
Server Details
SSL Enabled
Password Protected
Enabled
Connected
Active Sessions
Messages
I Starting Server...
I Attempting to bind to port 31415...
I Waiting for connections...
E IO Error. Resetting connection.
I Stopped.
I Starting Server...
I Attempting to bind to port 31415...
I Waiting for connections...
E IO Error. Resetting connection.
I Stopped.
I Starting Server...

도면7


서 버
test.dex
test.class
⑪ $H_A$,$H_B$ 검증
① $R$,$E_K$(test.dex)
⑩ $E_K(H_A)$,$E_K(H_B)$
⑫ 검증결과
안드로이드 뱅킹
어 플 리 케 이 션
안드로이드 뱅킹 어플리케이션
달빅가상머신
② $R$,$E_K$(test.dex)
③ $D_K$(test.dex)
④ $H_B$=Hash(test.dex||$R$)
⑤ Load(test.dex)
⑥ 원본 APK경로요청
⑦ 원본 APK경로(동적)
⑧ HA=Hash(APK||$R$)
⑨ $E_K(H_A)$,$E_K(H_B)$
동적로딩모듈
test.class
네 이 티 브
라이브러리

도면8


시작
뱅킹 어플리케이션이 무결성 검증에 필요한 test.dex파일을 요청
S1
서버가 암호화한 $E_K$(test.dex) 값을 난수 R과 함께 뱅킹 어플리케이션으로 전송
S2
뱅킹 어플리케이션 네이티브 라이브러리가 암호화된 test.dex 파일을 세션키 K로 복호화
S3
네이티브 라이브러리가 해시값 $H_B$계산
S4
달빅 가상머신에 로드
S5
APK 파일 경로를 동적으로 요청
S6
네이티브 라이브러리가 원본 APK 경로를 동적으로 받아 무결성 검증용 해시값 $H_A$를 생성
S7
해시값 A,B를 암호화한 $E_K(H_A)$,$E_K(H_B)$값을 서버로 전송
S8
서버가, 전송된 해시값을 이용하여 뱅킹 어플리케이션 및 동적 로딩 모듈인 test.dex 파일의 무결성 검증
S9
검증결과를 뱅킹 어플리케이션으로 전송
S10
종료