(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2016-0036752
(43) 공개일자 2016년04월05일

(51) 국제특허분류(Int. Cl.)
G06F 21/00 (2006.01) *G06F 21/12* (2013.01)
(21) 출원번호 10-2014-0128573
(22) 출원일자 2014년09월25일
심사청구일자 2014년09월25일

(71) 출원인
주식회사 안랩
경기도 성남시 분당구 판교역로 220 (삼평동)
(72) 발명자
박준용
서울 송파구 올림픽로35길 104, 29동 1009호 (신천동, 장미아파트)
(74) 대리인
제일특허법인

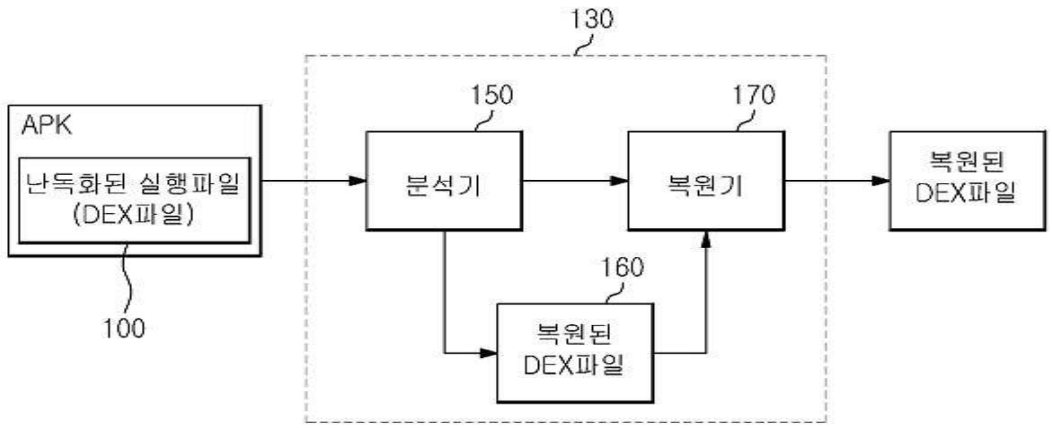
전체 청구항 수 : 총 20 항

(54) 발명의 명칭 **실행파일 복원 장치 및 방법**

(57) 요 약

본 발명에 따르면, 안드로이드 플랫폼의 실행파일에 대한 난독화에 대응하는 실행파일 복원에 있어서, DEX파일 등의 실행파일에 대한 난독화가 수행된 경우, 난독화된 실행파일의 헤더상 클래스 관련 정보를 참조하여 클래스 간 부모-자식 관계를 재귀적으로 설정하는 것을 통해 실행파일내 클래스 정의 아이템에 의해 정의되는 각 클래스에 대한 부모-자식 관계에 대한 정보를 분석함으로써 원래의 클래스 계층구조를 복원시킬 수 있다.

대 표 도 - 도1



명 세 서

청구범위

청구항 1

난독화된 실행파일의 헤더를 참조하여 상기 실행파일에 포함된 각 클래스 정의 아이템의 부모-자식 관계에 대한 정보를 분석하는 분석기와,

상기 분석기에서 분석된 각 클래스 정의 아이템의 상기 부모-자식 관계에 대한 정보를 저장하는 분석결과 저장소와,

상기 분석결과 저장소에 저장된 상기 각 클래스 정의 아이템에 대한 부모-자신 관계에 대한 정보를 이용하여 상기 실행파일의 클래스 계층 구조를 복원하는 복원기

를 포함하는 실행파일 복원장치.

청구항 2

제 1 항에 있어서,

상기 분석기는,

상기 각 클래스 정의 아이템에서 파이널 스테이틱(final static)으로 선언된 클래스 필드(class field)에서 제1 문자열 아이디를 확인하고, 해당 클래스 정의 아이템에서 생성자(contructor)로 선언된 클래스 메소드(class method)에서 제2 문자열 아이디를 확인한 후, 상기 제1 문자열 아이디와 제2 문자열 아이디를 이용하여 상기 부모-자식 관계에 대한 정보를 분석하는 것을 특징으로 하는 실행파일 복원장치.

청구항 3

제 1 항에 있어서,

상기 분석기는,

상기 클래스 필드(class field)의 타입에 대한 정보를 포함하는 상기 제1 문자열 아이디를 확인하고, 상기 클래스 메소드(class method)의 첫 번째 인자에 대한 정보를 포함하는 상기 제2 문자열 아이디를 확인해서 상기 두 개의 문자열 아이디가 같은 경우 상기 클래스 정의 아이템이 정의하는 클래스가 상기 제1 문자열 아이디 및 제2 문자열 아이디에 해당하는 부모 클래스의 자식 클래스인 것으로 분석하는 것을 특징으로 하는 실행파일 복원장치.

청구항 4

제 2 항에 있어서,

상기 분석기는,

상기 실행파일의 헤더에서 상기 실행파일의 각 클래스 정의 아이템이 가지는 클래스 필드 목록을 추출하고, 상기 클래스 필드 목록에서 파이널 스테이틱(final static)으로 선언된 클래스 필드를 검색하는 것을 특징으로 하는 실행파일 복원장치.

청구항 5

제 4 항에 있어서,

상기 분석기는,

상기 검색된 클래스 필드의 필드 아이디 인덱스를 추출하고, 상기 필드 아이디 인덱스를 이용하여 상기 클래스 필드의 디스크립터 아이디를 추출하며, 상기 디스크립터 아이디를 이용하여 상기 제1 문자열 아이디를 추출하는 것을 특징으로 하는 실행파일 복원장치.

청구항 6

제 5 항에 있어서,

상기 분석기는,

상기 필드 아이디 인덱스가 가리키는 헤더상 필드 아이템을 검색하고, 상기 검색된 필드 아이템이 가리키는 타입 아이디 인덱스를 추출하며, 상기 타입 아이디 인덱스가 가리키는 상기 헤더상 타입 아이디를 검색한 후, 상기 검색된 타입 아이디에서 상기 클래스 필드의 디스크립터 아이디를 추출하는 것을 특징으로 하는 실행파일 복원장치.

청구항 7

제 2 항에 있어서,

상기 분석기는,

상기 클래스 정의 아이템의 클래스 메소드 목록에서 생성자로 선언된 클래스 메소드를 추출하고, 상기 클래스 메소드의 상기 첫 번째 인자에 해당하는 프로토 인덱스를 추출하며, 상기 프로토 인덱스를 이용하여 상기 제2 문자열 아이디를 추출하는 것을 특징으로 하는 실행파일 복원장치.

청구항 8

제 7 항에 있어서,

상기 분석기는,

상기 프로토 인덱스가 가리키는 헤더의 프로토 아이디 목록을 추출하고, 상기 프로토 아이디 목록에서 해당 프로토 아이디 아이템의 쇼티 인덱스를 추출하며, 상기 쇼티 인덱스가 가리키는 문자열 아이디 목록을 추출한 후, 상기 문자열 아이디 목록에 포함된 다수의 문자열 아이디 중 첫 번째 문자열 아이디를 상기 제2 문자열 아이디로 추출하는 것을 특징으로 하는 실행파일 복원장치.

청구항 9

제 5 항에 있어서,

상기 디스크립터 아이디는,

상기 헤더상 다수의 문자열을 기록하고 있는 스트링 테이블내 특정 문자열과 대응되는 상기 제1 문자열 아이디를 가리키도록 설정되는 것을 특징으로 하는 실행파일 복원장치.

청구항 10

제 4 항에 있어서,

상기 클래스 필드는,

상기 클래스 필드 목록에서 0x10 비트와 0x1000 비트의 액세스 플래그를 가지는 필드인 것을 특징으로 하는 실

행파일 복원장치.

청구항 11

제 2 항에 있어서,

상기 클래스 메소드는,

상기 클래스 정의 아이템의 클래스 메소드 목록에서 0x10000비트의 액세스 플래그를 가지는 메소드인 것을 특징으로 하는 실행파일 복원장치.

청구항 12

제 1 항에 있어서,

상기 복원기는,

상기 부모-자식 관계에 대한 정보를 이용하여 부모 클래스로 분석된 클래스의 하위에 상기 부모 클래스의 자식 클래스로 분석된 클래스를 정렬시켜 상기 실행파일의 클래스 계층 구조를 복원시키는 것을 특징으로 하는 실행파일 복원장치.

청구항 13

실행파일이 난독화된 경우 상기 실행파일의 헤더에서 상기 실행파일의 각 클래스 정의 아이템이 가지는 클래스 필드 목록을 추출하는 단계와,

상기 클래스 필드 목록에서 파이널 스테이틱으로 선언된 클래스 필드를 검색하는 단계와,

상기 검색된 클래스 필드의 다수의 변수 중에서 상기 클래스 필드의 타입에 대한 정보를 포함하는 제1 문자열 아이디를 추출하는 단계와,

상기 클래스 정의 아이템에서 생성자로 선언된 클래스 메소드의 첫 번째 인자에 대한 정보를 포함하는 제2 문자열 아이디를 추출하는 단계와,

상기 두 개의 문자열 아이디가 같은 경우 상기 클래스 정의 아이템이 정의하는 클래스가 상기 문자열 아이디에 해당하는 부모 클래스의 자식 클래스인 것으로 분석하는 단계와,

상기 분석된 부모 클래스와 자식 클래스의 관계를 이용하여 상기 실행파일의 클래스 계층 구조를 복원하는 단계

를 포함하는 실행파일 복원방법.

청구항 14

제 13 항에 있어서,

상기 제1 문자열 아이디를 추출하는 단계는,

상기 검색된 클래스 필드의 필드 아이디 인덱스를 추출하는 단계와,

상기 필드 아이디 인덱스를 이용하여 상기 클래스 필드의 디스크립터 아이디를 추출하는 단계와,

상기 디스크립터 아이디를 이용하여 상기 제1 문자열 아이디를 추출하는 단계

를 포함하는 것을 특징으로 하는 실행파일 복원방법.

청구항 15

제 14 항에 있어서,

상기 디스크립터 아이디는 추출하는 단계는,

상기 필드 아이디 인덱스가 가리키는 헤더상 필드 아이템을 검색하는 단계와,

상기 검색된 필드 아이템이 가리키는 타입 아이디 인덱스를 추출하는 단계와,

상기 타입 아이디 인덱스가 가리키는 상기 헤더상 타입 아이디를 검색하는 단계와,

상기 검색된 타입 아이디에서 상기 클래스 필드의 디스크립터 아이디를 추출하는 단계

를 포함하는 것을 특징으로 하는 실행파일 복원방법.

청구항 16

제 13 항에 있어서,

상기 제2 문자열 아이디를 추출하는 단계는,

상기 클래스 정의 아이템의 클래스 메소드 목록에서 생성자로 선언된 클래스 메소드를 추출하는 단계와,

상기 생성자로 선언된 클래스 메소드의 상기 첫 번째 인자에 해당하는 프로토 인덱스를 추출하는 단계와,

상기 프로토 인덱스를 이용하여 상기 제2 문자열 아이디를 추출하는 단계

를 포함하는 것을 특징으로 하는 실행파일 복원방법.

청구항 17

제 16 항에 있어서,

상기 제2 문자열 아이디를 추출하는 단계는,

상기 프로토 인덱스가 가리키는 헤더의 프로토 아이디 목록을 추출하는 단계와,

상기 프로토 아이디 목록에서 해당 프로토 아이디 아이템의 쇼티 인덱스를 추출하는 단계와,

상기 쇼티 인덱스가 가리키는 문자열 아이디 목록을 추출하는 단계와,

상기 문자열 아이디 목록에 포함된 다수의 문자열 아이디 중 첫 번째 문자열 아이디를 상기 제2 문자열 아이디로 추출하는 단계

를 포함하는 것을 특징으로 하는 실행파일 복원방법.

청구항 18

제 14 항에 있어서,

상기 디스크립터 아이디는,

상기 헤더상 다수의 문자열을 기록하고 있는 스트링 테이블내 특정 문자열과 대응되는 상기 제1 문자열 아이디를 가리키도록 설정되는 것을 특징으로 하는 실행파일 복원방법.

청구항 19

제 13 항에 있어서,

상기 클래스 필드는,

상기 클래스 필드 목록에서 0x10 비트와 0x1000 비트의 액세스 플래그를 가지는 필드인 것을 특징으로 하는 실

행파일 복원방법.

청구항 20

제 13 항에 있어서,

상기 클래스 메소드는,

상기 클래스 정의 아이템의 클래스 메소드 목록에서 0x10000비트의 액세스 플래그를 가지는 메소드인 것을 특징으로 하는 실행파일 복원방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 난독화된 실행파일 복원에 관한 것으로, 특히 안드로이드 플랫폼의 실행파일에 대한 난독화에 대응하는 실행파일 복원에 있어서, DEX파일 등의 실행파일에 대한 난독화가 수행된 경우, 난독화된 실행파일의 헤더상 클래스(class) 관련 정보를 참조하여 클래스간 부모-자식 관계를 재귀적으로 설정하는 것을 통해 실행파일내 클래스 정의 아이템(class definition item)에 의해 정의되는 각 클래스에 대한 부모-자식 관계에 대한 정보를 분석함으로써 원래의 클래스 계층구조를 복원시킬 수 있도록 하는 실행파일 복원 장치 및 방법에 관한 것이다.

배 경 기 술

[0002] 근래에 들어, 유무선 인터넷뿐만 아니라 이동통신 기술의 발달로, 단순히 전화통화 기능뿐만이 아닌 무선 인터넷 기능 등 다양한 기능을 갖춘 스마트폰, 테블릿 PC 등의 휴대용 단말기가 보급되고 있으며, 이러한 휴대용 단말기는 응용프로그램의 설치 및 삭제가 가능하여 사용자가 필요에 따라 원하는 응용프로그램을 설치하거나 삭제할 수 있도록 하는 등 사용상 편리성이 크게 개선되었으며, 다양한 기능을 제공하고 있다.

[0003] 이러한 스마트폰과, 테블릿 PC 등의 휴대용 단말기는 예를들어 IOS, 안드로이드 등 저마다의 운영체제가 존재하며, 해당 운영체제에 의해 실행 가능한 애플리케이션의 개발이 활발히 이루어지고 있다.

[0004] 위와 같은 스마트폰의 운영체제 중 안드로이드(Android) 플랫폼은 구글(Google) 사가 주도하는 OHA(Open Handset Alliance)에서 공개한 오픈 소스 플랫폼으로, 리눅스(Linux) 커널, 가상머신(VM : Virtual Machine), 프레임워크(Framework), 응용프로그램을 모두 포함하는 소프트웨어 패키지를 의미한다.

[0005] 현재 안드로이드 플랫폼에 대한 사용자들의 기대가 상승하고, 단말 제조사와 이동 통신 서비스 제공사의 높은 호응으로 안드로이드 플랫폼을 탑재하는 스마트폰 등의 휴대용 단말기가 점점 늘어남에 따라 안드로이드 응용프로그램 시장이 활성화되기 시작하였고, 양질의 안드로이드 응용프로그램 공급에 대한 요구가 높아지고 있다.

[0006] 한편, 위와 같이 안드로이드 플랫폼을 탑재한 스마트폰 사용자가 늘어남에 따라 안드로이드 OS를 타겟으로 하는 악성코드 또한 급격히 증가하고 있으며, 안드로이드 악성코드 제작자들은 기존의 PC환경에서 익혔던 다양한 기술을 안드로이드 악성 애플리케이션의 제작에 반영하여 PC에서의 것보다 빠르게 발전시켜 나가고 있다.

[0007] 또한, 최근 들어 이러한 악성코드들은 안티 바이러스 등의 진단장치에 의한 검출이 어렵도록 하기 위해 윈도우 플랫폼에서와 같이 난독화 등의 방법이 사용되고 있다.

[0008] 종래 윈도우 플랫폼의 난독화 기법은 역사가 오래되어 그 복원 방법이 일부 공개된 바 있으나, 안드로이드 플랫폼에서의 난독화에 대한 복원 방법은 아직까지 개발되지 않은 상태이므로, 이러한 안드로이드 플랫폼에서 난독화된 실행파일을 복원하지 못하는 경우 난독화된 실행파일내에 삽입되어 있는 악성코드에 대해서는 진단이 어렵게 되는 문제점이 있다.

[0009] 즉, 예를 들어 실행파일이 난독화된 이후에는 클래스 계층 구조에 대한 정보가 사라질 수 있으며, 이에 따라 난독화되지 않았을 때 클래스 계층구조의 분석을 통해 진단될 수 있는 악성코드가 난독화를 통해 변형이 대량으로 발생하는 경우 악성코드로 진단되지 않는 등 악성코드에 대한 정확한 진단이 어려운 문제점이 있었다.

선행기술문헌

[0010] (특허문헌)

[0011] 대한민국 공개특허번호 10-2010-0010749호(공개일자 2010년 02월 02일)

발명의 내용

해결하려는 과제

[0012] 따라서, 본 발명에서는 안드로이드 플랫폼의 실행파일에 대한 난독화에 대응하는 실행파일 복원에 있어서, DEX 파일 등의 실행파일에 대한 난독화가 수행된 경우, 난독화된 실행파일의 헤더상 클래스 관련 정보를 참조하여 클래스간 부모-자식 관계를 재귀적으로 설정하는 것을 통해 실행파일내 클래스 정의 아이템에 의해 정의되는 각 클래스에 대한 부모-자식 관계에 대한 정보를 분석함으로써 원래의 클래스 계층구조를 복원시킬 수 있도록 하는 실행파일 복원 장치 및 방법을 제공하고자 한다.

과제의 해결 수단

[0013] 상술한 본 발명은 실행파일 복원장치로서, 난독화된 실행파일의 헤더를 참조하여 상기 실행파일에 포함된 각 클래스 정의 아이템의 부모-자식 관계에 대한 정보를 분석하는 분석기와, 상기 분석기에서 분석된 각 클래스 정의 아이템의 상기 부모-자식 관계에 대한 정보를 저장하는 분석결과 저장소와, 상기 분석결과 저장소에 저장된 상기 각 클래스 정의 아이템에 대한 부모-자신 관계에 대한 정보를 이용하여 상기 실행파일의 클래스 계층 구조를 복원하는 복원기를 포함한다.

[0014] 또한, 상기 분석기는, 상기 각 클래스 정의 아이템에서 파이널 스테이틱(final static)으로 선언된 클래스 필드(class field)에서 제1 문자열 아이디를 확인하고, 해당 클래스 정의 아이템에서 생성자(contructor)로 선언된 클래스 메소드(class method)에서 제2 문자열 아이디를 확인한 후, 상기 제1 문자열 아이디와 제2 문자열 아이디를 이용하여 상기 부모-자식 관계에 대한 정보를 분석하는 것을 특징으로 한다.

[0015] 또한, 상기 분석기는, 상기 클래스 필드(class field)의 타입에 대한 정보를 포함하는 상기 제1 문자열 아이디를 확인하고, 상기 클래스 메소드(class method)의 첫 번째 인자에 대한 정보를 포함하는 상기 제2 문자열 아이디를 확인해서 상기 두 개의 문자열 아이디가 같은 경우 상기 클래스 정의 아이템이 정의하는 클래스가 상기 제1 문자열 아이디 및 제2 문자열 아이디에 해당하는 부모 클래스의 자식 클래스인 것으로 분석하는 것을 특징으로 한다.

[0016] 또한, 상기 분석기는, 실행파일이 난독화된 경우 상기 실행파일의 헤더에서 상기 실행파일의 각 클래스 정의 아이템이 가지는 클래스 필드 목록을 추출하고, 상기 클래스 필드 목록에서 파이널 스테이틱(final static)으로 선언된 클래스 필드를 검색하는 것을 특징으로 한다.

[0017] 또한, 상기 분석기는, 상기 검색된 클래스 필드의 필드 아이디 인덱스를 추출하고, 상기 필드 아이디 인덱스를 이용하여 상기 클래스 필드의 디스크립터 아이디를 추출하며, 상기 디스크립터 아이디를 이용하여 상기 제1 문자열 아이디를 추출하는 것을 특징으로 한다.

[0018] 또한, 상기 분석기는, 상기 필드 아이디 인덱스가 가리키는 헤더상 필드 아이템을 검색하고, 상기 검색된 필드 아이템이 가리키는 타입 아이디 인덱스를 추출하며, 상기 타입 아이디 인덱스가 가리키는 상기 헤더상 타입 아이디를 검색한 후, 상기 검색된 타입 아이디에서 상기 클래스 필드의 디스크립터 아이디를 추출하는 것을 특징으로 한다.

[0019] 또한, 상기 분석기는, 상기 클래스 정의 아이템의 클래스 메소드 목록에서 생성자로 선언된 클래스 메소드를 추출하고, 상기 클래스 메소드의 상기 첫 번째 인자에 해당하는 프로토 인덱스를 추출하며, 상기 프로토 인덱스를 이용하여 상기 제2 문자열 아이디를 추출하는 것을 특징으로 한다.

[0020] 또한, 상기 분석기는, 상기 프로토 인덱스가 가리키는 헤더의 프로토 아이디 목록을 추출하고, 상기 프로토 아이디 목록에서 해당 프로토 아이디 아이템의 쇼티 인덱스를 추출하며, 상기 쇼티 인덱스가 가리키는 문자열 아

이디 목록을 추출한 후, 상기 문자열 아이디 목록에 포함된 다수의 문자열 아이디 중 첫 번째 문자열 아이디를 상기 제2 문자열 아이디로 추출하는 것을 특징으로 한다.

[0021] 또한, 상기 디스크립터 아이디는, 상기 헤더상 다수의 문자열을 기록하고 있는 스트링 테이블내 특정 문자열과 대응되는 상기 제1 문자열 아이디를 가리키도록 설정되는 것을 특징으로 한다.

[0022] 또한, 상기 클래스 필드는, 상기 클래스 필드 목록에서 0x10 비트와 0x1000 비트의 액세스 플래그를 가지는 필드인 것을 특징으로 한다.

[0023] 또한, 상기 클래스 메소드는, 상기 클래스 정의 아이템의 클래스 메소드 목록에서 0x10000비트의 액세스 플래그를 가지는 메소드인 것을 특징으로 한다.

[0024] 또한, 상기 복원기는, 상기 부모-자식 관계에 대한 정보를 이용하여 부모 클래스로 분석된 클래스의 하위에 상기 부모 클래스의 자식 클래스로 분석된 클래스를 정렬시켜 상기 실행파일의 클래스 계층 구조를 복원시키는 것을 특징으로 한다.

[0025] 또한, 본 발명은 실행파일 복원방법으로서, 실행파일이 난독화된 경우 상기 실행파일의 헤더에서 상기 실행파일의 각 클래스 정의 아이템이 가지는 클래스 필드 목록을 추출하는 단계와, 상기 클래스 필드 목록에서 파이널 스테이틱으로 선언된 클래스 필드를 검색하는 단계와, 상기 검색된 클래스 필드의 다수의 변수 중에서 상기 클래스 필드의 타입에 대한 정보를 포함하는 제1 문자열 아이디를 추출하는 단계와, 상기 클래스 정의 아이템에서 생성자로 선언된 클래스 메소드의 첫 번째 인자에 대한 정보를 포함하는 제2 문자열 아이디를 추출하는 단계와, 상기 두 개의 문자열 아이디가 같은 경우 상기 클래스 정의 아이템이 정의하는 클래스가 상기 문자열 아이디에 해당하는 부모 클래스의 자식 클래스인 것으로 분석하는 단계와, 상기 분석된 부모 클래스와 자식 클래스의 관계를 이용하여 상기 실행파일의 클래스 계층 구조를 복원하는 단계를 포함한다.

[0026] 또한, 상기 제1 문자열 아이디를 추출하는 단계는, 상기 검색된 클래스 필드의 필드 아이디 인덱스를 추출하는 단계와, 상기 필드 아이디 인덱스를 이용하여 상기 클래스 필드의 디스크립터 아이디를 추출하는 단계와, 상기 디스크립터 아이디를 이용하여 상기 제1 문자열 아이디를 추출하는 단계를 포함하는 것을 특징으로 한다.

[0027] 또한, 상기 디스크립터 아이디는 추출하는 단계는, 상기 필드 아이디 인덱스가 가리키는 헤더상 필드 아이템을 검색하는 단계와, 상기 검색된 필드 아이템이 가리키는 타입 아이디 인덱스를 추출하는 단계와, 상기 타입 아이디 인덱스가 가리키는 상기 헤더상 타입 아이디를 검색하는 단계와, 상기 검색된 타입 아이디에서 상기 클래스 필드의 디스크립터 아이디를 추출하는 단계를 포함하는 것을 특징으로 한다.

[0028] 또한, 상기 제2 문자열 아이디를 추출하는 단계는, 상기 클래스 정의 아이템의 클래스 메소드 목록에서 생성자로 선언된 클래스 메소드를 추출하는 단계와, 상기 생성자로 선언된 클래스 메소드의 상기 첫 번째 인자에 해당하는 프로토 인덱스를 추출하는 단계와, 상기 프로토 인덱스를 이용하여 상기 제2 문자열 아이디를 추출하는 단계를 포함하는 것을 특징으로 한다.

[0029] 또한, 상기 제2 문자열 아이디를 추출하는 단계는, 상기 프로토 인덱스가 가리키는 헤더의 프로토 아이디 목록을 추출하는 단계와, 상기 프로토 아이디 목록에서 해당 프로토 아이디 아이템의 쇼티 인덱스를 추출하는 단계와, 상기 쇼티 인덱스가 가리키는 문자열 아이디 목록을 추출하는 단계와, 상기 문자열 아이디 목록에 포함된 다수의 문자열 아이디 중 첫 번째 문자열 아이디를 상기 제2 문자열 아이디로 추출하는 단계를 포함하는 것을 특징으로 한다.

[0030] 또한, 상기 디스크립터 아이디는, 상기 헤더상 다수의 문자열을 기록하고 있는 스트링 테이블내 특정 문자열과 대응되는 상기 제1 문자열 아이디를 가리키도록 설정되는 것을 특징으로 한다.

[0031] 또한, 상기 클래스 필드는, 상기 클래스 필드 목록에서 0x10 비트와 0x1000 비트의 액세스 플래그를 가지는 필드인 것을 특징으로 한다.

[0032] 또한, 상기 클래스 메소드는, 상기 클래스 정의 아이템의 클래스 메소드 목록에서 0x10000비트의 액세스 플래그를 가지는 메소드인 것을 특징으로 한다.

발명의 효과

[0033] 본 발명에 따르면, 안드로이드 플랫폼의 실행파일에 대한 난독화에 대응하는 실행파일 복원에 있어서, DEX파일

등의 실행파일에 대한 난독화가 수행된 경우, 난독화된 실행파일의 헤더상 클래스 관련 정보를 참조하여 클래스 간 부모-자식 관계를 재귀적으로 설정하는 것을 통해 실행파일내 클래스 정의 아이템에 의해 정의되는 각 클래스에 대한 부모-자식 관계에 대한 정보를 분석함으로써 원래의 클래스 계층구조를 복원시킬 수 있는 이점이 있다.

[0034] 또한, 위와 같은 복원방법을 적용하는 경우 난독화 되지 않았을 때 클래스 계층 구조로 진단되는 악성코드가 난독화를 통해 변형이 대량으로 발생하는 경우에도 난독화를 복원한 후 백신의 기존 진단 방법과 조합하여 악성코드로 모두 진단할 수 있는 이점이 있다. 또한, 난독화 악성코드의 복원방법을 자동화하여 침해사고의 사후 대응과 같은 대응 시간이 매우 중요한 경우 대응시간을 대폭 단축시킬 수 있어 서비스 품질을 높일 수 있는 이점이 있다.

도면의 간단한 설명

[0035] 도 1은 본 발명의 실시예에따른 실행파일 복원 장치의 상세 블록 구성도,

도 2는 본 발명의 실시예예 따른 실행파일 복원장치에서 클래스 계층구조를 복원하는 동작 제어 흐름도,

도 3은 종래 상용 디컴파일러가 난독화된 실행파일에 대한 복원을 수행한 결과 화면 예시도,

도 4는 본 발명의 실시예에 따른 실행파일 복원장치에서 난독화된 실행파일에 대한 복원을 수행한 결과 화면 예시도.

발명을 실시하기 위한 구체적인 내용

[0036] 이하, 첨부된 도면을 참조하여 본 발명의 동작 원리를 상세히 설명한다. 하기에서 본 발명을 설명함에 있어서 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다. 그리고 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.

[0037] 첨부된 블록도의 각 블록과 흐름도의 각 단계의 조합들은 컴퓨터 프로그램 인스트럭션들에 의해 수행될 수도 있다. 이들 컴퓨터 프로그램 인스트럭션들은 범용 컴퓨터, 특수용 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서에 탑재될 수 있으므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비의 프로세서를 통해 수행되는 그 인스트럭션들이 블록도의 각 블록 또는 흐름도의 각 단계에서 설명된 기능들을 수행하는 수단을 생성하게 된다. 이들 컴퓨터 프로그램 인스트럭션들은 특정 방식으로 기능을 구현하기 위해 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 지향할 수 있는 컴퓨터 이용 가능 또는 컴퓨터 판독 가능 메모리에 저장되는 것도 가능하므로, 그 컴퓨터 이용가능 또는 컴퓨터 판독 가능 메모리에 저장된 인스트럭션들은 블록도의 각 블록 또는 흐름도 각 단계에서 설명된 기능을 수행하는 인스트럭션 수단을 내포하는 제조 품목을 생산하는 것도 가능하다. 컴퓨터 프로그램 인스트럭션들은 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에 탑재되는 것도 가능하므로, 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비 상에서 일련의 동작 단계들이 수행되어 컴퓨터로 실행되는 프로세스를 생성해서 컴퓨터 또는 기타 프로그램 가능한 데이터 프로세싱 장비를 수행하는 인스트럭션들은 블록도의 각 블록 및 흐름도의 각 단계에서 설명된 기능들을 실행하기 위한 단계들을 제공하는 것도 가능하다.

[0038] 또한, 각 블록 또는 각 단계는 특정된 논리적 기능(들)을 실행하기 위한 하나 이상의 실행 가능한 인스트럭션들을 포함하는 모듈, 세그먼트 또는 코드의 일부를 나타낼 수 있다. 또, 몇 가지 대체 실시 예들에서는 블록들 또는 단계들에서 언급된 기능들이 순서를 벗어나서 발생하는 것도 가능함을 주목해야 한다. 예컨대, 잇달아 도시되어 있는 두 개의 블록들 또는 단계들은 사실 실질적으로 동시에 수행되는 것도 가능하고 또는 그 블록들 또는 단계들이 때때로 해당하는 기능에 따라 역순으로 수행되는 것도 가능하다.

[0039] 도 1은 본 발명의 실시 예에 따른 실행파일 복원 장치(130)의 상세 블록 구성을 도시한 것으로, 분석기(150), 분석결과 저장소(160), 복원기(170) 등을 포함할 수 있다. 이하, 도 1을 참조하여 본 발명의 실행파일 복원 장치(130)의 각 구성요소에서의 동작을 상세히 설명하기로 한다.

[0040] 먼저, 분석기(150)는 난독화된 실행파일 예를 들어 DEX 파일의 헤더(header)를 참조하여 실행파일에 포함된 각

클래스 정의 아이템(class definition item)에 대한 부모-자식 관계를 분석한다.

[0041] 이하, 분석기(150)에서의 동작을 보다 자세히 설명하면, 분석기(150)는 각 클래스 정의 아이템에서 파이널 스테이틱(final static)으로 선언된 클래스 필드(class field)의 타입(type)에 대한 정보를 포함하는 제1 문자열 아이디를 확인한다. 이때, 위와 같은 필드는 클래스 정의 아이템이 가지는 다양한 변수를 의미할 수 있으며, 이러한 필드에는 액세스 플래그(access flag), 타입(type), 이름(name) 등의 정보가 포함될 수 있다.

[0042] 이어, 분석기(150)는 해당 클래스 정의 아이템의 메소드(method) 중 생성자(constructor)로 선언된 클래스 메소드의 첫 번째 인자에 대한 정보를 포함하는 제2 문자열 아이디를 확인하고, 제1 문자열 아이디와 제2 문자열 아이디를 비교한다. 이때, 제1 문자열 아이디와 제2 문자열 아이디가 같은 경우 분석기(150)는 클래스 정의 아이템이 정의하는 해당 클래스가 문자열 아이디에 해당하는 부모 클래스(parent class)의 자식 클래스(child class)인 것으로 분석하게 된다.

[0043] 또한, 이때 분석기(150)는 파이널 스테이틱으로 선언된 클래스 필드를 검색함에 있어서, 실행파일인 DEX파일 등이 난독화된 경우 실행파일의 헤더상 각 클래스 정의 아이템이 가지는 클래스 필드 목록을 추출하고, 클래스 필드 목록에서 파이널 스테이틱(final static)으로 선언된 클래스 필드를 검색할 수 있다. 이때, 이러한 클래스 필드 목록에는 정적 필드 목록과 인스턴스 필드 목록이 포함될 수 있으며, 파이널 스테이틱으로 선언된 클래스 필드는 정적 필드 목록에 포함될 수 있다. 따라서 분석기(150)는 예를 들어 클래스 필드 목록의 정적 필드 목록에서 파이널 스테이틱으로 선언된 클래스 필드를 검색할 수 있다.

[0044] 이때, 위와 같이 파이널 스테이틱으로 선언된 클래스 필드는 클래스 필드 목록에서 0x00비트 및 0x1000비트의 액세스 플래그를 가질 수 있으며, 분석기(150)는 클래스 필드 목록에서 해당 비트의 액세스 플래그를 검색하여 파이널 스테이틱으로 선언된 클래스 필드를 검색할 수 있다.

[0045] 또한, 분석기(150)는, 위와 같이 클래스 필드를 검색한 후, 검색된 클래스 필드를 이용하여 필드 아이디 인덱스를 추출하고, 필드 아이디 인덱스를 이용하여 해당 클래스 필드의 디스크립터(discriptor) 아이디를 추출하며, 다시 디스크립터 아이디를 이용하여 필드 문자열 아이디인 제1 문자열 아이디를 추출할 수 있다.

[0046] 이때, 분석기(150)는 디스크립터 아이디를 추출함에 있어서, 필드 아이디 인덱스가 가리키는 헤더상 필드 아이템을 검색하고, 다시 검색된 필드 아이템이 가리키는 타입 아이디 인덱스를 추출하며, 타입 아이디 인덱스가 가리키는 헤더상 타입 아이디를 검색한 후, 검색된 타입 아이디에서 클래스 필드의 디스크립터 아이디를 추출할 수 있다.

[0047] 이때, 이러한 디스크립터 아이디는 헤더상 다수의 문자열을 기록하고 있는 스트링 테이블(string table)내 특정 문자열과 대응되는 제1 문자열 아이디를 가리키도록 설정될 수 있다. 따라서, 분석기(150)는 디스크립터 아이디를 이용하여 헤더상 스트링 테이블(string table) 저장된특정 문자열을 읽어올 수 있는 것이다.

[0048] 또한, 분석기(150)는 제2 문자열 아이디를 추출함에 있어서, 해당 클래스 정의 아이템의 클래스 메소드 목록에서 생성자(constructor)로 선언된 클래스 메소드를 추출하고, 생성자로 선언된 메소드의 첫 번째 인자에 해당하는 프로토(proto) 인덱스를 추출하며, 프로토 인덱스를 이용하여 프로토 문자열 아이디인 제2 문자열 아이디를 추출할 수 있다. 이때, 이러한 생성자로 선언된 클래스 메소드는 클래스 정의 아이템의 클래스 메소드 목록에서 0x10000비트의 액세스 플래그를 가지는 메소드를 말할 수 있다.

[0049] 분석결과 저장소(160)는 분석기(150)에서 분석된 각 클래스 정의 아이템의 부모-자식 관계에 대한 정보를 저장한다.

[0050] 복원기(170)는 실행파일 복원장치(130)의 분석기(150)와 분석결과 저장소(160)와 연결되며, 분석기(150)를 통해 분석된 실행파일의 각 클래스의 부모-자식 관계에 대한 정보가 분석결과 저장소(160)에 저장되는 경우, 분석결과 저장소(160)에 저장된 각 클래스 정의 아이템에 대한 부모-자식 관계에 대한 정보를 이용하여 난독화된 실행파일의 클래스 계층 구조를 복원한다. 즉, 복원기(170)는, 예를 들어 부모 클래스로 분석된 클래스의 하위에 부모 클래스의 자식 클래스로 분석된 클래스를 정렬시켜 실행파일의 클래스 계층 구조를 복원시키게 된다.

[0051] 도 2는 본 발명의 실시예예 따른 실행파일 복원장치에서 클래스 계층구조를 복원하는 동작 제어 흐름을 도시한 것이다. 이하, 도 1 및 도 2를 참조하여 본 발명의 실시예를 상세히 설명하기로 한다.

[0052] 먼저, 안드로이드 플랫폼에서 DEX 파일 등의 실행파일이 난독화되어 인가되는 경우 분석기(150)는 난독화된 실행파일(100)의 헤더(header)상 클래스 정의 아이템 목록을 추출한다(S200). 이어, 분석기(150)는 클래스 정의 아이템 목록에서 각 클래스 정의 아이템이 가지는 정적 필드 목록과 인스턴스 필드 목록을 클래스 필드 목록으

로 추출한다(S202).

[0053] 그런 후, 분석기(150)는 위와 같이 추출한 클래스 필드 목록에서 파이널 스테이틱(final static)으로 선언된 클래스 필드를 검색한다(S204). 이때, 이러한 클래스 필드 목록에는 위에서 설명한 바와 같이 정적 필드 목록과 인스턴스 필드 목록이 포함될 수 있으며, 파이널 스테이틱으로 선언된 클래스 필드는 정적 필드 목록에 포함될 수 있다. 따라서, 분석기(150)는 예를 들어 클래스 필드 목록의 정적 필드 목록에서 파이널 스테이틱으로 선언된 클래스 필드를 검색할 수 있다.

[0054] 또한, 이때 위와 같이 파이널 스테이틱으로 선언된 클래스 필드는 클래스 필드 목록에서 0x00비트 및 0x1000비트의 액세스 플래그를 가질 수 있으며, 분석기(150)는 클래스 필드 목록에서 해당 비트의 액세스 플래그를 검색하여 파이널 스테이틱으로 선언된 클래스 필드를 검색할 수 있다.

[0055] 위와 같이 파이널 스테이틱으로 선언된 클래스 필드를 검색한 경우, 분석기(150)는 검색된 클래스 필드를 이용하여 필드 아이디 인덱스를 추출한다(S206).

[0056] 이어, 분석기(150)는 필드 아이디 인덱스를 이용하여 필드 아이디 인덱스가 가리키는 헤더상 필드 아이템을 검색하고, 다시 검색된 필드 아이템이 가리키는 타입 아이디 인덱스를 추출한다(S208).

[0057] 위와 같이 타입 아이디 인덱스가 추출되는 경우, 분석기(150)는 다시 타입 아이디 인덱스가 가리키는 헤더상 타입 아이디를 검색한 후, 검색된 타입 아이디에서 클래스 필드의 디스크립터 아이디를 추출한다(S210).

[0058] 이어, 분석기(150)는 위와 같이 추출한 해당 클래스 필드의 디스크립터 아이디를 이용하여 필드 문자열 아이디인 제1 문자열 아이디를 추출할 수 있다(S212).

[0059] 이때, 위와 같은 디스크립터 아이디는 헤더상 다수의 문자열을 기록하고 있는 스트링 테이블내 특정 문자열과 대응되는 제1 문자열 아이디를 가리키도록 설정될 수 있으며, 분석기(150)는 이러한 제1 문자열 아이디를 이용하여 스트링 테이블내 기록된 특정 문자열을 읽어올 수 있게 된다. 즉, 분석기(150)는 제1 문자열 아이디를 이용하여 읽어온 문자열을 통해 각 클래스 정의 아이템에서 파이널 스테이틱(final static)으로 선언된 클래스 필드의 타입에 대한 정보 등을 확인할 수 있게 된다.

[0060] 이어, 분석기(150)는 해당 클래스 정의 아이템이 가지는 직접 메소드 목록과 가상 메소드 목록을 클래스 메소드 목록으로 추출하고, 클래스 메소드 목록에서 생성자(constructor)로 선언된 클래스 메소드를 추출한다(S214).

[0061] 이어, 분석기(150)는 위와 같이 추출된 클래스 메소드를 이용하여 클래스 메소드의 첫 번째 인자에 해당하는 프로토(proto) 인덱스를 추출한다(S216).

[0062] 위와 같이 프로토 인덱스를 추출한 경우 분석기(150)는 다시 프로토 인덱스가 가리키는 헤더의 프로토 아이디 목록을 추출하고, 프로토 아이디 목록에서 해당 프로토 아이디 아이템의 쇼티 인덱스를 추출한 후, 쇼티 인덱스가 가리키는 문자열 아이디 목록을 추출한다(S218).

[0063] 이어, 분석기(150)는 문자열 아이디 목록에 포함된 다수의 문자열 아이디 중 첫 번째 문자열 아이디를 프로토 문자열 아이디인 제2 문자열 아이디로 추출한다(S220). 이때, 이러한 프로토 인덱스는 클래스 정의 아이템의 클래스 메소드 중 0x10000비트의 액세스 플래그를 가지는 클래스 메소드의 인덱스 정보를 말할 수 있다.

[0064] 위와 같이 제1 문자열 아이디와 제2 문자열 아이디를 추출한 경우, 분석기(150)는 제1 문자열 아이디와 제2 문자열 아이디를 비교하여 동일한 지 여부를 검사한다(S222).

[0065] 이때, 제1 문자열 아이디와 제2 문자열 아이디가 같은 경우 분석기(150)는 클래스 정의 아이템이 정의하는 해당 클래스가 제1 문자열 아이디와 제2 문자열 아이디에 해당하는 부모 클래스의 자식 클래스인 것으로 분석하게 된다(S224). 이때, 제1 문자열 아이디와 제2 문자열 아이디에는 부모 클래스의 이름 등과 같은 식별정보가 기록되어 있을 수 있다.

[0066] 이어, 분석기(150)는 위와 같은 과정을 통해 분석한 각 클래스 정의 아이템의 부모-자식 관계에 대한 정보를 분석결과 저장소(160)에 저장한다(S226).

[0067] 그러면, 복원기(170)에서는 분석결과 저장소(160)에 저장된 각 클래스 정의 아이템에 대한 부모-자식 관계에 대한 정보를 이용하여 난독화된 실행파일의 클래스 계층 구조를 복원한다(S228). 이때, 복원기는, 예를 들어 부모 클래스로 분석된 클래스의 하위에 부모 클래스의 자식 클래스로 분석된 클래스를 정렬시켜 실행파일의 클래스 계층 구조를 복원시키게 된다.

[0068] 도 3은 종래 상용 디컴파일러가 난독화된 실행파일에 대한 복원을 수행한 결과 화면을 예시한 것이다.

[0069] 도 3에서와 보여지는 바와 같이 종래 디컴파일러(de-complier)에서 수행된 파일 복원 결과에서는 난독화로 인해 자식 클래스 등이 알파벳 한 글자로 바뀐 것을 확인할 수 있다. 또한, c, d, e 클래스가 GoogleSyncServiceA 클래스의 자식 클래스로 분석되지 않은 것을 볼 수 있으며, i, j, k, l, m, n 클래스가 NotifyActivity 클래스의 자식 클래스로 분석되지 않은 것을 볼 수 있는 등 계층 구조 복원에 실패한 모습을 확인할 수 있다.

[0070] 도 4는 본 발명의 실시예에 따른 실행파일 복원장치에서 난독화된 실행파일에 대한 복원을 수행한 결과 화면을 예시한 것이다.

[0071] 도 4에서 보여지는 바와 같이 본 발명의 실시예에 따른 복원방법을 통해서는 난독화된 실행파일의 역분석을 통해 c, d, e 클래스가 GoogleSyncServiceA 클래스의 자식 클래스로 분석된 것을 볼 수 있으며, i, j, k, l, m, n 클래스 또한 NotifyActivity 클래스의 자식 클래스로 분석된 것을 볼 수 있다.

[0072] 따라서, 난독화에도 불구하고 자식 클래스가 모두 부모 클래스를 찾아 계층 구조로 표시됨으로써 실행파일의 원래의 계층 구조 복원에 성공한 모습을 확인할 수 있다.

[0073] 상기한 바와 같이, 안드로이드 플랫폼의 실행파일에 대한 난독화에 대응하는 실행파일 복원에 있어서, DEX파일 등의 실행파일에 대한 난독화가 수행된 경우, 난독화된 실행파일의 헤더상 클래스 관련 정보를 참조하여 클래스 간 부모-자식 관계를 재귀적으로 설정하는 것을 통해 실행파일내 클래스 정의 아이템에 의해 정의되는 각 클래스에 대한 부모-자식 관계에 대한 정보를 분석함으로써 원래의 클래스 계층구조를 복원시킬 수 있다.

[0074] 한편 상술한 본 발명의 설명에서는 구체적인 실시예에 관해 설명하였으나, 여러 가지 변형이 본 발명의 범위에서 벗어나지 않고 실시될 수 있다. 따라서 발명의 범위는 설명된 실시 예에 의하여 정할 것이 아니고 특허청구범위에 의해 정하여져야 한다.

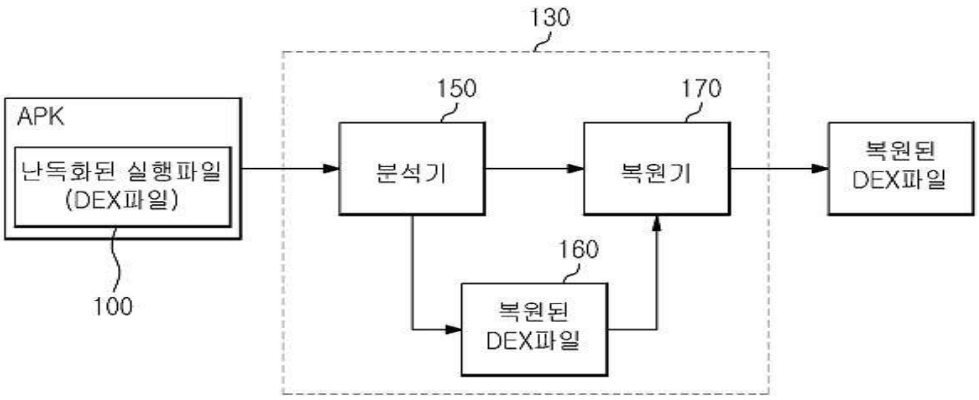
부호의 설명

[0075] 150 : 분석기 160 : 분석결과 저장소

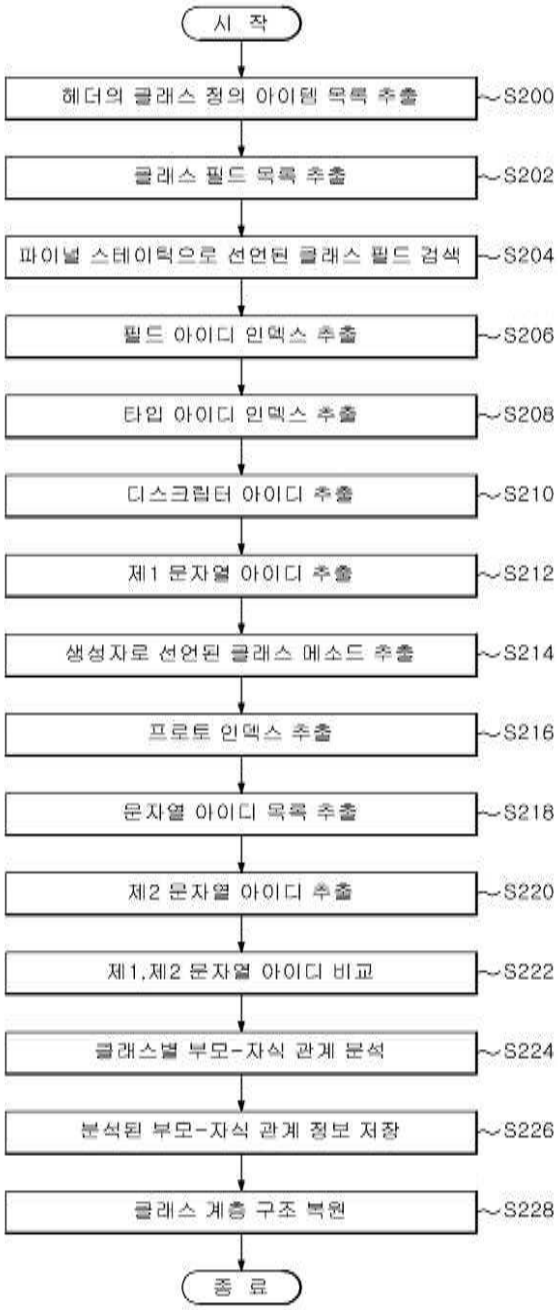
170 : 복원기

도면

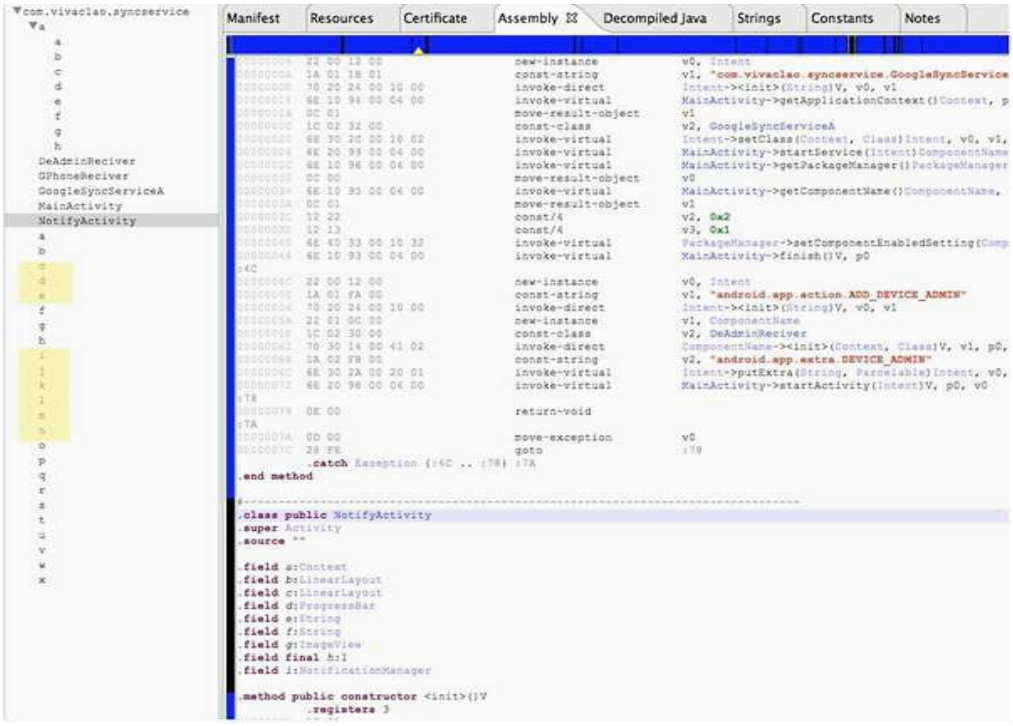
도면1



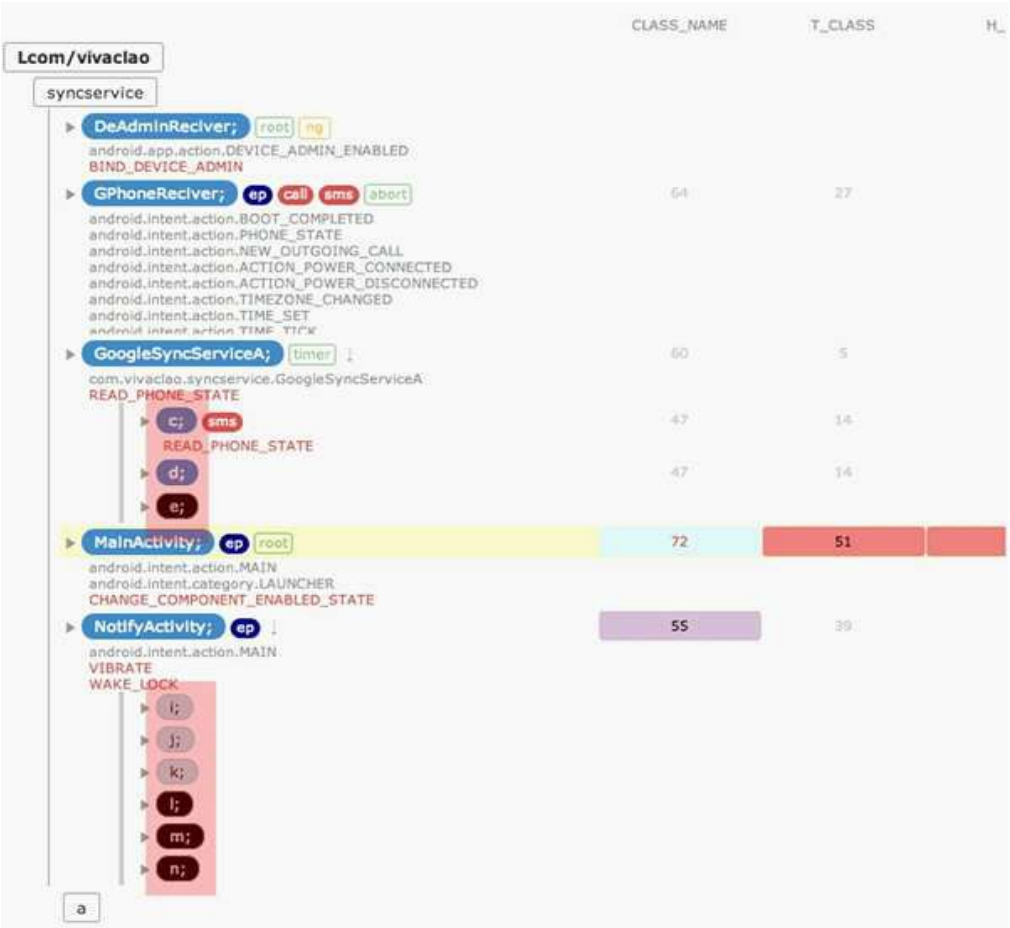
도면2


시 작
헤더의 클래스 정의 아이템 목록 추출
S200
클래스 필드 목록 추출
S202
파이널 스테이틱으로 선언된 클래스 필드 검색
S204
필드 아이디 인덱스 추출
S206
타입 아이디 인덱스 추출
S208
디스크립터 아이디 추출
S210
제1 문자열 아이디 추출
S212
생성자로 선언된 클래스 메소드 추출
S214
프로토 인덱스 추출
S216
문자열 아이디 목록 추출
S218
제2 문자열 아이디 추출
S220
제1,제2 문자열 아이디 비교
S222
클래스별 부모-자식 관계 분석
S224
분석된 부모-자식 관계 정보 저장
S226
클래스 계층 구조 복원
S228
종 료

도면3


Manifest
Resources
Certificate
Assembly
Decompiled Java
Strings
Constants
Notes
.class public NotifyActivity
.super Activity
.source ""
.field a:Context
.field b:LinearLayout
.field c:LinearLayout
.field d:ProgressBar
.field e:String
.field f:String
.field g:ImageView
.field final h:I
.field i:NotificationManager
.method public constructor <init>()V
.registers 3
.end method

도면4


CLASS_NAME
T_CLASS
H_
Lcom/vivaclao
syncservice
DeAdminReciver;
root
ng
android.app.action.DEVICE_ADMIN_ENABLED
BIND_DEVICE_ADMIN
GPhoneReciver;
ep
call
sms
abort
64
27
android.intent.action.BOOT_COMPLETED
android.intent.action.PHONE_STATE
android.intent.action.NEW_OUTGOING_CALL
android.intent.action.ACTION_POWER_CONNECTED
android.intent.action.ACTION_POWER_DISCONNECTED
android.intent.action.TIMEZONE_CHANGED
android.intent.action.TIME_SET
GoogleSyncServiceA;
timer
60
5
com.vivaclao.syncservice.GoogleSyncServiceA
READ_PHONE_STATE
c;
sms
47
14
READ_PHONE_STATE
d;
47
14
e;
MainActivity;
ep
root
72
51
android.intent.action.MAIN
android.intent.category.LAUNCHER
CHANGE_COMPONENT_ENABLED_STATE
NotifyActivity;
ep
55
39
android.intent.action.MAIN
VIBRATE
WAKE_LOCK
i;
j;
k;
l;
m;
n;
a