(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2015년06월09일
(11) 등록번호 10-1527098
(24) 등록일자 2015년06월02일

(51) 국제특허분류(Int. Cl.)
G06F 21/55 (2013.01) *G06F 11/36* (2006.01)
G06F 21/56 (2013.01)
(21) 출원번호 10-2013-0157471
(22) 출원일자 2013년12월17일
심사청구일자 2013년12월17일
(65) 공개번호 10-2015-0026716
(43) 공개일자 2015년03월11일
(30) 우선권주장
1020130102693 2013년08월28일 대한민국(KR)
(56) 선행기술조사문헌
KR1020040080845 A*
KR1020120081360 A*
KR101290565 B1
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
고려대학교 산학협력단

(72) 발명자
이희조
이찬영

(뒷면에 계속)

(74) 대리인
특허법인엠에이피에스

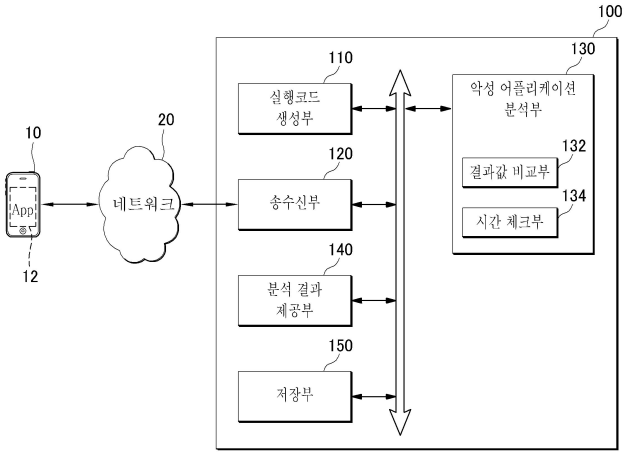
전체 청구항 수 : 총 10 항 심사관 : 구본재

(54) 발명의 명칭 **랜덤 실행 코드를 이용한 스마트 기기 내 어플리케이션 검증 서버 및 검증방법**

(57) 요 약

본 발명은 어플리케이션 검증 서버 및 그 검증방법에 대해 개시한다. 특히, 스마트 기기 내 소정의 어플리케이션을 검증하는 서버는, 상기 어플리케이션에 대한 검증용 실행코드를 생성하되, 상기 어플리케이션과 관련된 정보를 무작위로 조합하여 상기 실행코드를 생성하는 실행코드 생성부; 상기 스마트 기기로부터 검증 요청신호를 수신하면 상기 생성된 실행코드 중 무작위로 선택된 실행코드를 상기 스마트 기기로 송신하고, 상기 스마트 기기에서 상기 어플리케이션에 대해 상기 선택된 실행코드를 실행시킨 결과값을 수신하는 송수신부; 상기 수신된 결과값을 기초로 상기 어플리케이션이 악성 어플리케이션인지 여부를 분석하는 악성 어플리케이션 분석부; 및 상기 악성 어플리케이션 분석부의 분석결과를 사용자에게 제공하는 분석결과 제공부를 포함한다.

대 표 도 - 도2



(72) 발명자 **정지환**

서동원

명 세 서

청구범위

청구항 1

스마트 기기 내 소정의 어플리케이션을 검증하는 서버에 있어서,

상기 어플리케이션에 대한 검증용 실행코드를 생성하되, 상기 어플리케이션과 관련된 정보를 무작위로 조합하여 상기 실행코드를 생성하는 실행코드 생성부;

상기 스마트 기기로부터 검증 요청신호를 수신하면 상기 생성된 실행코드 중 무작위로 선택된 실행코드를 상기 스마트 기기로 송신하고, 상기 스마트 기기에서 상기 어플리케이션에 대해 상기 선택된 실행코드를 실행시킨 결과값을 수신하는 송수신부;

상기 수신된 검증 요청신호가 최초로 수신된 것인지 판단하는 검증 요청신호 판단부;

상기 수신된 결과값을 기초로 상기 어플리케이션이 악성 어플리케이션인지 여부를 분석하는 악성 어플리케이션 분석부; 및

상기 악성 어플리케이션 분석부의 분석결과를 사용자에게 제공하는 분석결과 제공부를 포함하되,

상기 결과값은 상기 검증 요청신호 판단부의 판단 결과에 따라 상이한 루트를 통해 수신되고,

상기 분석결과 제공부는 상기 결과값이 수신된 루트를 통해 상기 분석결과를 제공하는 어플리케이션 검증 서버.

청구항 2

제 1 항에 있어서,

상기 어플리케이션과 관련된 정보를 저장하는 저장부를 더 포함하고,

상기 어플리케이션과 관련된 정보는 오리지널 어플리케이션 파일을 구성하는 특정 함수, 특정 변수, 상기 특정 변수의 해시값, 특정 문자열, 및 순서가 재배열된 문자열 중 적어도 하나 이상을 포함하는 것인, 어플리케이션 검증 서버.

청구항 3

제 1 항에 있어서,

상기 수신된 검증 요청신호가 상기 스마트 기기로부터 최초로 수신된 것이면 상기 송수신부는 상기 서버와 연결된 웹페이지를 통해 상기 결과값을 수신하고, 상기 분석결과 제공부는 상기 웹페이지를 통해 상기 분석결과를 제공하고,

상기 수신된 검증 요청신호가 상기 스마트 기기로부터 최초로 수신된 것이 아니라면 상기 송수신부는 상기 스마트 기기로부터 상기 결과값을 수신하고, 상기 분석결과 제공부는 상기 스마트 기기로 상기 분석결과를 제공하는 어플리케이션 검증 서버.

청구항 4

제 1 항에 있어서,

상기 악성 어플리케이션 분석부는

상기 서버에서 오리지널 어플리케이션에 대해 상기 선택된 실행코드를 실행시킨 오리지널 결과값과 상기 수신된 결과값을 비교하는 결과값 비교부를 포함하고,

상기 결과값 비교부의 비교 결과에 따라 상기 어플리케이션이 정상 어플리케이션인지 또는 악성 어플리케이션인지 결정하는 어플리케이션 검증 서버.

청구항 5

제 4 항에 있어서,

상기 악성 어플리케이션 분석부는

상기 선택된 실행코드를 상기 스마트 기기로 송신한 시간과 상기 결과값을 수신한 시간 간의 차이가 기준 시간 범위 이내인지 체크하는 시간 체크부를 더 포함하고,

상기 시간 체크부의 체크 결과에 따라 상기 어플리케이션이 악성 어플리케이션인지 여부에 대한 분석을 보류 또는 진행하는 어플리케이션 검증 서버.

청구항 6

서버가 스마트 기기 내 소정의 어플리케이션을 검증하는 방법에 있어서,

상기 어플리케이션에 대한 검증용 실행코드를 생성하되, 상기 어플리케이션과 관련된 정보를 무작위로 조합하여 상기 실행코드를 생성하는 단계;

상기 스마트 기기로부터 검증 요청신호를 수신하면, 상기 생성된 실행코드 중 무작위로 선택된 실행코드를 상기 스마트 기기로 송신하는 단계;

상기 수신된 검증 요청신호가 상기 스마트 기기로부터 최초로 수신된 것인지 판단하는 단계;

상기 스마트 기기에서 상기 어플리케이션에 대해 상기 선택된 실행코드를 실행시킨 결과값을 수신하되, 상기 최초로 수신된 것인지 판단하는 단계의 판단 결과에 따라 상이한 루트를 통해 수신하는 단계;

상기 수신된 결과값을 기초로 상기 어플리케이션이 악성 어플리케이션인지 여부를 분석하는 단계; 및

상기 분석하는 단계의 분석결과를 상기 결과값이 수신된 루트를 통해 사용자에게 제공하는 단계를 포함하는 어플리케이션 검증방법.

청구항 7

제 6 항에 있어서,

상기 어플리케이션과 관련된 정보는 오리지널 어플리케이션 파일을 구성하는 특정 함수, 특정 변수, 상기 특정 변수의 해시값, 특정 문자열, 및 순서가 재배열된 문자열 중 적어도 하나 이상을 포함하는 것인, 어플리케이션 검증방법.

청구항 8

제 6 항에 있어서,

상기 판단하는 단계의 판단결과가 참인 경우, 상기 수신하는 단계는 상기 서버와 연결된 웹페이지를 통해 상기 결과값을 수신하고, 상기 제공하는 단계는 상기 웹페이지를 통해 상기 분석결과를 제공하고,

상기 판단하는 단계의 판단결과가 거짓인 경우, 상기 수신하는 단계는 상기 스마트 기기로부터 상기 결과값을 수신하고, 상기 제공하는 단계는 상기 스마트 기기로 상기 분석결과를 제공하는 어플리케이션 검증방법.

청구항 9

제 6 항에 있어서,

상기 분석하는 단계는

상기 서버에서 오리지널 어플리케이션에 대해 상기 선택된 실행코드를 실행시킨 오리지널 결과값과 상기 수신된 결과값을 비교하는 단계를 포함하고,

상기 오리지널 결과값과 상기 수신된 결과값이 상이한 경우 상기 어플리케이션이 악성 어플리케이션이라고 분석하는 어플리케이션 검증방법.

청구항 10

제 9 항에 있어서,

상기 분석하는 단계는

상기 선택된 실행코드를 상기 스마트 기기로 송신한 시간과 상기 결과값을 수신한 시간 간의 차이가 기준 시간 범위 이내인지 체크하는 단계; 및

상기 시간 간의 차이가 상기 기준 시간 범위를 벗어나는 경우 상기 어플리케이션이 악성 어플리케이션인지 여부에 대한 분석을 보류하는 단계를 더 포함하는 어플리케이션 검증방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 스마트 기기 내 어플리케이션 검증 서버 및 검증방법에 관한 것으로서, 보다 구체적으로 예측 불가능한 랜덤 실행 코드를 이용한 스마트 기기 내 어플리케이션을 검증하는 서버 및 검증방법에 관한 것이다.

배 경 기 술

[0002] 최근 스마트 폰, 테블릿 PC 등과 같은 스마트 기기의 사용이 폭발적으로 증가함에 따라, 스마트 기기에 직간접적으로 설치되는 어플리케이션(Application; App)에 대한 관심도 함께 증가하고 있다. 이러한 스마트 기기 및 어플리케이션은 직관적인 조작이 가능하고 사용하는 데에 용이하고 편리하다는 긍정적인 면이 있지만, 보안 문제로 인해 외부의 공격에 노출될 가능성 및 악의적인 해커에 의해 개인정보가 유출될 가능성도 높아진다는 부정적인 면도 함께 존재한다.

[0003] 특히, 스마트 기기 내에서 사용자의 의사와 이익에 반해 시스템을 파괴하거나 정보를 유출하는 등 악의적 활동을 수행하도록 의도적으로 제작된 멀웨어(Malicious Software; Malware)가 다수 발견되고 있는 실정이다. 스마트 기기에 대한 멀웨어는 리패키징(Repackaging) 기법에 의해 설계되는 경우가 약 80% 이상을 차지하고, 스미싱(Smishing), 악성 URL 등에 의한 경우도 종종 나타나고 있다. 리패키징 기법에 의한 악성 어플리케이션(Malicious Application)이 해커 또는 악의적인 공격자에 의해 쉽게 생성될 수 있고, 이에 따라 다양한 변종이 생성될 수 있다는 문제점이 있다.

[0004] 도 1은 오리지널 어플리케이션과 악성 어플리케이션을 비교하여 설명하기 위한 도면이다.

[0005] 예를 들어, 스마트 기기 내에 설치된 오리지널 어플리케이션은 기능 1, 2, 3을 수행하지만, 악성 어플리케이션은 리패키징 기법에 의해 설계되어 기능 3 대신에 악성 기능을 수행할 수 있다. 도 1에 도시된 것처럼 오리지널 어플리케이션과 악성 어플리케이션의 실행 화면 및 사용자 인터페이스 등이 서로 상이하다는 것을 알 수 있지만, 이러한 차이점은 정교하게 조작되어 있어 일반 사용자는 발견 또는 인지하지 못할 가능성이 매우 높다.

[0006] 이와 같은 악성 어플리케이션에 대항하기 위한 모바일 보안 솔루션들에 대한 개발이 지속적으로 이루어지고 있는 실정이다. 예를 들어, 블랙 리스트 선정 및 휘슬 프로그램, 시그니쳐 및 행동에 기반을 두는 악성 어플리케이션 감지 프로그램 등이 현재 존재한다.

[0007] 한편, 이와 관련하여 한국등록특허 제10-1272026호(발명의 명칭: 해킹 방지 시스템 및 그 제어방법과, 그 시스

템에 포함되는 해킹 방지 지원 서버 및 그 제어방법)는 특정 단말기 상에서의 해킹이 쉽게 이루어지지 않도록 하고, 더 나아가 그 해킹 상태를 외부의 서버에서 쉽게 확인할 수 있도록 하는 해킹 방지시스템 및 그와 관련된 기술에 대해 개시하고 있다.

[0008] 구체적으로, 단말기가 검증 함수 요청 신호를 서버로 전송하면, 서버가 결정된 검증 함수 목록과 검증 함수 실행 순서를 포함하는 검증 함수 정보(예를 들어, 3번, 37번, 11번, 21번, 85번, 57번 검증 함수를 순서대로 실행하라는 내용)를 단말기로 전송한다. 또한, 단말기는 특정 어플리케이션과 관련된 프로그램 코드에 대한 검증 함수를 수행하고, 서버는 저장된 검증 함수 결과값과 산출된 검증 함수 결과값을 비교하여 특정 어플리케이션에 대한 해킹 여부를 판단할 수 있다.

[0009] 다만, 한국등록특허 제10-1272026호에서, 단말기 내 특정 어플리케이션 자체 또는 특정 어플리케이션에 포함되어 있던 다수의 검증 함수가 리버스 엔지니어링(Reverse Engineering)에 기반한 해킹 공격을 받는 경우, 공격자가 분석된 검증 함수를 통해 서버로 전송될 올바른 검증 값을 예측하거나 추출하여 검증 방식을 무효화 할 수 있다는 문제가 여전히 존재하였다.

발명의 내용

해결하려는 과제

[0010] 본 발명은 전술한 종래 기술의 문제점을 해결하기 위한 것으로서, 본 발명의 일부 실시예는 스마트 기기에서 무작위로 선택된 예측 불가능한 랜덤 실행코드를 실행시켜, 스마트 기기 내 소정의 어플리케이션이 악성 어플리케이션인지 여부를 검증할 수 있는 서버 및 검증방법을 제공하는 데에 그 목적이 있다.

[0011] 또한, 본 발명의 일부 실시예는 소정의 어플리케이션이 처음부터 악성 마켓으로부터 다운로드된 악성 어플리케이션인지 검증할 수 있고, 소정의 어플리케이션이 정상 마켓으로부터 다운로드되었으나 이후에 악성 어플리케이션으로 변했는지 여부를 검증할 수 있는 서버 및 검증방법을 제공하는 데에 다른 목적이 있다.

[0012] 다만, 본 실시예가 이루고자 하는 기술적 과제는 상기된 바와 같은 기술적 과제로 한정되지 않으며, 또 다른 기술적 과제들이 존재할 수 있다.

과제의 해결 수단

[0013] 상술한 기술적 과제를 달성하기 위한 기술적 수단으로서, 본 발명의 일 실시예에 따른 스마트 기기 내 소정의 어플리케이션을 검증하는 서버는, 상기 어플리케이션에 대한 검증용 실행코드를 생성하되, 상기 어플리케이션과 관련된 정보를 무작위로 조합하여 상기 실행코드를 생성하는 실행코드 생성부; 상기 스마트 기기로부터 검증 요청신호를 수신하면 상기 생성된 실행코드 중 무작위로 선택된 실행코드를 상기 스마트 기기로 송신하고, 상기 스마트 기기에서 상기 어플리케이션에 대해 상기 선택된 실행코드를 실행시킨 결과값을 수신하는 송수신부; 상기 수신된 결과값을 기초로 상기 어플리케이션이 악성 어플리케이션인지 여부를 분석하는 악성 어플리케이션 분석부; 및 상기 악성 어플리케이션 분석부의 분석결과를 사용자에게 제공하는 분석결과 제공부를 포함한다.

[0014] 또한, 본 발명의 일 실시예에 따른 서버가 스마트 기기 내 소정의 어플리케이션을 검증하는 방법은, 상기 어플리케이션에 대한 검증용 실행코드를 생성하되, 상기 어플리케이션과 관련된 정보를 무작위로 조합하여 상기 실행코드를 생성하는 단계; 상기 스마트 기기로부터 검증 요청신호를 수신하면, 상기 생성된 실행코드 중 무작위로 선택된 실행코드를 상기 스마트 기기로 송신하는 단계; 상기 스마트 기기에서 상기 어플리케이션에 대해 상기 선택된 실행코드를 실행시킨 결과값을 수신하는 단계; 상기 수신된 결과값을 기초로 상기 어플리케이션이 악성 어플리케이션인지 여부를 분석하는 단계; 및 상기 분석하는 단계의 분석결과를 사용자에게 제공하는 단계를 포함한다.

발명의 효과

[0015] 전술한 본 발명의 과제 해결 수단에 의하면, 기존에 알려진 악성 어플리케이션뿐만 아니라 알려지지 않은 변종까지 탐지할 수 있고, 예측 불가능하고 전체적인 동작 방식을 알 수 없는 랜덤 실행코드 조각을 이용함으로써,

검증 알고리즘을 거꾸로 추적 및 분석하는 리버스 엔지니어링에 기반한 공격에 대해 효과적으로 대응할 수 있다.

[0016] 또한, 본 발명에서 제안하는 스마트 기기 내 소정의 어플리케이션을 검증하는 서버를 이용하여, 소정의 어플리케이션이 변조되었는지 여부를 알아낼 수 있고, 개인정보 및 기타 정보의 유출을 미연에 방지할 수 있다.

[0017] 또한, 본 발명의 일부 실시예에서는 검증 요청신호의 횟수를 판단함으로써, 소정의 어플리케이션이 마켓에서 다운로드 될 때부터 악성 어플리케이션이었는지 혹은 마켓에서 다운로드 될 때에는 정상이었으나 이후에 변조된 것인지 구별하여 검증할 수 있다.

[0018] 또한, 본 발명에서 제안하는 스마트 기기 내 소정의 어플리케이션을 검증하는 방법은, 어플리케이션 제작 업체뿐만 아니라, 각종 은행, 금융 업계에서도 개인 및 기관의 보안을 위해 사용될 수 있다.

도면의 간단한 설명

[0019] 도 1은 오리지널 어플리케이션과 악성 어플리케이션을 비교하여 설명하기 위한 도면,

도 2는 본 발명의 일 실시예에 따른 어플리케이션 검증 서버의 각 구성을 설명하기 위한 도면,

도 3은 도 2에 도시된 실행코드 생성부를 보다 상세하게 설명하기 위한 도면,

도 4는 종래의 검증모듈을 이용한 검증방식과 본 발명에서 제안하는 랜덤 실행코드를 이용한 검증방식을 비교하여 설명하기 위한 도면,

도 5는 본 발명의 다른 실시예에 따른 어플리케이션 검증 서버의 각 구성 및 동작을 나타낸 도면,

도 6은 도 5에 도시된 ①의 동작을 보다 상세하게 설명하기 위한 흐름도,

도 7은 도 5에 도시된 ②의 동작을 보다 상세하게 설명하기 위한 흐름도,

도 8은 본 발명의 일 실시예에 따른 어플리케이션 검증방법의 각 단계를 설명하기 위한 순서도이다.

발명을 실시하기 위한 구체적인 내용

[0020] 아래에서는 첨부한 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 본 발명의 실시예를 상세히 설명한다. 그러나 본 발명은 여러 가지 상이한 형태로 구현될 수 있으며 여기에서 설명하는 실시예에 한정되지 않는다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략하였으며, 명세서 전체를 통하여 유사한 부분에 대해서는 유사한 도면 부호를 붙였다.

[0021] 명세서 전체에서, 어떤 부분이 다른 부분과 "연결"되어 있다고 할 때, 이는 "직접적으로 연결"되어 있는 경우뿐 아니라, 그 중간에 다른 소자를 사이에 두고 "전기적으로 연결"되어 있는 경우도 포함한다. 또한 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0022] 이하, 본 발명의 구체적인 실시예를 첨부한 도면을 참조하여 상세히 설명하면 다음과 같다. 다만, 본 발명의 사상은 제시되는 일 실시예에 제한되지 아니하며, 본 발명의 사상을 이해하는 동일한 사상의 범위 내에서 구성요소의 부가, 변경, 삭제, 추가 등에 의해서 다른 실시예를 쉽게 발명할 수 있을 것이나, 이 또한 본 발명의 사상의 범위 내에 포함된다고 할 것이다.

[0023] **<어플리케이션 검증 서버>**

[0024] 프로그램에 대한 검증(Attestation)은 해당 프로그램에 대한 완전성(Integrity) 체크를 이용하는 동작으로서, 센서 네트워크, 임베디드 장치 등을 위해 널리 사용되고 있다. 일반적으로 검증기(Verifier) 측에서 난수 또는 변수와 같은 챌린지 데이터(Challenge Data)를 피검증 기기 측으로 전달하면 피검증 기기 내에서 검증 알고리즘이 수행되고, 피검증 기기는 그에 대한 응답(Challenge Response)을 검증기 측으로 전달한다. 다만, 이러한 종래의 검증방법은 네트워크 공격이나 리버스 엔지니어링에 기반한 해킹 공격에 취약하고, 검증 알고리즘 자체가 조작될 수 있다.

[0025] 본 발명에서 제안하는 소정의 어플리케이션을 검증하는 서버(시스템)는 기존에 알려진 악성 어플리케이션뿐만 아니라 알려지지 않은 변종까지 탐지할 수 있고, 공격자들이 검증 알고리즘을 거꾸로 추적 및 분석하는 리버스

엔지니어링에 대해 효과적으로 대응할 수 있으며, 피검증 기기에 부담을 줄이기 위해 낮은 리소스 소모량만을 요구할 수 있다.

[0026] 도 2는 본 발명의 일 실시예에 따른 어플리케이션 검증 서버의 각 구성을 설명하기 위한 도면이다.

[0027] 스마트 기기(10) 내 소정의 어플리케이션(12)을 검증하는 서버(100)는 실행코드 생성부(110), 송수신부(120), 결과값 비교부(132)와 시간 체크부(134)를 포함한 악성 어플리케이션 분석부(130), 분석결과 제공부(140), 및 저장부(150)를 포함한다.

[0028] 피검증 기기인 스마트 기기(10)와 어플리케이션 검증 서버(100)는 네트워크(Network)를 통해 다양한 데이터를 주고 받는다.

[0029] 여기서, 스마트 기기(10)는 컴퓨터나 휴대용 단말기로 구현될 수 있다. 이때, 컴퓨터는 예를 들어, 웹 브라우저(WEB Browser)가 탑재된 노트북, 데스크톱(desktop), 랩톱(laptop), 태블릿 PC, 슬레이트 PC, 스마트 TV와 같은 가전제품, 그에 상응하는 제어 구성이 내재된 임베디드 시스템 등을 포함하고, 휴대용 단말기는 예를 들어, 휴대성과 이동성이 보장되는 무선 통신 장치로서, PCS(Personal Communication System), GSM(Global System for Mobile communications), PDC(Personal Digital Cellular), PHS(Personal Handyphone System), PDA(Personal Digital Assistant), IMT(International Mobile Telecommunication)-2000, CDMA(Code Division Multiple Access)-2000, W-CDMA(W-Code Division Multiple Access), WiBro(Wireless Broadband Internet) 단말, 스마트 폰(Smart Phone) 등과 같은 모든 종류의 핸드헬드(Handheld) 기반의 무선 통신 장치를 포함할 수 있다. 덧붙여, 스마트 기기(10)는 안드로이드(Android) 운영체제 기반일 수 있으나, 다른 운영체제가 적용될 수도 있다.

[0030] 또한, 네트워크(20)는 근거리 통신망(Local Area Network; LAN), 광역 통신망(Wide Area Network; WAN) 또는 부가가치 통신망(Value Added Network; VAN) 등과 같은 유선 네트워크나 이동 통신망(mobile radio communication network) 또는 위성 통신망 등과 같은 모든 종류의 무선 네트워크로 구현될 수 있다.

[0031] 사용자는 스마트 기기(10)를 사용하다가 소정의 어플리케이션(12)을 마켓에서 다운로드 받아, 스마트 기기(10) 내에 설치할 수 있다. 소정의 어플리케이션(12)은 스마트 기기(10)의 출고 당시 이미 설치되어 있었던 기본 어플리케이션일 수도 있다.

[0032] 실행코드 생성부(110)는 소정의 어플리케이션(12)에 대한 검증용 실행코드(Executable Code)를 생성하되, 소정의 어플리케이션(12)과 관련된 정보를 무작위로 조합하여 검증용 실행코드를 생성한다.

[0033] 참고로, 실행코드는 단순히 데이터만 담고 있는 일반적인 파일과 달리, 암호화된 명령에 따라 지시된 작업을 수행하도록 하는 컴퓨터 파일을 의미한다. 따라서, 본 발명에서는 기존 방식처럼 난수 또는 변수와 같은 첼린지 데이터를 이용하는 것이 아니라, 동작방식을 예측할 수 없는 랜덤 실행코드 조각을 이용한다는 차이가 있다.

[0034] 도 3은 도 2에 도시된 실행코드 생성부를 보다 상세하게 설명하기 위한 도면이다.

[0035] 실행코드 생성부(110)는 주요 검증모듈(112), 검증모듈 생성기(114) 및 수정 검증모듈(116)을 포함하고, 소정의 어플리케이션(12)과 관련된 정보를 저장하는 저장부(150)로부터 해당 정보를 수신하여 랜덤 실행코드를 생성할 수 있다. 여기서, 소정의 어플리케이션(12)과 관련된 정보는 오리지널(Original) 어플리케이션 파일을 구성하는 특정 함수, 특정 변수, 특정 변수의 해시(Hash)값, 특정 문자열(String), 및 순서가 재배열된 문자열 중 적어도 하나 이상을 포함할 수 있다.

[0036] 주요 검증모듈(112)은 공유된 이진 정보 및 필수검증 알고리즘을 포함할 수 있다.

[0037] 검증모듈 생성기(114)는 오리지널 어플리케이션 파일로부터 특이 정보(설치 경로, 섹션의 해시 결과, 어플리케이션 내 파일 리스트 등)를 선택하고, 해시모듈과 함께 특이 정보를 무작위로 조합하여 랜덤 실행코드를 생성할 수 있다. 이 과정에서 문자열 재배치, 키 확장, 난수 추가 등이 이루어질 수 있다.

[0038] 수정 검증모듈(116)은 공유된 이진 정보 및 검증모듈 생성기(114)에서 생성된 랜덤 실행코드를 포함할 수 있다.

[0039] 보다 구체적으로, 본 발명에서의 실행코드는 해시 함수의 결과값을 블록 단위로 나누어 재정렬하는 방식을 거쳐 생성될 수도 있다. 이는 기존의 단순히 해시 함수를 거쳐서 검증값을 생성하는 방식, 소프트웨어 난독화(Obfuscation)를 이용한 검증방식, 디지털 서명을 사용해 소프트웨어 개발자 이름을 프로그램이나 인터넷 애플릿과 연결시키는 코드 사이닝 검증방식 등과 차별화된 새로운 검증방식이다.

[0040] 해시 함수 자체를 변경해 이용하면 안티-백트래킹(Anti-backtracking) 및 스노우볼(snowball) 특성에 문제가 생길 수 있으므로, 해시 함수의 결과값을 변경하여 이용할 수 있다.

[0041] 특히, 해시 종류, 블록을 나누는 방법, 및 블록을 재정렬하는 방법을 이용하여, 해시 함수의 결과값을 변경하는 방식의 복잡도를 산출해낼 수 있다.

[0042] 이때, 해시 종류(해시 함수의 개수)를 l, 블록의 크기(개수)를 m, 검증값의 총 길이(해시 결과값의 사이즈)를 n으로 나타낸다면, 해시 함수의 결과값을 블록 단위로 나누어 블록들을 새로이 재정렬하는 방법은 $\sum_{m=1}^{n} nPm$ 과 같이 표현될 수 있고, 해시 함수의 결과값을 변경하는 방식의 복잡도 F는 다음 수학식 1처럼 표현될 수 있다.

수학식 1

[0043] $$F=(\sum_{m=1}^{n}\frac{n!}{(n-m)!})l$$

[0044] 만일 메시지 축약 알고리즘인 MD5(Message-Digest Algorithm 5)의 해시를 이용한다면, 32글자를 기준으로 하기 때문에 검증값의 총 길이가 32이다.

수학식 2

[0045] $$\sum_{m=1}^{32}\frac{32!}{(32-m)!}=715\,263\,772\,544\,079\,320\,945\,495\,293\,616\,151\,424$$

[0046] 이와 같이, 실행코드 생성부(110)는 약 10^35 경우의 수를 가진 실행코드(예측 불가능한 복잡도의 랜덤화된 결과값)를 생성할 수 있다.

[0047] 다시 도 2를 참고하면, 송수신부(120)는 스마트 기기(10)로부터 검증 요청신호를 수신하면 실행코드 생성부(110)에서 생성된 실행코드 중 무작위로 선택된 실행코드를 해당 스마트 기기(10)로 송신한다. 또한, 송수신부(120)는 해당 스마트 기기(10)에서 소정의 어플리케이션(12)에 대해 상기 선택된 실행코드를 실행시킨 결과값을 수신한다.

[0048] 악성 어플리케이션 분석부(130)는 수신된 결과값을 기초로 소정의 어플리케이션이 악성 어플리케이션인지 여부를 분석한다. 여기서, 악성 어플리케이션이란 오리지널 어플리케이션 파일 또는 오리지널 어플리케이션의 기능과 관련하여 일부가 수정, 누락, 추가, 혹은 변경된 것을 의미할 수 있다.

[0049] 구체적으로, 악성 어플리케이션 분석부(130)는 결과값 비교부(132) 및 시간 체크부(134)를 포함한다.

[0050] 결과값 비교부(132)는 검증 서버(100)에서 오리지널 어플리케이션에 대해 선택된 실행코드를 실행시킨 오리지널 결과값(예측된 결과값)과, 상술한 수신된 결과값을 서로 비교한다. 악성 어플리케이션 분석부(130)는 결과값 비교부(132)의 비교 결과에 따라 소정의 어플리케이션(12)이 정상 어플리케이션인지 또는 악성 어플리케이션인지 결정할 수 있다.

[0051] 또한, 시간 체크부(134)는 선택된 실행코드를 스마트 기기(10)로 송신한 시간과, 상술한 결과값을 수신한 시간 간의 차이가 기준 시간 범위 이내인지 체크한다. 악성 어플리케이션 분석부(130)는 시간 체크부(134)의 체크 결과에 따라 소정의 어플리케이션(12)이 악성 어플리케이션인지 여부에 대한 분석을 보류 또는 진행할 수 있다.

[0052] 분석결과 제공부(140)는 상술한 악성 어플리케이션 분석부(130)의 분석결과를 사용자에게 제공하고, 제공방식은 특별히 제한되지 않는다.

[0053] 아울러, 도 4는 종래의 검증모듈을 이용한 검증방식과 본 발명에서 제안하는 랜덤 실행코드를 이용한 검증방식을 비교하여 설명하기 위한 도면이다.

[0054] 도 4(a)에 도시된 종래의 검증방식은 스마트 기기(10)에 내장된 검증모듈(14)을 이용한다. 검증모듈(14)은 검

증기(30)로부터 수신된 임의의 값(nonce)을 활용하여 스마트 기기(10) 내 소정의 어플리케이션(12)을 검증하고, 결과값을 획득한다.

[0055] 반면에, 도 4(b)에 도시된 본 발명에서 제안하는 검증방식은 검증 서버(100)에 의해 무작위로 선택된 랜덤 실행코드(A)를 이용한다. 스마트 기기(10) 내 소정의 어플리케이션(12)에 대해 실행코드(A)가 실행되고, 그에 따른 결과값이 획득된다.

[0056] 예를 들어, 실행코드(A)는 해시된 제 1 출력, 오리지널 어플리케이션 파일을 구성하는 A 부분, 임의의 값, 재배치된 문자열, 해시된 제 2 출력, 오리지널 어플리케이션 파일을 구성하는 B 부분 등의 무작위 조합에 의해 생성된 것일 수 있다.

[0057] 따라서, 스마트 기기(10)에서 소정의 어플리케이션(12)에 대한 검증 작업이 수행될 때마다 실행코드(A)가 달라지고, 시그니쳐를 위한 저장 및 스캔 동작이 필요하지 않아, 외부 공격자가 해당 검증 작업을 분석해내거나 리버스 엔지니어링에 기반한 악의적인 공격을 수행하기 어렵다.

[0058] 한편, 도 5는 본 발명의 다른 실시예에 따른 어플리케이션 검증 서버의 각 구성 및 동작을 나타낸 도면이다.

[0059] 소정의 어플리케이션(12)이 스마트 기기(10)에 사용자(40)에 의해 설치되거나 출고시에 미리 설치되어 있다(S11).

[0060] 스마트 기기(10)는 검증 서버(100)로 소정의 어플리케이션(12)에 대한 검증 요청신호를 송신하고(S12), 검증 서버(100)는 상술한 실행코드 생성부(110)에서 생성된 랜덤 실행코드 중 무작위로 선택된 실행코드(A)를 스마트 기기(10)로 송신한다(S13).

[0061] 스마트 기기(10)에서 소정의 어플리케이션에 대한 일종의 검증 작업이 선택된 실행코드(A)에 의해 실행되고, 검증 작업의 결과값은 검증 서버(100)로 사용자(40)에 의해 간접적으로 전달되거나 스마트 기기(10)에 의해 직접적으로 전달될 수 있다.

[0062] 이때, 검증 작업의 결과값이 전달되는 과정은 검증 요청신호 판단부(160)의 판단결과에 따라 달라질 수 있다. 본 발명의 다른 실시예에 따른 어플리케이션 검증 서버(100)는 도 5에 도시된 것처럼 상술한 구성들 이외에 검증 요청신호 판단부(160)를 더 포함한다.

[0063] 검증 요청신호 판단부(160)는 앞서 검증 서버(100)로 수신된 검증 요청신호가 최초로 수신된 것인지 판단한다. 즉, 검증 요청신호 판단부(160)는 수신된 검증 요청신호의 횟수를 판단할 수 있다.

[0064] 일 예로, 수신된 검증 요청신호가 스마트 기기(10)로부터 최초로 수신된 것이면, ①에 도시된 것처럼 송수신부(120)는 검증 서버(100)와 연결된 웹페이지(50) 또는 문자메시지(SMS)를 통해 검증 작업의 결과값을 수신하고(S15), 분석결과 제공부(140)는 위 웹페이지(50) 또는 문자메시지를 통해 악성 어플리케이션 분석부(130)의 분석결과를 제공한다(S16).

[0065] 이때, 검증 작업의 결과값은 스마트 기기(10)를 사용하는 사용자(40)에 의해 확인된 후(S14), 사용자(40)에 의해 웹페이지(50) 또는 문자메시지를 통해 입력될 수 있다(S15). 이를 위해, 실행코드 생성부(110)에서 생성되는 각각의 실행코드는 ①에 도시된 것과 같은 결과값 전달 경로에 대한 정보를 포함할 수 있고, 웹페이지(50) 또는 문자메시지 방식 이외의 다른 방식이 적용될 수도 있다.

[0066] 상술한 방식을 이용하여 검증 서버(100)는 소정의 어플리케이션이 마켓에서 다운로드 될 때부터 악성 어플리케이션이었는지 검증할 수 있다.

[0067] 다른 예로, 수신된 검증 요청신호가 스마트 기기(10)로부터 최초로 수신된 것이 아니라면, ②에 도시된 것처럼 송수신부(120)는 스마트 기기(10)로부터 검증 작업의 결과값을 수신하고(S17), 분석결과 제공부(140)는 스마트 기기(10)로 악성 어플리케이션 분석부(130)의 분석결과를 제공한다(S18).

[0068] 이를 위해, 실행코드 생성부(110)에서 생성되는 각각의 실행코드는 ②에 도시된 것과 같은 결과값 전달 경로에 대한 정보를 포함할 수 있다.

[0069] 상술한 방식을 이용하여 검증 서버(100)는 소정의 어플리케이션이 마켓에서 다운로드 될 때에는 정상이었으나 이후에 변조된 것인지 검증할 수 있다.

[0070] 이하에서는 도 6 및 도 7을 참고하여 상술한 내용을 좀더 구체적으로 설명하기로 한다.

[0071] 도 6은 도 5에 도시된 ①의 동작을 보다 상세하게 설명하기 위한 흐름도이다.

[0072] 사용자(40)는 소정의 어플리케이션(12)을 자신의 스마트 기기(10)에 설치하고, 설치 당시에 사용자 식별정보(ID)를 입력할 수 있다(S20).

[0073] 스마트 기기(10)는 소정의 어플리케이션(12)에 대한 검증 작업을 위해 검증 요청신호 및 사용자 식별정보를 검증 서버(100)로 전달한다(S21).

[0074] 검증 서버(100)는 적어도 하나 이상의 랜덤 실행코드를 생성하는데(S22), 생성하는 타이밍은 검증 요청신호를 수신하기 전 또는 후일 수 있다.

[0075] 또한, 스마트 기기로부터 검증 요청신호를 수신하면, 검증 서버(100)는 소정의 어플리케이션(12)이 설치되어 있는 다수의 스마트 기기(A, B, C, ...) 중 어떤 스마트 기기로부터 검증 요청신호를 수신하였는지 사용자 식별정보를 이용하여 판단하고(S23), 해당 스마트 기기(10)로부터 수신된 검증 요청신호의 횟수가 1회인지 체크한다(S24).

[0076] 이후 검증 서버(100)는 생성된 실행코드 중 무작위로 실행코드를 선택하고(S25), 식별된 스마트 기기(10)로 선택된 랜덤 실행코드와 결과값 전달 경로에 대한 정보를 송신한다(S26). 이때, 해당 스마트 기기(10)로 선택된 랜덤 실행코드를 송신한 시간(t_1)이 검증 서버(100) 내에 기록된다.

[0077] 스마트 기기(10)는 수신된 실행코드를 이용하여 소정의 어플리케이션(12)에 대한 검증 작업을 수행하고(S27), 검증 작업의 결과값을 획득한다(S28). 또한, 스마트 기기(10)는 결과값 전달 경로에 대한 정보에 기초하여 검증 작업의 결과값을 사용자(40)에게 표시한다(S29).

[0078] 사용자는 검증 서버(100)와 연결된 웹페이지(50)에 접속하여 검증 작업의 결과값을 입력할 수 있다(S30). 이때, 검증 서버(100)는 검증 작업의 결과값을 수신한 시간 (t_2)을 기록한다.

[0079] 검증 서버(100)는 상기 과정(S27 내지 S30)이 이루어지고 있는 도중에, 기저장되어 있는 오리지널 어플리케이션에 대해 스마트 기기(10)로 전달한 랜덤 실행코드를 실행시키고(S31), 오리지널 결과값을 획득한다(S32).

[0080] 또한, 검증 서버(100)는 선택된 실행코드를 스마트 기기(10)로 송신한 시간(t_1)과, 검증 작업의 결과값을 수신한 시간(t_2) 간의 차이가 기준 시간 범위(t_x) 이내인지 체크하고(S33), 기준 시간 범위(t_x) 이내이면 소정의 어플리케이션(12)이 악성 어플리케이션인지 여부에 대한 분석을 진행한다.

[0081] 분석 작업은 수신된 결과값을 기초로 이루어지되, 검증 서버(100)는 오리지널 결과값(예측된 결과값)과 수신된 결과값을 서로 비교하여(S34), 소정의 어플리케이션(12)이 정상 어플리케이션인지 또는 악성 어플리케이션인지 결정한다.

[0082] 또한, 검증 서버(100)는 위 웹페이지(50)를 통해 분석 작업의 분석결과를 사용자에게 제공한다(S35). 이때, 소정의 어플리케이션(12)이 악성 어플리케이션인 경우에만 분석결과를 사용자에게 제공할 수 있고, 소정의 어플리케이션(12)이 정상 어플리케이션인 경우에는 추가적인 동작을 수행하지 않을 수도 있다.

[0083] 한편, 도 7은 도 5에 도시된 ②의 동작을 보다 상세하게 설명하기 위한 흐름도이다.

[0084] 스마트 기기(10)는 소정의 어플리케이션(12)에 대한 검증 작업을 위해 검증 요청신호 및 사용자 식별정보를 검증 서버(100)로 전달한다(S40). 사용자 식별정보는 소정의 어플리케이션(12)가 스마트 기기(10)에 설치될 당시 입력된 것일 수 있다.

[0085] 검증 서버(100)는 적어도 하나 이상의 랜덤 실행코드를 생성하는데(S41), 생성하는 타이밍은 검증 요청신호를 수신하기 전 또는 후일 수 있다.

[0086] 또한, 스마트 기기로부터 검증 요청신호를 수신하면, 검증 서버(100)는 소정의 어플리케이션(12)이 설치되어 있는 다수의 스마트 기기(A, B, C, ...) 중 어떤 스마트 기기로부터 검증 요청신호를 수신하였는지 사용자 식별정보를 이용하여 판단하고(S42), 해당 스마트 기기(10)로부터 수신된 검증 요청신호의 횟수가 1회 초과인지 체크한다(S43).

[0087] 이후 검증 서버(100)는 생성된 실행코드 중 무작위로 실행코드를 선택하고(S44), 식별된 스마트 기기(10)로 선택된 랜덤 실행코드와 결과값 전달 경로에 대한 정보를 송신한다(S45). 이때, 해당 스마트 기기(10)로 선택된

랜덤 실행코드를 송신한 시간(t_1)이 검증 서버(100) 내에 기록된다.

[0088] 스마트 기기(10)는 수신된 실행코드를 이용하여 소정의 어플리케이션(12)에 대한 검증 작업을 수행하고(S46), 검증 작업의 결과값을 획득한다(S47). 또한, 스마트 기기(10)는 결과값 전달 경로에 대한 정보에 기초하여 검증 작업의 결과값을 검증 서버(100)로 전달한다(S48). 이때, 검증 서버(100)는 검증 작업의 결과값을 수신한 시간 (t_2)을 기록한다.

[0089] 검증 서버(100)는 상기 과정(S46 내지 S48)이 이루어지고 있는 도중에, 기저장되어 있는 오리지널 어플리케이션에 대해 스마트 기기(10)로 전달한 랜덤 실행코드를 실행시키고(S49), 오리지널 결과값을 획득한다(S50).

[0090] 또한, 검증 서버(100)는 선택된 실행코드를 스마트 기기(10)로 송신한 시간(t_1)과, 검증 작업의 결과값을 수신한 시간(t_2) 간의 차이가 기준 시간 범위(t_x) 이내인지 체크하고(S51), 기준 시간 범위(t_x) 이내이면 소정의 어플리케이션(12)이 악성 어플리케이션인지 여부에 대한 분석을 진행한다. 기준 시간 범위(t_x)를 벗어난 경우 소정의 어플리케이션(12)이 악성 어플리케이션인지 여부에 대한 분석을 보류하거나 위 과정을 다시 수행하라는 요청을 스마트 기기(10)로 전달할 수 있다.

[0091] 분석 작업은 수신된 결과값을 기초로 이루어지되, 검증 서버(100)는 오리지널 결과값(예측된 결과값)과 수신된 결과값을 서로 비교하여(S52), 소정의 어플리케이션(12)이 정상 어플리케이션인지 또는 악성 어플리케이션인지 결정한다.

[0092] 또한, 검증 서버(100)는 스마트 기기(10)로 분석 작업의 분석결과를 전달하여, 분석결과를 사용자에게 제공한다(S53). 이때, 소정의 어플리케이션(12)이 악성 어플리케이션인 경우에만 분석결과를 사용자에게 제공할 수 있고, 소정의 어플리케이션(12)이 정상 어플리케이션인 경우에는 추가적인 동작을 수행하지 않을 수도 있다.

[0093] 지금까지 설명한 스마트 기기 내 소정의 어플리케이션을 검증하는 서버를 이용하면, 빠르게 정확하게 소정의 어플리케이션의 변조 여부를 판단할 수 있고, 리버스 엔지니어링 등의 공격에 대해 효과적으로 대응할 수 있다.

[0094] **<어플리케이션 검증방법>**

[0095] 한편, 상술한 어플리케이션 검증 서버(100)가 스마트 기기(10) 내 소정의 어플리케이션(12)을 검증하는 방법에 대해 도 8을 참고하여 설명하기로 한다. 참고로, 설명의 편의를 위해 도 2 및 도 5에 도시된 각 구성의 식별번호를 인용한다.

[0096] 도 8은 본 발명의 일 실시예에 따른 어플리케이션 검증방법의 각 단계를 설명하기 위한 순서도이다.

[0097] 우선, 검증 서버(100)는 소정의 어플리케이션(12)에 대한 검증용 실행코드를 생성하되, 소정의 어플리케이션과 관련된 정보를 무작위로 조합하여 랜덤 실행코드를 생성한다(S110).

[0098] 이때, 소정의 어플리케이션과 관련된 정보는 오리지널 어플리케이션 파일을 구성하는 특정 함수, 특정 변수, 상기 특정 변수의 해시값, 특정 문자열, 및 순서가 재배열된 문자열 중 적어도 하나 이상을 포함하는 것일 수 있다.

[0099] 이어서, 스마트 기기(10)로부터 검증 요청신호를 수신하면, 검증 서버(100)는 생성하는 단계(S110)에서 생성된 실행코드 중 무작위로 선택된 실행코드를 스마트 기기(10)로 송신한다(S120).

[0100] 계속해서, 검증 서버(100)는 스마트 기기(10)에서 소정의 어플리케이션(12)에 대해 선택된 실행코드를 실행시킨 결과값을 수신한다(S130).

[0101] 또한, 검증 서버(100)는 수신하는 단계(S130)에서 수신된 결과값을 기초로 소정의 어플리케이션(12)이 악성 어플리케이션인지 여부를 분석한다(S140).

[0102] 구체적으로, 검증 서버(100)는, 수신하는 단계(S130)에서 수신된 결과값과 검증 서버(100)에서 오리지널 어플리케이션에 대해 선택된 실행코드를 실행시킨 오리지널 결과값을 비교할 수 있다. 만약 오리지널 결과값과 수신된 결과값이 상이한 경우 검증 서버(100)는 소정의 어플리케이션(12)이 악성 어플리케이션이라고 분석하고, 오리지널 결과값과 수신된 결과값이 동일한 경우 검증 서버(100)는 소정의 어플리케이션(12)이 정상 어플리케이션이라고 분석할 수 있다.

[0103] 추가적으로, 검증 서버(100)는 무작위로 선택된 실행코드를 스마트 기기(100)로 송신한 시간과, 검증 작업의 결

과값을 수신한 시간 간의 차이가 기준 시간 범위 이내인지 체크할 수 있다. 만약 시간 간의 차이가 기준 시간 범위를 벗어나는 경우 검증 서버(100)는 소정의 어플리케이션(12)이 악성 어플리케이션인지 여부에 대한 분석을 보류하고, 시간 간의 차이가 기준 시간 범위 이내인 경우 검증 서버(100)는 소정의 어플리케이션(12)이 악성 어플리케이션인지 여부에 대한 분석을 진행할 수 있다.

[0104] 이후에, 검증 서버(100)는 분석하는 단계(S140)의 분석결과를 사용자에게 제공한다(S150).

[0105] 아울러, 본 발명의 다른 실시예에 따른 어플리케이션 검증방법의 경우, 수신하는 단계(S130)에서 수신된 검증 요청신호가 스마트 기기(10)로부터 최초로 수신된 것인지 판단하는 과정이 더 포함될 수 있다.

[0106] 만약 판단결과가 참인 경우(검증 요청신호의 수신 횟수=1회), 수신하는 단계(S130)에서 검증 서버(100)는 검증 서버(100)와 연결된 웹페이지(50)를 통해 검증 작업의 결과값을 수신하고, 제공하는 단계(S150)에서 검증 서버(100)는 그 웹페이지(50)를 통해 분석결과를 사용자에게 제공할 수 있다.

[0107] 만약 판단결과가 거짓인 경우(검증 요청신호의 수신 횟수≠1회), 수신하는 단계(S130)에서 검증 서버(100)는 스마트 기기(10)로부터 검증 작업의 결과값을 수신하고, 제공하는 단계(S150)에서 검증 서버(100)는 스마트 기기(10)로 분석결과를 사용자에게 제공할 수 있다.

[0108] 이때, 소정의 어플리케이션(12)이 악성 어플리케이션인 경우에만 분석결과를 사용자에게 제공할 수 있고, 소정의 어플리케이션(12)이 정상 어플리케이션인 경우에는 추가적인 동작을 수행하지 않을 수도 있다.

[0109] 나아가, 본 발명에서 제안하는 기술을 실제로 적용시킨 결과, 스마트 기기의 경우 랜덤 실행코드를 이용하여 검증 작업을 수행하는 데에 평균 0.49초가 걸렸고, 검증 서버의 경우 오리지널 어플리케이션에 대해 선택된 실행코드를 실행시키는 데에 평균 0.13초가 걸렸다.

[0110] 이를 통해 피검증기기인 스마트 기기와 검증 서버가 필요로 하는 리소스도 매우 낮은 수준이라는 것을 확인할 수 있었다. 또한, 사용자는 이와 같은 어플리케이션 검증방법을 이용하여 빠르고 정확하게 소정의 어플리케이션의 변조 여부를 판단할 수 있고, 리버스 엔지니어링 등의 공격에 대해 효과적으로 대응할 수 있다.

[0111] 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. 그러므로 이상에서 기술한 실시예들은 모든 면에서 예시적인 것이며 한정적이 아닌 것으로 이해해야만 한다. 예를 들어, 단일형으로 설명되어 있는 각 구성 요소는 분산되어 실시될 수도 있으며, 마찬가지로 분산된 것으로 설명되어 있는 구성 요소들도 결합된 형태로 실시될 수 있다.

[0112] 본 발명의 범위는 상기 상세한 설명보다는 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.

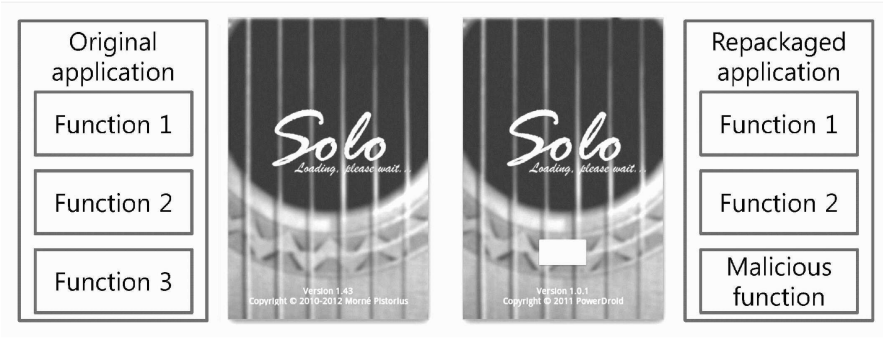
부호의 설명

[0113]

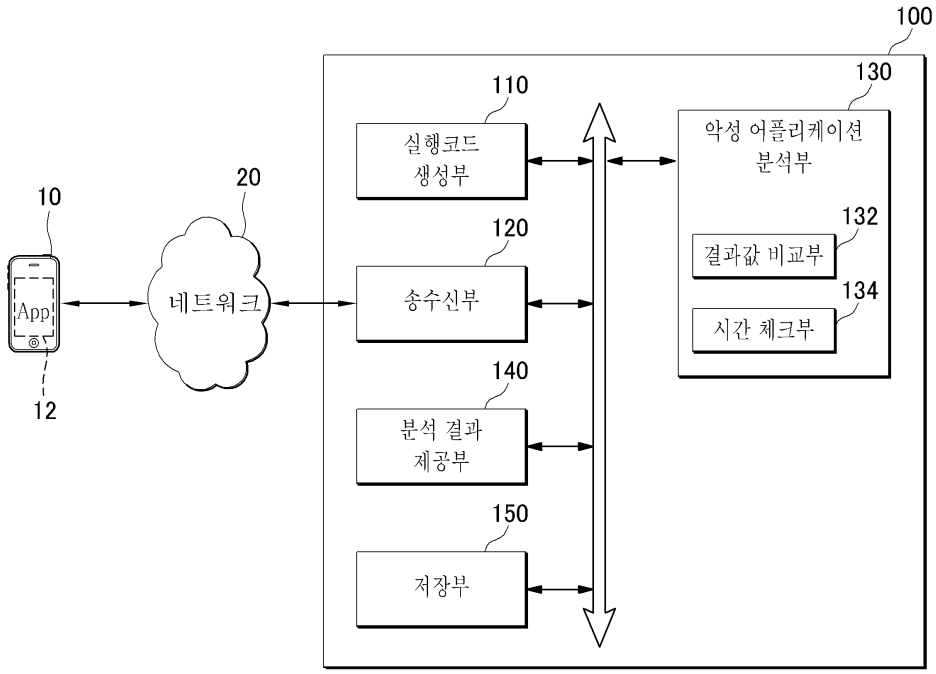
10: 스마트 기기
12: 소정의 어플리케이션(App)
20: 네트워크
100: 어플리케이션 검증 서버
110: 실행코드 생성부
120: 송수신부
130: 악성 어플리케이션 분석부
132: 결과값 비교부
134: 시간 체크부
140: 분석결과 제공부
150: 저장부

도면

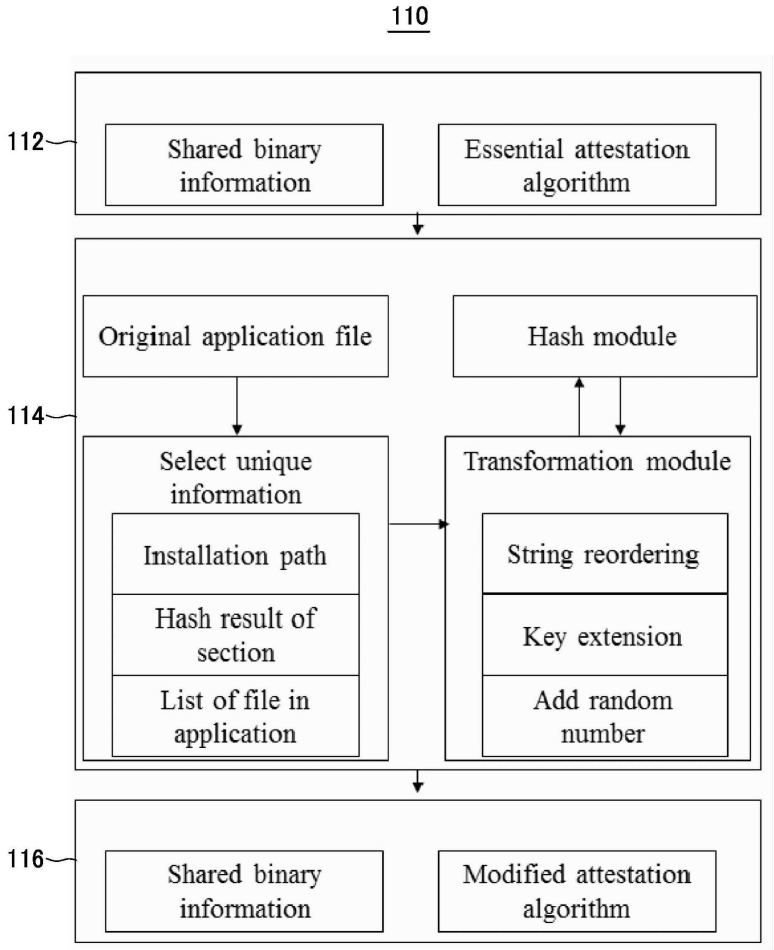
도면1


Original application
Function 1
Function 2
Function 3
Solo
Loading, please wait...
Version 1.43
Copyright © 2010-2012 Morné Pistorius
Solo
Loading, please wait...
Version 1.0.1
Copyright © 2011 PowerDroid
Repackaged application
Function 1
Function 2
Malicious function

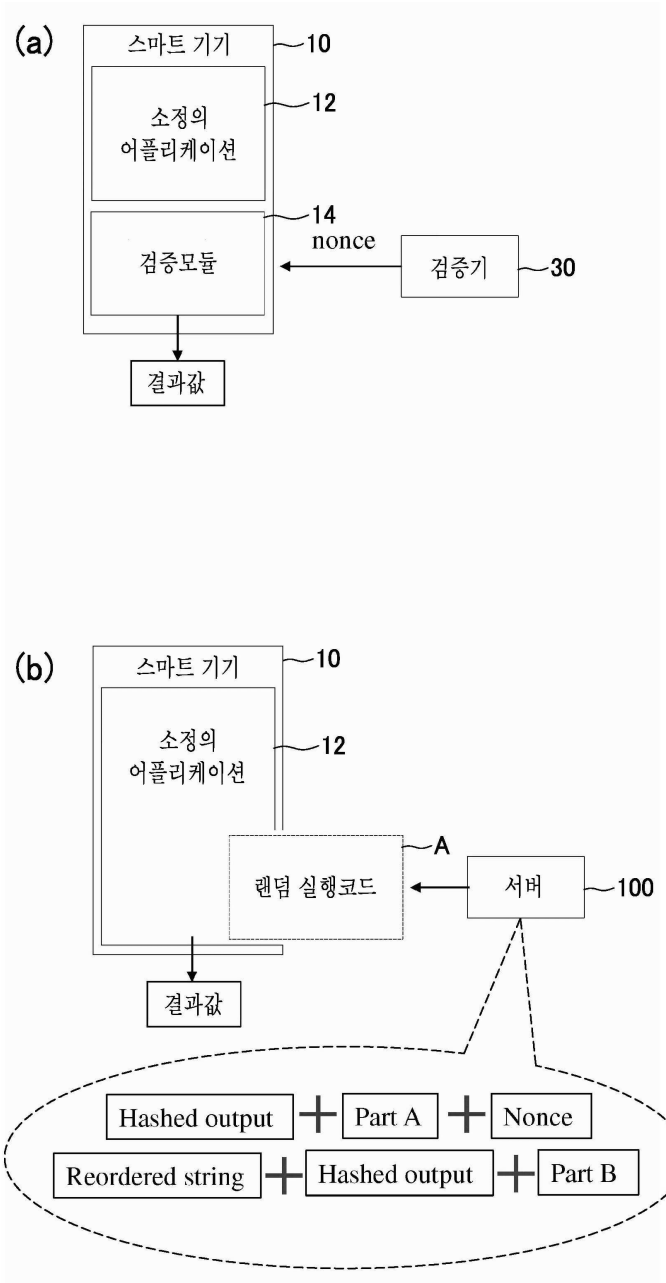
도면2


100
110
실행코드 생성부
120
송수신부
140
분석 결과 제공부
150
저장부
130
악성 어플리케이션 분석부
132
결과값 비교부
134
시간 체크부
10
App
12
20
네트워크

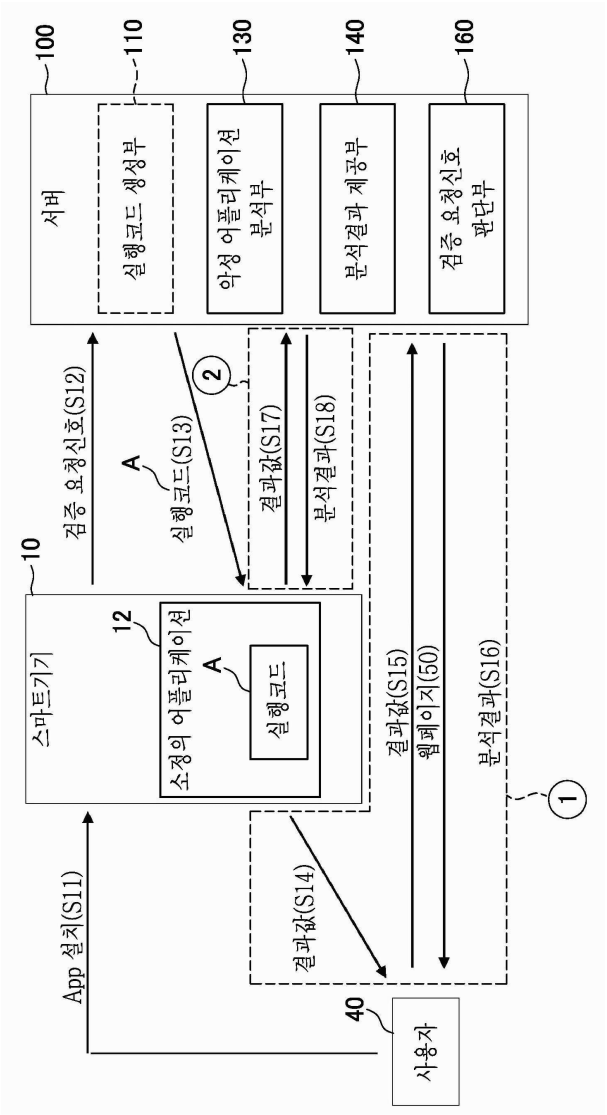
도면3


110
112
Shared binary information
Essential attestation algorithm
114
Original application file
Hash module
Select unique information
Installation path
Hash result of section
List of file in application
Transformation module
String reordering
Key extension
Add random number
116
Shared binary information
Modified attestation algorithm

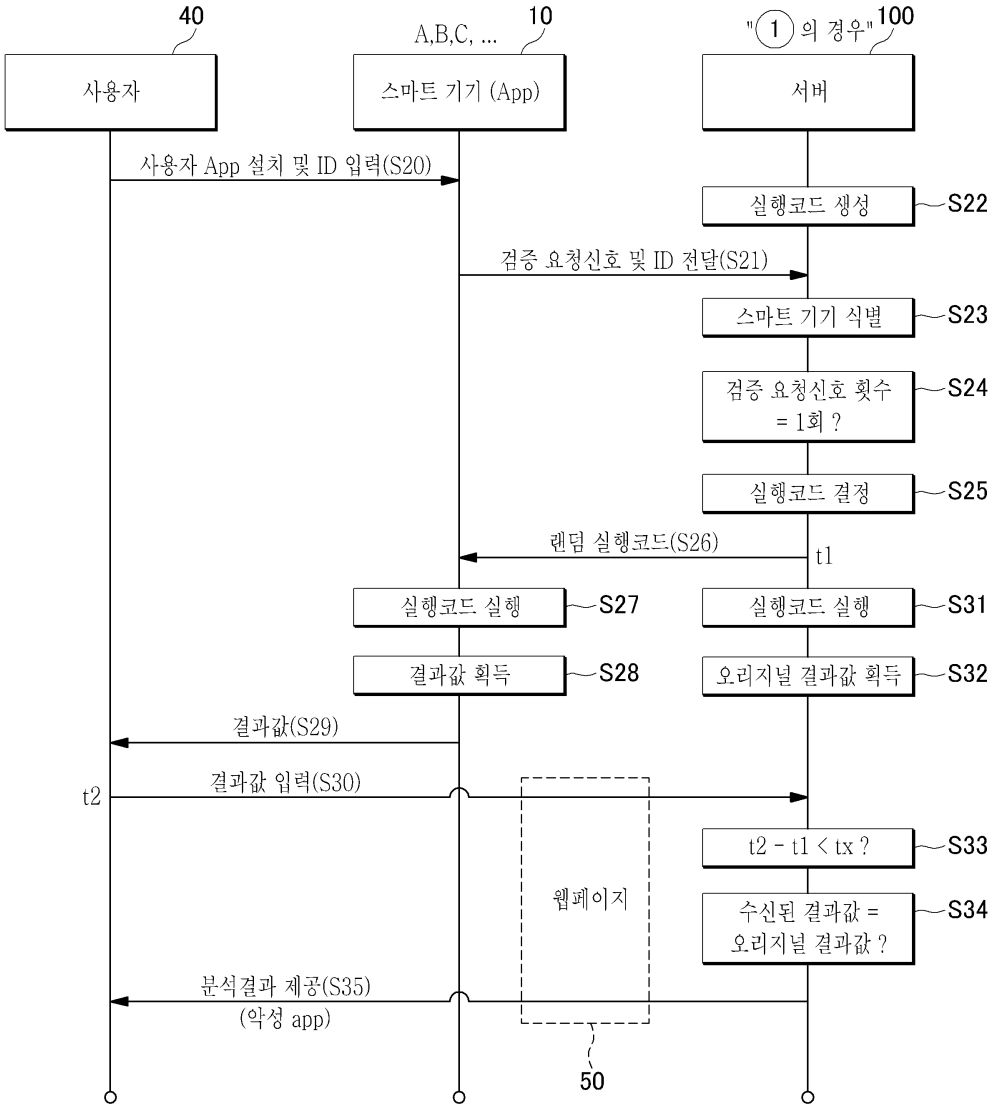
도면4


(a)
스마트 기기
10
소정의
어플리케이션
12
14
nonce
검증모듈
검증기
30
결과값
(b)
스마트 기기
10
소정의
어플리케이션
12
A
랜덤 실행코드
서버
100
결과값
Hashed output
Part A
Nonce
Reordered string
Hashed output
Part B

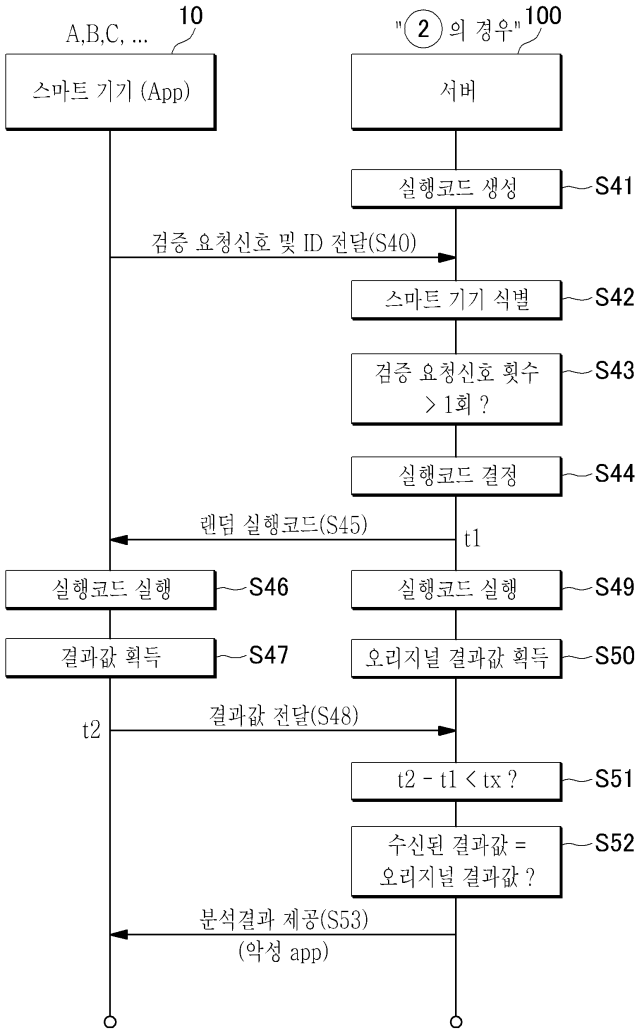
도면5


100
110
130
140
160
서버
실행코드 생성부
악성 어플리케이션 분석부
분석결과 제공부
검증 요청신호 판단부
검증 요청신호(S12)
A
실행코드(S13)
2
결과값(S17)
분석결과(S18)
10
스마트기기
12
소정의 어플리케이션
A
실행코드
결과값(S15)
웹페이지(50)
분석결과(S16)
1
App 설치(S11)
결과값(S14)
40
사용자

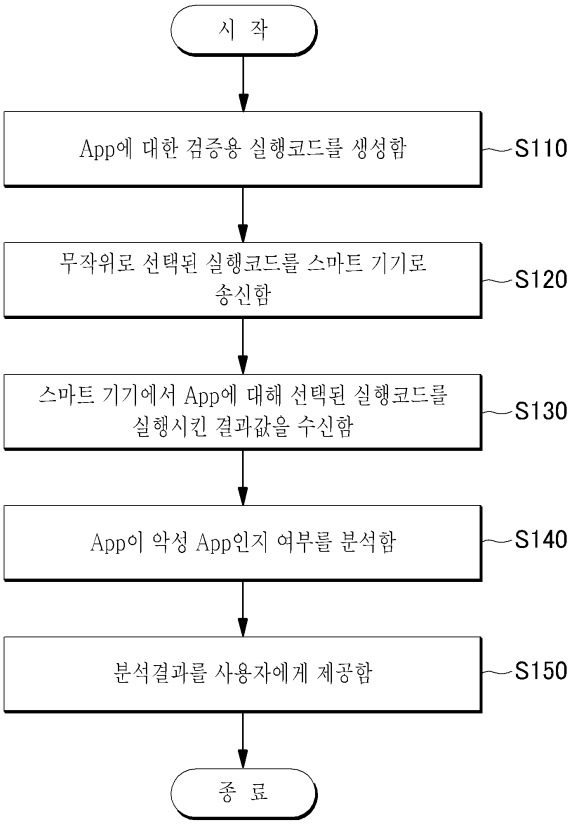
도면6


40
사용자
A,B,C, ...
10
스마트 기기 (App)
"① 의 경우"
100
서버
사용자 App 설치 및 ID 입력(S20)
실행코드 생성
S22
검증 요청신호 및 ID 전달(S21)
스마트 기기 식별
S23
검증 요청신호 횟수 = 1회 ?
S24
실행코드 결정
S25
랜덤 실행코드(S26)
t1
실행코드 실행
S27
실행코드 실행
S31
결과값 획득
S28
오리지널 결과값 획득
S32
결과값(S29)
t2
결과값 입력(S30)
t2 - t1 < tx ?
S33
웹페이지
수신된 결과값 = 오리지널 결과값 ?
S34
분석결과 제공(S35)
(악성 app)
50

도면7


A,B,C, ...
10
"2 의 경우"
100
스마트 기기 (App)
서버
실행코드 생성
S41
검증 요청신호 및 ID 전달(S40)
스마트 기기 식별
S42
검증 요청신호 횟수
> 1회 ?
S43
실행코드 결정
S44
랜덤 실행코드(S45)
t1
실행코드 실행
S46
실행코드 실행
S49
결과값 획득
S47
오리지널 결과값 획득
S50
t2
결과값 전달(S48)
t2 - t1 < tx ?
S51
수신된 결과값 =
오리지널 결과값 ?
S52
분석결과 제공(S53)
(악성 app)

도면8


시 작
App에 대한 검증용 실행코드를 생성함
S110
무작위로 선택된 실행코드를 스마트 기기로 송신함
S120
스마트 기기에서 App에 대해 선택된 실행코드를 실행시킨 결과값을 수신함
S130
App이 악성 App인지 여부를 분석함
S140
분석결과를 사용자에게 제공함
S150
종 료