(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2019-0046022
(43) 공개일자 2019년05월07일

(51) 국제특허분류(Int. Cl.)
G06F 21/51 (2013.01) *G06F 21/31* (2013.01)
G06F 21/53 (2013.01) *H04L 9/08* (2006.01)
(52) CPC특허분류
G06F 21/51 (2013.01)
G06F 21/12 (2013.01)
(21) 출원번호 10-2017-0139090
(22) 출원일자 2017년10월25일
심사청구일자 2017년10월25일

(71) 출원인
이화여자대학교 산학협력단
서울특별시 서대문구 이화여대길 52 (대현동, 이화여자대학교)
(72) 발명자
유서희
서울특별시 중구 서소문로6길 34, 505호(중림동, 성요셉아파트)
도인실
인천광역시 연수구 센트럴로 194, 201동 2705호(송도동, 더샵 센트럴파크2)
채기준
서울특별시 강남구 강남대로136길 34, 402호
(74) 대리인
특허법인(유한)아이시스

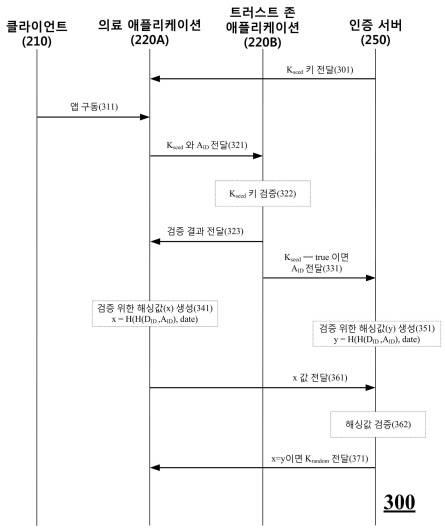
전체 청구항 수 : 총 12 항

(54) 발명의 명칭 **의료 엔터프라이즈 애플리케이션에 대한 검증 방법 및 시스템**

(57) 요 약

의료 엔터프라이즈 애플리케이션에 대한 검증 방법은 단말이 인증 서버로부터 시드키를 수신하는 단계, 상기 단말이 설치된 애플리케이션 실행 명령을 확인하면 상기 시드키를 상기 단말의 TEE(Trusted Execution Environment)에서 검증하는 단계, 상기 단말이 상기 시드키에 대한 검증이 성공하면 상기 애플리케이션의 식별자 및 상기 단말의 식별자 중 적어도 하나의 식별자를 상기 인증 서버에 전달하는 단계 및 상기 단말이 상기 인증 서버로부터 상기 적어도 하나의 식별자를 해싱한 값을 이용하여 상기 애플리케이션의 무결성을 확인한 결과를 수신하는 단계를 포함한다.

대 표 도 - 도3



(52) CPC특허분류
G06F 21/31 (2013.01)
G06F 21/53 (2013.01)
H04L 9/0877 (2013.01)
G06F 2221/2149 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 2016R1A2B4015899
부처명 과학기술정보통신부
연구관리전문기관 한국연구재단
연구사업명 기초연구사업(학술진흥)-중견연구자지원사업
연구과제명 CPS 환경에서의 신뢰성있는 서비스 제공을 위한 보안기법 및 공격 대응 방안
기 여 율 1/1
주관기관 이화여자대학교 산학협력단
연구기간 2016.06.01 ~ 2019.05.31

명 세 서

청구범위

청구항 1

단말이 인증 서버로부터 시드키를 수신하는 단계;

상기 단말이 설치된 애플리케이션 실행 명령을 확인하면 상기 시드키를 상기 단말의 TEE(Trusted Execution Environment)에서 검증하는 단계;

상기 단말이 상기 시드키에 대한 검증이 성공하면 상기 애플리케이션의 식별자 및 상기 단말의 식별자 중 적어도 하나의 식별자를 상기 인증 서버에 전달하는 단계; 및

상기 단말이 상기 인증 서버로부터 상기 적어도 하나의 식별자를 해싱한 값을 이용하여 상기 애플리케이션의 무결성을 확인한 결과를 수신하는 단계를 포함하는 의료 엔터프라이즈 애플리케이션에 대한 검증 방법.

청구항 2

제1항에 있어서,

상기 단말은 상기 TEE에서 상기 애플리케이션의 식별자와 상기 시드키를 비교하여 상기 시드키를 검증하는 의료 엔터프라이즈 애플리케이션에 대한 검증 방법.

청구항 3

제1항에 있어서,

상기 단말은 상기 적어도 하나의 식별자를 해싱한 제1 값을 상기 인증 서버에 전달하는 단계를 더 포함하고,

상기 인증 서버는 상기 단말과 동일한 방식으로 상기 적어도 하나의 식별자를 해싱한 제2 값을 생성하고, 상기 제1 값과 상기 제2 값을 비교하여 상기 애플리케이션에 대한 인증을 수행하는 의료 엔터프라이즈 애플리케이션에 대한 검증 방법.

청구항 4

제3항에 있어서,

상기 제1 값은 상기 적어도 하나의 식별자를 해싱한 값을 해싱한 시점의 날짜로 다시 해싱한 값인 의료 엔터프라이즈 애플리케이션에 대한 검증 방법.

청구항 5

제4항에 있어서,

상기 애플리케이션에 대한 무결성이 확인되면 상기 단말은 상기 적어도 하나의 식별자를 해싱한 값을 저장하고,

상기 애플리케이션에 대한 검증이 요청된 경우 상기 단말은 검증 시점의 날짜로 상기 저장한 값을 다시 해싱하여 상기 제1 값을 생성하는 의료 엔터프라이즈 애플리케이션에 대한 검증 방법.

청구항 6

제1항에 있어서,

상기 단말은 상기 애플리케이션의 무결성이 확인되면 상기 인증 서버로부터 랜덤키를 수신하는 의료 엔터프라이즈 애플리케이션에 대한 검증 방법.

청구항 7

제6항에 있어서,

상기 단말은 상기 랜덤키와 사용자 식별자를 이용하여 사용자를 인증하는 단계를 더 포함하는 의료 엔터프라이

즈 애플리케이션에 대한 검증 방법.

청구항 8

의료 데이터를 수집하는 클라이언트 장치;

트러스트 존(trustZone)을 지원하는 프로세서를 포함하고, 상기 클라이언트 장치로부터 설치된 애플리케이션에 대한 실행 요청이 수신되면 상기 애플리케이션 설치 시에 서버로부터 수신한 시드키를 TEE(Trusted Execution Environment)에서 검증하고, 상기 검증이 성공하면 상기 애플리케이션의 식별자 및 상기 단말의 식별자 중 적어도 하나의 식별자를 해싱한 제1 값을 생성하는 단말 장치; 및

상기 제1 값과 상기 적어도 하나의 식별자를 상기 단말 장치로부터 수신하고, 상기 단말과 동일한 해싱 방법으로 수신한 상기 적어도 하나의 식별자를 해싱한 제2 값을 생성하고, 수신한 상기 제1 값과 상기 제2값을 비교하여 상기 애플리케이션에 대한 무결성을 확인하는 인증 서버를 포함하는 의료 엔터프라이즈 애플리케이션 검증 시스템.

청구항 9

제8항에 있어서,

상기 제1 값은 상기 적어도 하나의 식별자를 해싱한 값을 해싱한 시점의 날짜로 다시 해싱한 값인 의료 엔터프라이즈 애플리케이션 검증 시스템.

청구항 10

제9항에 있어서,

상기 단말 장치는 상기 인증 서버로부터 상기 무결성을 확인한 결과를 수신하고, 무결성이 확인되면 상기 단말은 상기 적어도 하나의 식별자를 해싱한 값을 저장하고, 상기 애플리케이션에 대한 검증이 다시 요청된 경우 상기 단말은 검증 시점의 날짜로 상기 저장한 값을 다시 해싱하여 제1 값을 생성하는 의료 엔터프라이즈 애플리케이션 검증 시스템.

청구항 11

제8항에 있어서,

상기 단말 장치는 상기 애플리케이션의 무결성이 확인되면 상기 인증 서버로부터 랜덤키를 수신하는 의료 엔터프라이즈 애플리케이션 검증 시스템.

청구항 12

제11항에 있어서,

상기 단말 장치는 상기 랜덤키와 사용자 식별자를 이용하여 사용자를 인증하는 의료 엔터프라이즈 애플리케이션 검증 시스템.

발명의 설명

기 술 분 야

[0001] 이하 설명하는 기술은 의료용 엔터프라이즈 애플리케이션의 무결성을 검증하는 기법에 관한 것이다.

배 경 기 술

[0002] 사용자 개인의 의료 데이터를 모니터링 하기 위하여 WBSN(Wireless Body Sensor Network) 환경이 주목받고 있다. WBSN란 지능형 센서 노드로 구성되어 사람의 일상적인 활동을 탐지하는 시스템을 말하며 의료 환경에서는 모바일 디바이스를 이용하여 환자의 진찰과 모니터링에 사용되고 있다.

[0003] 특정 기업, 병원에서는 특정 사용자만 이용 가능한 엔터프라이즈 앱을 제공하는데 앱스토어에서 설치하는 일반적인 방식과 달리 엔터프라이즈 앱은 설치 파일을 자사 웹 사이트에서 직접 내려받는다. 이 경우 앱스토어에서 발급받은 인증서를 비교하는 신원증명(Code Signing) 검증단계를 포함하지 않기 때문에 악성 앱을 판별할 수 없

고, 인터넷 엑세스 권한이 주어지는 경우 악성 앱을 내려받을 위험성이 높다.

선행기술문헌

특허문헌

[0004] (특허문헌 0001) 한국공개특허 제10-2017-0067003호

발명의 내용

해결하려는 과제

[0005] WBSN 환경이 발전하면서 의료환경에서 기존 엔터프라이즈 앱 보안을 위해 제시되었던 자사 앱스토어 구축, 사용자 모니터링 등의 방법은 한계가 있다. 실시간 데이터 전송으로 인한 데이터 증가 및 보호자와 같은 제3자 사용자 증가로 인해 제한된 환경에서 개개인의 애플리케이션을 인증할 수 있는 방법이 필요하다. 이하 설명하는 기술은 의료 엔터프라이즈 애플리케이션에 대한 무결성 인증 방법을 제공하고자 한다.

과제의 해결 수단

[0006] 의료 엔터프라이즈 애플리케이션에 대한 검증 방법은 단말이 인증 서버로부터 시드키를 수신하는 단계, 상기 단말이 설치된 애플리케이션 실행 명령을 확인하면 상기 시드키를 상기 단말의 TEE(Trusted Execution Environment)에서 검증하는 단계, 상기 단말이 상기 시드키에 대한 검증이 성공하면 상기 애플리케이션의 식별자 및 상기 단말의 식별자 중 적어도 하나의 식별자를 상기 인증 서버에 전달하는 단계 및 상기 단말이 상기 인증 서버로부터 상기 적어도 하나의 식별자를 해싱한 값을 이용하여 상기 애플리케이션의 무결성을 확인한 결과를 수신하는 단계를 포함한다.

발명의 효과

[0007] 이하 설명하는 기술은 트러스트존(TrustZone)을 지원하는 단말을 이용하여 애플리케이션에 대한 무결성을 확인할 수 있다.

도면의 간단한 설명

[0008] 도 1은 의료 엔터프라이즈 시스템의 구조에 대한 예이다.

도 2는 의료 엔터프라이즈 애플리케이션 검증 시스템의 구조에 대한 예이다.

도 3은 의료 엔터프라이즈 애플리케이션에 대한 검증 과정에 대한 예이다.

발명을 실시하기 위한 구체적인 내용

[0009] 이하 설명하는 기술은 다양한 변경을 가할 수 있고 여러 가지 실시례를 가질 수 있는 바, 특정 실시례들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나 이는 이하 설명하는 기술을 특정한 실시 형태에 대해 한정하려는 것이 아니며, 이하 설명하는 기술의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다.

[0010] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 해당 구성요소들은 상기 용어들에 의해 한정되지는 않으며, 단지 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 이하 설명하는 기술의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. 및/또는 이라는 용어는 복수의 관련된 기재된 항목들의 조합 또는 복수의 관련된 기재된 항목들 중의 어느 항목을 포함한다.

[0011] 본 명세서에서 사용되는 용어에서 단수의 표현은 문맥상 명백하게 다르게 해석되지 않는 한 복수의 표현을 포함하는 것으로 이해되어야 하고, "포함한다" 등의 용어는 설시된 특징, 개수, 단계, 동작, 구성요소, 부분품 또는 이들을 조합한 것이 존재함을 의미하는 것이지, 하나 또는 그 이상의 다른 특징들이나 개수, 단계 동작 구성요소, 부분품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 배제하지 않는 것으로 이해되어야 한다.

[0012] 도면에 대한 상세한 설명을 하기에 앞서, 본 명세서에서의 구성부들에 대한 구분은 각 구성부가 담당하는 주기능 별로 구분한 것에 불과함을 명확히 하고자 한다. 즉, 이하에서 설명할 2개 이상의 구성부가 하나의 구성부로 합쳐지거나 또는 하나의 구성부가 보다 세분화된 기능별로 2개 이상으로 분화되어 구비될 수도 있다. 그리고 이하에서 설명할 구성부 각각은 자신이 담당하는 주기능 이외에도 다른 구성부가 담당하는 기능 중 일부 또는 전부의 기능을 추가적으로 수행할 수도 있으며, 구성부 각각이 담당하는 주기능 중 일부 기능이 다른 구성부에 의해 전담되어 수행될 수도 있음은 물론이다.

[0013] 또, 방법 또는 동작 방법을 수행함에 있어서, 상기 방법을 이루는 각 과정들은 문맥상 명백하게 특정 순서를 기재하지 않은 이상 명기된 순서와 다르게 일어날 수 있다. 즉, 각 과정들은 명기된 순서와 동일하게 일어날 수도 있고 실질적으로 동시에 수행될 수도 있으며 반대의 순서대로 수행될 수도 있다.

[0015] 먼저, 일반적인 의료 엔터프라이즈 애플리케이션이 동작하는 환경에 대하여 설명한다. 도 1은 의료 엔터프라이즈 시스템(100)의 구조에 대한 예이다. 도 1은 안드로이드 스마트폰의 웹 페이지를 이용하여 내려받은 애플리케이션의 의료 데이터 전송 구조를 나타낸 것이다. 의료 엔터프라이즈 시스템(100)은 센서(50), 클라이언트 장치(110), 단말 장치(120), 웹 서버(150) 및 의료 서비스 서버(160)를 포함한다.

[0016] 센서(50)는 사용자의 신체에 부착되거나, 신체 주변에서 데이터를 센싱한다. 의료 서비스의 종류에 따라 센서(50)는 다양한 데이터를 수집할 수 있다. 예컨대, 센서(50)는 체온, 맥박, 심전도, 주변 온도, 신체의 움직임 등과 같은 다양한 정보를 수집할 수 있다. 이하 센서(50)가 센싱하는 데이터를 의료 데이터라고 명명한다.

[0017] 클라이언트 장치(110)는 센서(50)가 획득한 의료 데이터를 수집한다. 클라이언트 장치(110)는 유선 또는 무선으로 센서(50)로부터 의료 데이터를 수신한다. 클라이언트 장치(110)는 사용자의 신체에 고정되는 형태일 수도 있고, 사용자의 주변에 위치한 별도의 통신 장치(게이트웨이, AP 등)일 수도 있다. 클라이언트 장치(110)는 의료 데이터를 수신하면 즉시 또는 정기적으로 의료 데이터를 단말 장치(120)에 전달한다. 클라이언트 장치(110)는 근거리 무선 통신(예컨대, 블루투스 등)을 통해 의료 데이터를 단말 장치(120)에 전달한다. 클라이언트 장치(110)는 의료 데이터를 전송하기 전에 또는 전송하면서 단말 장치(120)에 의료 엔터프라이즈 애플리케이션의 실행을 요청할 수 있다.

[0018] 단말 장치(120)는 웹 서버(150)로부터 애플리케이션 설치 파일(예컨대, APK)을 다운로드받아 애플리케이션을 설치한다. 일반적으로 웹 서버(150)는 의료 서비스를 제공하는 사업자(병원 등)가 운영하는 웹 페이지를 위한 서버이다.

[0019] 의료 서비스 서버(160)는 의료 서비스를 제공하기 위한 서버이다. 단말 장치(120)는 수신한 의료 데이터를 의료 서비스 서버(160)에 전달한다. 단말 장치(120)는 WiFi 또는 LTE 통신망 등을 이용하여 의료 데이터를 의료 서비스 서버(160)에 전달할 수 있다.

[0020] 의료 서비스 서버(160)는 의료 데이터를 기준으로 일정한 분석을 수행한다. 예컨대, 의료 서비스 서버(160)는 사용자의 건강 상태를 분석할 수 있다. 도 1에서 웹 서버(150)와 의료 서비스 서버(160)를 별개의 객체로 도시하였지만, 웹 서버(150)와 의료 서비스 서버(160)는 물리적으로 하나의 서버일 수도 있다.

[0021] 스마트폰 앱스토어는 앱의 변조, 악성코드 발견시 사용자 알림을 제공하여 예방한다. 그러나 웹 사이트를 통해 직접 설치한 경우 범용적으로 사용되는 모바일 헬스케어 애플리케이션의 범주를 벗어난다. 다시 말하면 내부적으로 사용되는 경우 사용자 모바일 플랫폼에서 자체적 검증을 진행하여 CIA Triad(Confidentiality, Integrity, Availability)를 바탕으로 한 보안성을 확립해야 한다.

[0023] 도 2는 의료 엔터프라이즈 애플리케이션 검증 시스템(200)의 구조에 대한 예이다. 의료 엔터프라이즈 애플리케이션 검증 시스템(200)은 센서(50), 클라이언트 장치(210), 단말 장치(220) 및 인증 서버(260)를 포함한다. 단말 장치(220)는 의료 엔터프라이즈 애플리케이션을 설치한 상태라고 가정한다. 센서(50) 및 클라이언트 장치(210)는 도 1에서 설명한 센서(50)와 클라이언트 장치(110)와 동일하게 동작한다.

[0024] 단말 장치(220)는 트러스트존을 지원하는 프로세서(AP)를 포함한다. 트러스트존은 AP 칩을 통해 프로세서와 주변장치, 저장장치를 대상으로 보안 서비스를 제동한다. 단말 장치(220)는 트러스트존 하드웨어(AP)를 이용하여 개방형 OS(운영체제)에 독립적으로 격리된 실행 환경인 TEE(Trusted Execution Environment)를 제공한다. 안드로이드 기기는 보안성을 위해 TEE 보안 플랫폼을 탑재하는데 TEE는 운영체제의 접근 가능/불가능 영역을 분리하

여 고립된 환경을 제공하는 것으로 내부에 로드된 코드 및 데이터의 기밀성과 무결성을 보장한다.

[0025] 인증 서버(250)는 단말 장치(220)에 설치된 의료 엔터프라이즈 애플리케이션에 대한 무결성을 검증한다. 인증 서버(250)는 도 1에서 도시한 웹 서버(150) 또는 의료 서비스 서버(160)와 별개의 서버일 수 있다. 또는 도 1에서 도시한 웹 서버(150) 또는 의료 서비스 서버(160)가 인증 서버(250)로 동작할 수도 있다.

[0026] 이하 의료 엔터프라이즈 애플리케이션 검증 시스템(200)의 동작을 설명한다. 도 3은 의료 엔터프라이즈 애플리케이션에 대한 검증 과정(300)에 대한 예이다. 단말 장치(220)의 Rich OS가 구동되는 일반 영역(REE, Rich Execution Environments)과 REE와는 독립된 격리 실행 환경인 TEE를 갖는다. 도 3에서 동일한 단말 장치(220)에서 REE에 설치된 의료 애플리케이션과 TEE에 설치된 트러스트존 어플리케이션을 구분하였다. 의료 엔터프라이즈 애플리케이션은 REE에 설치되고, REE에서 동작한다. 트러스트존 애플리케이션은 의료 엔터프라이즈 애플리케이션이 설치되면서 TEE에 설치된 별도의 애플리케이션에 해당한다. 트러스트존 애플리케이션을 의료 엔터프라이즈 애플리케이션 검증을 위한 동작을 수행한다. 이하 단말 장치(220)의 동작이지만, REE와 TEE의 동작을 구분하기 위하여 REE에서 동작하는 의료 애플리케이션(220A)과 TEE에서 동작하는 트러스트존 애플리케이션(220B)으로 구분하여 설명한다.

[0027] 단말 장치(220)는 의료 엔터프라이즈 애플리케이션을 서버로부터 다운받을 때 블록 암호 알고리즘을 이용한 시드키(seed key)를 발급받는다고 가정한다(301). 예컨대, 시드키는 애플리케이션의 식별자를 암호화한 값일 수 있다. 물론 블록 암호 알고리즘 외에 인증을 위한 다른 암호화 방식을 사용할 수도 있다. 의료 애플리케이션(220A)이 발급받은 시드키를 관리한다.

[0028] 의료 애플리케이션(220A)은 클라이언트 장치(210)로부터 애플리케이션 구동에 대한 명령 내지 요청을 받는다(311). 의료 애플리케이션(220A)은 시드키 K_{seed}와 애플리케이션 식별자 A_{ID}를 트러스트존 애플리케이션(220B)에 전달한다(321). 트러스트존 애플리케이션(220B)은 K_{seed}를 이용하여 암호화 데이터를 복호하고, 복호한 데이터와 A_{ID}를 비교하여 시드키를 검증할 수 있다(322). 트러스트존 애플리케이션(220B)은 시드키를 검증한 결과를 의료 애플리케이션(220A)에 전달한다(323).

[0029] 트러스트존 애플리케이션(220B)은 시드키 검증 결과가 성공(K_{seed}=true)이면, 애플리케이션 식별자 A_{ID}를 인증 서버(250)에 전달한다(331).

[0030] 의료 애플리케이션(220A)은 시드키 검증이 성공이면, 애플리케이션 검증을 위한 해싱값(x)을 생성한다(341). 해싱값(x)는 다양한 방법으로 생성될 수 있다. 의료 애플리케이션(220A)은 애플리케이션 식별자 A_{ID} 및 디바이스(단말 장치) 식별자 D_{ID} 중 적어도 하나를 이용하여 해싱값을 생성할 수 있다. 예컨대, 의료 애플리케이션(220A)은 A_{ID}와 D_{ID}을 모두 이용하여 $H(A_{ID}, D_{ID})$로 해싱값을 생성할 수 있다. 나아가 의료 애플리케이션(220A)은 식별자를 이용하여 해싱한 값을 현재 날짜(date)로 다시 해싱하여 해상값(x)을 생성할 수도 있다. 도 3에서는 식별자를 이용하여 해싱한 값을 날짜로 이중 해싱한 예를 도시한다. 즉, 의료 애플리케이션(220A)은 $x = H(H(A_{ID}, D_{ID}), date)$로 해싱값을 생성할 수 있다. 이하 의료 애플리케이션(220A)은 식별자를 이용하여 해싱한 값을 현재 날짜(date)로 다시 해싱하여 해상값(x)을 생성한다고 가정한다. 물론 해싱 기법 자체는 다양한 방법을 사용할 수 있다.

[0031] 한편, 인증 서버(250)는 트러스트존 애플리케이션(220B)이 전달한 식별자를 이용하여 애플리케이션 검증을 위한 해싱값(y)을 별도로 생성한다(351). 인증 서버(250)가 생성하는 해싱값은 의료 애플리케이션(220A)이 이용한 해싱 방식과 동일해야 한다. 따라서 인증 서버(250)는 $y = H(H(A_{ID}, D_{ID}), date)$로 해싱값을 생성한다. 인증 서버(250)는 특정 애플리케이션이 설치된 디바이스 식별자 D_{ID}를 사전에 보유할 수 있다. 인증 서버(250)는 의료 엔터프라이즈 애플리케이션이 설치되는 과정에 단말 장치의 디바이스 식별자 D_{ID}를 저장할 수 있다. 만약 인증 서버(250)가 디바이스 식별자 D_{ID}를 관리하지 않는다면, 트러스트존 애플리케이션(220B)으로부터 애플리케이션 식별자 A_{ID} 및 디바이스(단말 장치) 식별자 D_{ID}를 모두 전달받아야 한다.

[0032] 의료 애플리케이션(220A)은 자신이 생성한 해싱값 x를 인증 서버(250)에 전달한다(361). 인증 서버(250)는 수신한 해싱값 x와 자신이 생성한 해싱값 y를 비교하여 애플리케이션에 대한 무결성 검증을 수행한다(362). 검증 결과가 성공(true)이면 인증 서버(250)는 인증 결과는 의료 애플리케이션(220A)에 전달한다(371). 이때 인증 서버

(250)는 랜덤키를 의료 애플리케이션(220A)에 전달할 수 있다.

[0033] 랜덤키는 자동 로그인, 비정상 종료 앱을 실행할 경우 남아있는 쿠키, 세션 값을 이용하여 악의적으로 데이터를 변형하거나 암호화 키를 가로채는 경우를 사전에 방지하기 위해 임시키로 사용되며 사용자(환자, 의사 및 관련된 제3자)의 ID, 사용자 비밀번호(사용자 코드)와 함께 암호화하여 사용자 인증에 이용한다.

[0034] 한편 의료 애플리케이션(220A)은 애플리케이션에 대한 무결성 검증이 성공한 경우 일정한 해싱값을 저장할 수 있다. 예컨대, 의료 애플리케이션(220A)은 $H(A_{ID}, D_{ID})$를 저장할 수 있다. 이후 애플리케이션을 재실행하거나, 사용 중에 애플리케이션에 대한 검증이 다시 요청되는 경우, 의료 애플리케이션(220A)은 저장한 값($H(A_{ID}, D_{ID})$)을 해당 시점의 날짜(date)로 이중 해싱한 값을 인증 서버(250)에 전달할 수 있다. 즉, 모든 과정을 반복하지 않고 날짜만을 기준으로 해싱값을 업데이트하는 것이다. 이를 통해 계산량을 줄여 오버헤드를 감소할 수 있다.

[0036] 본 실시례 및 본 명세서에 첨부된 도면은 전술한 기술에 포함되는 기술적 사상의 일부를 명확하게 나타내고 있는 것에 불과하며, 전술한 기술의 명세서 및 도면에 포함된 기술적 사상의 범위 내에서 당업자가 용이하게 유추할 수 있는 변형 예와 구체적인 실시례는 모두 전술한 기술의 권리범위에 포함되는 것이 자명하다고 할 것이다.

부호의 설명

[0037] 50 : 센서

100 : 의료 엔터프라이즈 시스템

110 : 클라이언트 장치

120 : 단말 장치

150 : 웹 서버

160 : 의료서비스 서버

200 : 애플리케이션 검증 시스템

210 : 클라이언트 장치

220 : 단말 장치

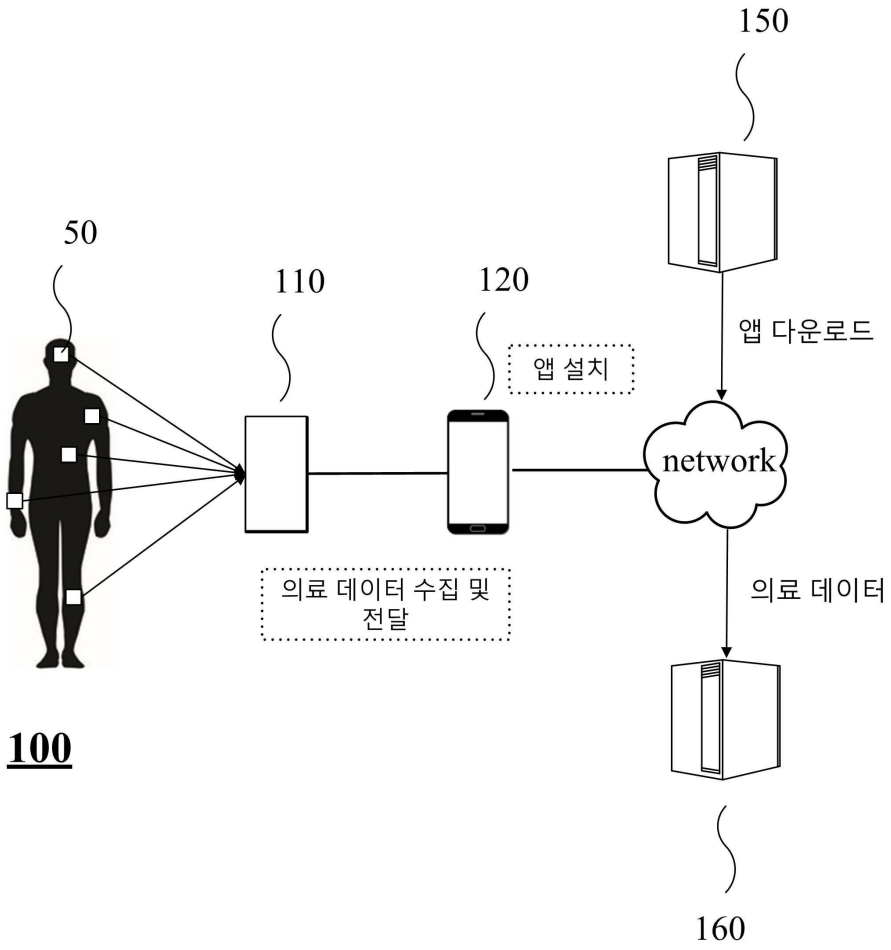
220A : 의료 애플리케이션

220B : 트러스트존 애플리케이션

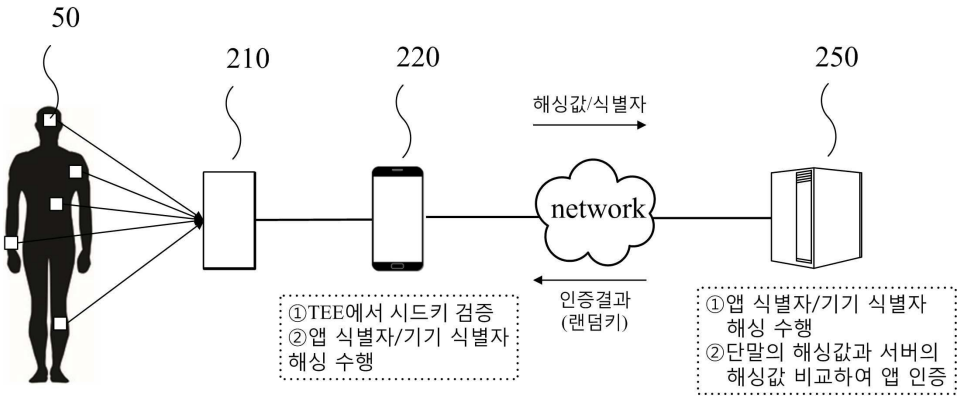
250 : 인증 서버

도면

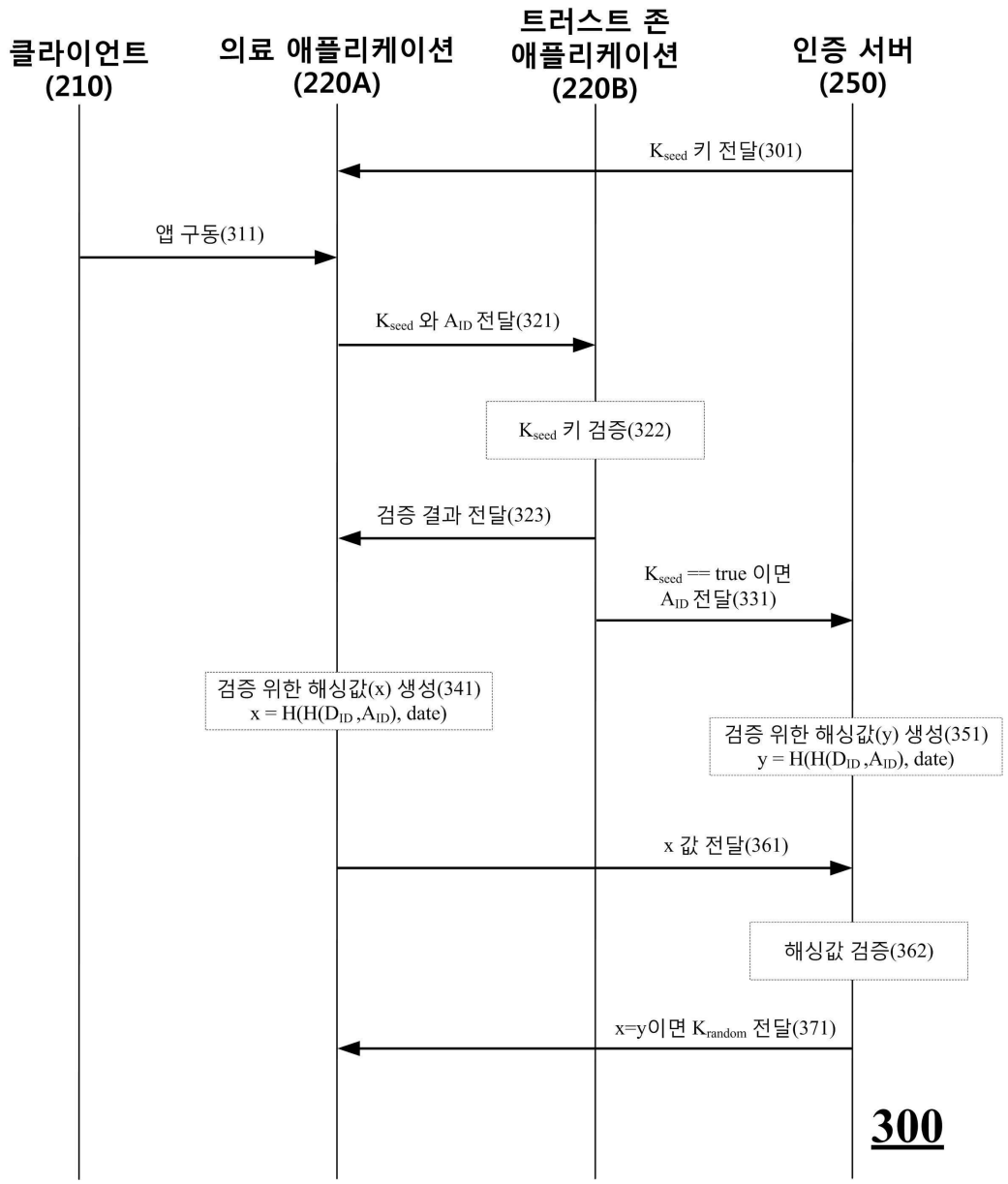
도면1


150
50
110
120
앱 다운로드
앱 설치
network
의료 데이터 수집 및
전달
의료 데이터
100
160

도면2


50
210
220
250
해싱값/식별자
network
①TEE에서 시드키 검증
②앱 식별자/기기 식별자
해싱 수행
인증결과
(랜덤키)
①앱 식별자/기기 식별자
해싱 수행
②단말의 해싱값과 서버의
해싱값 비교하여 앱 인증
200

도면3


클라이언트
(210)
의료 애플리케이션
(220A)
트러스트 존
애플리케이션
(220B)
인증 서버
(250)
K_{seed} 키 전달(301)
앱 구동(311)
K_{seed} 와 A_{ID} 전달(321)
K_{seed} 키 검증(322)
검증 결과 전달(323)
K_{seed} == true 이면
A_{ID} 전달(331)
검증 위한 해싱값(x) 생성(341)
x = H(H(D_{ID} ,A_{ID}), date)
검증 위한 해싱값(y) 생성(351)
y = H(H(D_{ID} ,A_{ID}), date)
x 값 전달(361)
해싱값 검증(362)
x=y이면 K_{random} 전달(371)
300