



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2020-0051262
(43) 공개일자 2020년05월13일

(51) 국제특허분류(Int. Cl.)
G06F 21/57 (2013.01) G06F 21/54 (2013.01)
G06F 21/55 (2013.01)
(52) CPC특허분류
G06F 21/577 (2013.01)
G06F 21/54 (2013.01)
(21) 출원번호 10-2018-0134411
(22) 출원일자 2018년11월05일
심사청구일자 2018년11월05일

(71) 출원인
충남대학교산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
(72) 발명자
조은선
대전광역시 유성구 어은로 57, 110동 1504호
목성균
대전광역시 서구 가장로 131-2, 403호
이수연
대전광역시 유성구 구즉로 16, 103동 302호
(74) 대리인
홍성욱, 심경식

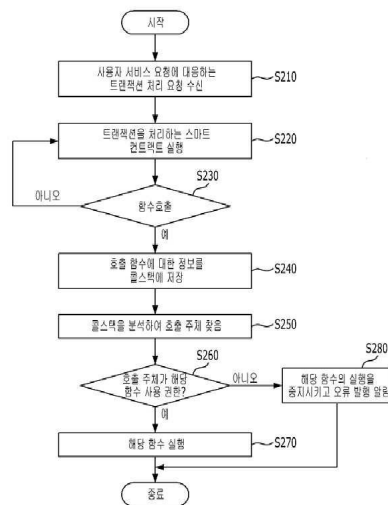
전체 청구항 수 : 총 7 항

(54) 발명의 명칭 스마트 컨트랙트 실행 장치 및 그 장치에서 취약점 탐지 방법

(57) 요약

스마트 컨트랙트 실행 장치 및 그 장치에서 취약점 탐지 방법이 개시된다. 본 발명의 일 실시예에 따른 스마트 컨트랙트 실행 장치는, 사용자 요청에 대응하여 트랜잭션을 처리하는 장치에 있어서, 상기 트랜잭션을 처리하는 적어도 하나 이상의 스마트 컨트랙트 실행 시 호출 함수에 대한 정보가 저장된 콜스택(call stack)을 검사하여 호출 주체를 찾고, 해당 스마트 컨트랙트에 설정된 보안 정책에 기초하여 상기 호출 주체의 해당 함수 호출 권한 여부에 판단하며, 그 판단결과에 따라 함수 실행 또는 실행 중지하는 가상머신을 포함한다.

대표도 - 도2



(52) CPC특허분류

G06F 21/55 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호 2018-0-00251

부처명 과학기술정보통신부

연구관리전문기관 정보통신기술진흥센터

연구사업명 정보보호핵심원천기술개발

연구과제명 스마트 컨트랙트 프라이머시 보호 및 취약점 분석 기술개발

기 여 율 1/1

주관기관 충남대학교 산학협력단

연구기간 2018.04.01 ~ 2020.12.31

명세서

청구범위

청구항 1

사용자 요청에 대응하여 트랜잭션을 처리하는 장치에 있어서,

상기 트랜잭션을 처리하는 적어도 하나 이상의 스마트 컨트랙트 실행 시 호출 함수에 대한 정보가 저장된 콜스택(call stack)을 검사하여 호출 주체를 찾고, 해당 스마트 컨트랙트에 설정된 보안 정책에 기초하여 상기 호출 주체의 해당 함수 호출 권한 여부에 판단하며, 그 판단결과에 따라 함수 실행 또는 실행 중지하는 가상머신;

을 포함하는 스마트 컨트랙트 실행 장치.

청구항 2

제1항에 있어서,

상기 가상머신은,

개발자가 생성한 스마트 컨트랙트를 배포한 경우, 상기 스마트 컨트랙트의 계정 ID, 배포자 ID 및 생성자 ID를 생성하여 상기 스마트 컨트랙트와 함께 등록하는 것을 특징으로 하는 스마트 컨트랙트 실행 장치.

청구항 3

제1항에 있어서,

상기 콜스택에는 함수가 호출되는 시간 순에 의하여 순차적으로 호출 함수에 대한 정보가 저장되고,

상기 호출 함수에 대한 정보는 호출 함수, 호출 함수를 가진 스마트 컨트랙트, 그 스마트 컨트랙트의 식별정보 및 보안정책을 포함하는 것을 특징으로 하는 스마트 컨트랙트 실행 장치.

청구항 4

제3항에 있어서,

상기 스마트 컨트랙트의 식별정보는 계정 ID, 배포자 ID, 생성자 ID의 조합으로 구성된 것으로 하는 스마트 컨트랙트 실행 장치.

청구항 5

제1항에 있어서,

상기 가상머신은,

상기 콜 스택 트레이스상에 나타나는 체인(chain)의 스마트 컨트랙트 식별정보의 리스트를 가공하여 호출 시점의 크리덴셜(credential)로 생성하고, 상기 생성된 호출시점의 크리덴셜을 호출 주체로 추출하는 것을 특징으로 하는 스마트 컨트랙트 실행 장치.

청구항 6

스마트 컨트랙트 실행 장치가 취약점을 탐지하는 방법에 있어서,

사용자 요청에 대응하여 트랜잭션을 처리하는 적어도 하나 이상의 스마트 컨트랙트 실행 시 호출 함수에 대한 정보가 저장된 콜스택(call stack)을 검사하여 호출 주체를 추출하는 단계;

해당 스마트 컨트랙트에 설정된 보안 정책에 기초하여 상기 호출 주체의 해당 함수 호출 권한 여부에 판단하고, 그 판단결과에 따라 함수 실행 또는 실행 중지하는 단계;

를 포함하는 스마트 컨트랙트 실행 장치에서 취약점 탐지 방법.

청구항 7

제6항에 있어서,

상기 호출 주체를 추출하는 단계는,

상기 사용자 요청에 대응하는 트랜잭션을 처리하는 스마트 컨트랙트를 실행하는 단계;

상기 스마트 컨트랙트의 실행 시, 함수 호출이 발생한 경우, 호출 함수에 대한 정보를 상기 콜스택에 저장하는 단계; 및

상기 콜스택의 트레이스상에 나타나는 체인(chain)의 스마트 컨트랙트 식별정보의 리스트를 가공하여 호출 시점의 크리덴셜(credential)로 생성하고, 상기 생성된 호출시점의 크리덴셜을 호출 주체로 추출하는 단계를 포함하는 것을 특징으로 하는 스마트 컨트랙트 실행 장치에서 취약점 탐지 방법.

발명의 설명**기술 분야**

[0001] 본 발명은 스마트 컨트랙트 실행 장치 및 그 장치에서 취약점 탐지 방법에 관한 것으로, 특히 스마트 컨트랙트의 실행 중에 콜 스택(call stack)을 검사하여 취약점을 탐지하는 스마트 컨트랙트 실행 장치 및 그 장치에서 취약점 탐지 방법에 관한 것이다.

배경 기술

[0003] 최초의 블록체인 기반 스마트 컨트랙트는 비트코인 스크립트이다. 비트코인 트랜잭션에 원시 언어인 OPCODE로 스크립트를 작성해서 보내면 조건에 따라 자동으로 거래를 수행한다. 하지만, 비트코인 스크립트는 반복문을 사용할 수 없고, 비트코인 잔고 외의 다른 정보를 관리 할 수 없는 한계가 있다. 이는 블록체인의 특이한 구조 때문인데 비트코인 스크립트에서 반복문을 허용할 경우 만약 스크립트 조건 때문에 무한 루프가 발생할 경우 네트워크 전체가 멈출 수 있다. 사용자는 무한 루프를 통해 쉽게 DoS(Denial of Service) 공격을 할 수 있다.

[0004] 이더리움은 이러한 비트코인 스크립팅 시스템의 한계를 극복하고자 나온 스마트 컨트랙트 특화 블록체인 플랫폼이다. 비트코인 스크립팅 시스템의 한계인 다양한 상태 저장과 반복문을 허용한 스마트 컨트랙트를 만들었다.

[0005] 그러나, 종래의 스마트 컨트랙트에서는 함수 호출의 주체를 확인하지 않으므로, 허가되지 않은 주체가 함수를 호출할 수 있고, 이로 인해 심각한 보안문제가 발생하는 문제가 있다. 예컨대, 악의적인 목적을 가진 스마트 컨트랙트 (가)가 스마트 컨트랙트 (나)의 함수를 이용해 스마트 컨트랙트 (다)의 함수를 호출했다고 가정했을 때, 스마트 컨트랙트 (다)의 함수가 계좌에서 돈을 빼내는 동작을 한다면 심각한 보안문제가 발생한다.

[0006] 이에, 스마트 컨트랙트 실행 시, 함수 호출의 주체를 확인하여, 스마트 컨트랙트의 신뢰성을 보장할 수 있는 기술 개발이 요구되고 있다.

선행기술문헌**특허문헌**

[0008] (특허문헌 0001) 한국 등록특허공보 제 10-1739125 호(2017.05.24. 공고)

발명의 내용**해결하려는 과제**

[0009] 본 발명이 해결하고자 하는 과제는 스마트 컨트랙트의 실행 중에 호출 주체를 확인하여, 스마트 컨트랙트의 신

뢰성을 보장할 수 있는 스마트 컨트랙트 실행 장치 및 그 장치에서 취약점 탐지 방법을 제공하는 것이다.

[0010] 본 발명이 해결하고자 하는 과제는 이상에서 언급한 과제(들)로 제한되지 않으며, 언급되지 않은 또 다른 과제(들)은 아래의 기재로부터 당업자에게 명확하게 이해될 수 있을 것이다.

과제의 해결 수단

[0012] 상기한 목적을 달성하기 위하여, 본 발명의 일 실시예에 따른 스마트 컨트랙트 실행 장치는, 사용자 요청에 대응하여 트랜잭션을 처리하는 장치에 있어서, 상기 트랜잭션을 처리하는 적어도 하나 이상의 스마트 컨트랙트 실행 시 호출 함수에 대한 정보가 저장된 콜스택(call stack)을 검사하여 호출 주체를 찾고, 해당 스마트 컨트랙트에 설정된 보안 정책에 기초하여 상기 호출 주체의 해당 함수 호출 권한 여부에 판단하며, 그 판단결과에 따라 함수 실행 또는 실행 중지하는 가상머신을 포함한다.

[0013] 바람직하게는, 상기 가상머신은, 개발자가 생성한 스마트 컨트랙트를 배포한 경우, 상기 스마트 컨트랙트의 계정 ID, 배포자 ID 및 생성자 ID를 생성하여 상기 스마트 컨트랙트와 함께 등록할 수 있다.

[0014] 바람직하게는, 상기 콜스택에는 함수가 호출되는 시간 순에 의하여 순차적으로 호출 함수에 대한 정보가 저장되고, 상기 호출 함수에 대한 정보는 호출 함수, 호출 함수를 가진 스마트 컨트랙트, 그 스마트 컨트랙트의 식별 정보 및 보안정책을 포함할 수 있다.

[0015] 바람직하게는, 상기 스마트 컨트랙트의 식별정보는 계정 ID, 배포자 ID, 생성자 ID의 조합으로 구성될 수 있다.

[0016] 바람직하게는, 상기 가상머신은, 상기 콜 스택 트레이스상에 나타나는 체인(chain)의 스마트 컨트랙트 식별정보의 리스트를 가공하여 호출 시점의 크리덴셜(credential)로 생성하고, 상기 생성된 호출시점의 크리덴셜을 호출 주체로 추출할 수 있다.

[0017] 본 발명의 다른 실시예에 따른 스마트 컨트랙트 실행 장치에서 취약점 탐지 방법은, 스마트 컨트랙트 실행 장치가 취약점을 탐지하는 방법에 있어서, 사용자 요청에 대응하여 트랜잭션을 처리하는 적어도 하나 이상의 스마트 컨트랙트 실행 시 호출 함수에 대한 정보가 저장된 콜스택(call stack)을 검사하여 호출 주체를 추출하는 단계, 해당 스마트 컨트랙트에 설정된 보안 정책에 기초하여 상기 호출 주체의 해당 함수 호출 권한 여부에 판단하고, 그 판단결과에 따라 함수 실행 또는 실행 중지하는 단계를 포함한다.

[0018] 바람직하게는, 상기 호출 주체를 추출하는 단계는, 상기 사용자 요청에 대응하는 트랜잭션을 처리하는 스마트 컨트랙트를 실행하는 단계, 상기 스마트 컨트랙트의 실행 시, 함수 호출이 발생한 경우, 호출 함수에 대한 정보를 상기 콜스택에 저장하는 단계, 상기 콜스택의 트레이스상에 나타나는 체인(chain)의 스마트 컨트랙트 식별정보의 리스트를 가공하여 호출 시점의 크리덴셜(credential)로 생성하고, 상기 생성된 호출시점의 크리덴셜을 호출 주체로 추출하는 단계를 포함할 수 있다.

발명의 효과

[0020] 본 발명에 따르면, 스마트 컨트랙트의 실행 중에 콜 스택(call stack)의 검사를 통해 허가되지 않은 주체가 함수를 호출하는지 판단하고, 그 판단결과에 따라 잘못된 호출을 저지할 수 있으므로, 악의적인 접근을 차단할 수 있고 이로 인해 보안문제의 신뢰성을 높일 수 있다.

[0021] 본 발명의 효과들은 이상에서 언급한 효과들로 제한되지 않으며, 언급되지 않은 또 다른 효과들은 아래의 기재로부터 통상의 기술자에게 명확하게 이해될 수 있을 것이다.

도면의 간단한 설명

[0023] 도 1은 본 발명의 일 실시예에 따른 스마트 컨트랙트 서비스 시스템을 설명하기 위한 도면이다.

도 2는 본 발명의 일 실시예에 따른 스마트 컨트랙트의 취약점을 탐지하는 방법을 설명하기 위한 도면이다.

도 3은 본 발명의 일 실시예에 따른 스마트 컨트랙트 동작을 설명하기 위한 예시도이다.

도 4는 본 발명의 일 실시예에 따른 콜 스택을 설명하기 위한 예시도이다.

발명을 실시하기 위한 구체적인 내용

- [0024] 본 발명은 다양한 변경을 가할 수 있고 여러 가지 실시예를 가질 수 있는 바, 특정 실시예들을 도면에 예시하고 상세하게 설명하고자 한다. 그러나 이는 본 발명을 특정한 실시 형태에 한정하려는 것이 아니며, 본 발명의 사상 및 기술 범위에 포함되는 모든 변경, 균등물 내지 대체물을 포함하는 것으로 이해되어야 한다. 각 도면을 설명하면서 유사한 참조부호를 유사한 구성요소에 대해 사용하였다.
- [0025] 제1, 제2, A, B 등의 용어는 다양한 구성요소들을 설명하는데 사용될 수 있지만, 상기 구성요소들은 상기 용어들에 의해 한정되어서는 안 된다. 상기 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 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.
- [0026] 어떤 구성요소가 다른 구성요소에 "연결되어" 있다거나 "접속되어" 있다고 언급된 때에는, 그 다른 구성요소에 직접적으로 연결되어 있거나 또는 접속되어 있을 수도 있지만, 중간에 다른 구성요소가 존재할 수도 있다고 이해되어야 할 것이다. 반면에, 어떤 구성요소가 다른 구성요소에 "직접 연결되어" 있다거나 "직접 접속되어" 있다고 언급된 때에는, 중간에 다른 구성요소가 존재하지 않는 것으로 이해되어야 할 것이다.
- [0027] 본 출원에서 사용한 용어는 단지 특정한 실시예를 설명하기 위해 사용된 것으로, 본 발명을 한정하려는 의도가 아니다. 단수의 표현은 문맥상 명백하게 다르게 뜻하지 않는 한, 복수의 표현을 포함한다. 본 출원에서, "포함하다" 또는 "가지다" 등의 용어는 명세서상에 기재된 특징, 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것이 존재함을 지정하려는 것이지, 하나 또는 그 이상의 다른 특징들이나 숫자, 단계, 동작, 구성요소, 부품 또는 이들을 조합한 것들의 존재 또는 부가 가능성을 미리 배제하지 않는 것으로 이해되어야 한다.
- [0028] 다르게 정의되지 않는 한, 기술적이거나 과학적인 용어를 포함해서 여기서 사용되는 모든 용어들은 본 발명이 속하는 기술분야에서 통상의 지식을 가진 자에 의해 일반적으로 이해되는 것과 동일한 의미를 가지고 있다. 일반적으로 사용되는 사전에 정의되어 있는 것과 같은 용어들은 관련 기술의 문맥상 가지는 의미와 일치하는 의미를 가지는 것으로 해석되어야 하며, 본 출원에서 명백하게 정의하지 않는 한, 이상적이거나 과도하게 형식적인 의미로 해석되지 않는다.
- [0029] 이하에서는 본 발명에 따른 바람직한 실시예를 첨부된 도면을 참조하여 상세하게 설명한다.
- [0030] 도 1은 본 발명의 일 실시예에 따른 스마트 컨트랙트 서비스 시스템을 설명하기 위한 도면이다.
- [0031] 도 1을 참조하면, 본 발명의 일 실시예에 따른 스마트 컨트랙트 서비스 시스템은 개발자 장치(100), 사용자 장치(200), 클라이언트(300), 블록체인(400)을 포함한다.
- [0032] 개발자 장치(100)는 스마트 컨트랙트를 생성하여 블록체인(400)에 등록한다. 즉, 개발자 장치(100)는 스마트 컨트랙트를 코딩하고, 코딩된 소스 코드를 컴파일하여 배포한다. 이때, 개발자는 구현하고자 하는 내용을 솔리디티나 다른 언어로 코딩하고, 코딩된 소스 코드를 컴파일하여 바이트 코드를 생성하며, 생성된 바이트 코드, 즉 스마트 컨트랙트를 배포한다. 스마트 컨트랙트를 배포한다는 것은 컴파일된 바이트 코드를 하나의 트랜잭션처럼 블록에 추가시켜 블록체인(400)에 등록시키는 작업이다.
- [0033] 사용자 장치(200)는 블록체인 데이터를 직접 저장하지 않고, 클라이언트(300)에 접속하여 스마트 컨트랙트를 이용한다. 이때, 사용자는 웹 브라우저나 ssh와 같은 서비스를 이용할 수 있다.
- [0034] 클라이언트(300)는 블록체인(400)에 참여하는 노드, 즉, 블록이 생성되면 그 정보를 전파받는다. 따라서, 클라이언트(300)는 싱크된 블록체인 데이터를 가지고 있다.
- [0035] 또한, 클라이언트(300)는 블록체인 네트워크의 노드이면서, 일반 사용자의 접속을 허용하고 블록체인과 연결시켜주는 역할도 수행한다. 그래서 일반 사용자(200)는 클라이언트(300)에 접속하여 geth와 같은 명령어로 블록체인 정보를 얻거나, 스마트 컨트랙트를 사용할 수 있다.
- [0036] 또한, 클라이언트(300)는 블록체인 데이터를 모두 가지고 있다. 개발자(100)가 생성한 스마트 컨트랙트는 클라이언트(300)를 통해서 블록체인(400)에 포함되고, 결국 블록체인(400)에 연결된다. 여기서, 스마트 컨트랙트가 블록체인(400)에 포함된다는 것은 블록체인 네트워크 상의 모든 노드들이 동일한 스마트 컨트랙트를 가지고 있

는 것이다. 즉, 개발자(100)가 생성한 스마트 컨트랙트 코드가 블록에 기록되어 다른 노드에서도 코드가 복사된다.

- [0037] 클라이언트(300)는 개발자가 생성한 스마트 컨트랙트를 배포한 경우, 그 스마트 컨트랙트를 블록에 등록한다. 스마트 컨트랙트 배포 시, 클라이언트(300)는 그 스마트 컨트랙트의 계정 ID, 배포자 ID, 생성자 ID를 생성하여, 스마트 컨트랙트와 함께 등록한다. 여기서, 계정 ID는 스마트 컨트랙트가 포함된 블록을 채굴한 경우 생성되는 식별정보, 배포자 ID는 스마트 컨트랙트가 최초 배포될 때 생성되는 식별정보, 생성자 ID는 스마트 컨트랙트 등록 시 생성되는 새로운 컨트랙트 객체의 식별정보를 의미할 수 있다.
- [0038] 만약, 스마트 컨트랙트 배포 시, 그 스마트 컨트랙트의 계정 ID와 배포자 ID만이 생성된 경우, 클라이언트(300)는 생성자 ID를 배포자 ID와 동일하게 하고, 스마트 컨트랙트와 함께 등록할 수 있다.
- [0039] 한편, 스마트 컨트랙트는 컴퓨터언어인 실행코드들로 작성되므로 특정 조건을 만족했을 때 자동적으로 프로그램이 실행되어 데이터를 저장하거나 그 값을 제공하도록 구현할 수 있다. 여기서, 특정 조건은 보안정책을 포함할 수 있고, 보안정책은 함수 호출 권한 정보 등을 포함할 수 있으며, 문자열, 정수, 소수점 등의 형태일 수 있다. 따라서, 클라이언트(300)는 특정 조건에 만족된 경우 기능을 수행하는 스마트 컨트랙트(Smart Contract)를 이용하여 블록체인(400)에 자원을 공유하거나 작업에 대한 트랜잭션을 생성한다. 각각의 스마트 컨트랙트는 기능 실행에 필요한 변수와 메소드를 가지며 블록체인 네트워크에 배포된 스마트 컨트랙트는 그 식별 정보로서 계정을 가진다. 이처럼, 스마트 컨트랙트에는 보안정책이 설정되어 있으므로, 클라이언트(300)는 스마트 컨트랙트 실행 중 특정 함수 호출 시 그 특정 함수를 호출한 주체의 특정 함수의 호출 권한을 보안정책을 이용하여 판단할 수 있다. 이때, 보안정책은 스마트 컨트랙트마다 가지고 있을 수도 있고, 모든 스마트컨트랙트에 일괄적으로 적용되는 보안정책이 있을 수도 있다. 또한, 보안 정책은 호출 시점의 크리덴셜(credential)을 포함해서 판단하므로, 호출 시점의 크리덴셜을 기준으로 결정될 수 있다.
- [0040] 또한, 클라이언트(300)는 사용자 서비스 요청에 대응하는 스마트 컨트랙트를 VM(Virtual Machine)(350)에서 실행시킨다. 즉, 스마트 컨트랙트가 블록체인(400)에 올라가면 블록체인 상의 모든 노드는 해당 스마트 컨트랙트의 바이트 코드를 가지고 있게 된다. 따라서, 클라이언트(300)는 해당 스마트 컨트랙트를 자신의 VM(350)에서 실행시킬 수 있다. 한 노드에서 스마트 컨트랙트의 내용을 변경시키는 명령은 트랜잭션을 발생시켜 내용 변경이 이루어져, 다른 모드 노드가 컨트랙트에 접근하면 바뀐 내용을 얻게 된다.
- [0041] 또한, 클라이언트(300)의 VM(350)은 스마트 컨트랙트가 실행될 때 콜 스택을 검사하여 호출의 주체를 찾고, 찾은 주체가 해당 함수를 호출할 권한이 있는지 확인하여, 호출 권한이 있는 경우 해당 함수를 실행시키지만, 호출 권한이 없는 경우 실행을 중지하고 오류 발생을 알린다. 즉, 가상머신(350)이 스마트 컨트랙트를 실행하기 시작하면 콜 스택이 생기고, 함수 호출이 발생하면 호출 함수에 대한 정보가 콜 스택에 쌓이게 된다. 따라서, 가상머신(350)은 콜 스택 트레이스상에 나타나는 콜 체인(chain)의 스마트 컨트랙트 식별정보의 리스트를 가공하여 호출 시점의 크리덴셜(credential)로 생성하고, 생성된 호출시점의 크리덴셜을 호출 주체로 추출할 수 있다. 이처럼 가상머신(350)은 호출 함수에 대한 정보가 저장된 콜 스택을 분석하면, 어떤 주체가 함수를 호출했는지 알 수 있다. 호출 주체가 찾아지면, 가상머신(350)은 해당 스마트 컨트랙트에 설정된 보안 정책에 기초하여 찾아낸 주체가 해당 함수를 호출할 권한이 있는지 확인할 수 있다.
- [0042] 클라이언트(300)가 콜 스택을 검사하여 호출 주체의 호출 권한 여부를 판단하고, 그 판단결과에 따라 해당 함수를 실행시키는 기술에 대한 상세한 설명은 후술하기로 한다.
- [0044] 도 2는 본 발명의 일 실시예에 따른 스마트 컨트랙트의 취약점을 탐지하는 방법을 설명하기 위한 도면, 도 3은 본 발명의 일 실시예에 따른 스마트 컨트랙트 동작을 설명하기 위한 예시도, 도 4는 본 발명의 일 실시예에 따른 콜 스택을 설명하기 위한 예시도이다.
- [0045] 도 2를 참조하면, 클라이언트의 VM은 사용자 서비스 요청에 대응하는 트랜잭션 처리 요청이 수신되면(S210), 트랜잭션을 처리하는 스마트 컨트랙트를 실행한다(S220). 즉, 가상머신은 사용자의 서비스 요청에 따라, 트랜잭션을 수행한다. 이때 트랜잭션은 요청된 서비스를 수행하는 전체 프로세스를 의미하는 것으로, 하나 이상의 스마트 컨트랙트를 실행하여 요청된 서비스를 수행할 수 있고, 스마트 컨트랙트 실행 중, 적어도 하나의 함수가 호출되어 실행될 수 있다.
- [0046] 단계 S220의 실행 후, 함수 호출이 발생하면(S230), 가상머신은 호출 함수에 대한 정보를 콜스택에 저장한다(S240). 즉, 가상머신은 스마트 컨트랙트가 실행되면, 콜스택을 생성하고, 함수가 호출되는 경우 호출 함수에

대한 정보를 콜스택에 저장한다. 이때, 가상머신은 함수가 호출되는 시간 순에 의하여 순차적으로 호출 함수에 대한 정보를 콜스택에 저장한다. 호출 함수에 대한 정보는 호출 함수, 호출 함수를 가진 스마트 컨트랙트 및 각 스마트 컨트랙트의 식별정보 등을 포함할 수 있고, 스마트 컨트랙트의 식별정보는 계정 ID, 배포자 ID, 생성자 ID의 조합으로 구성될 수 있다. 또한, 콜 스택에 저장된 호출 함수에 대한 정보는 필요시 스마트 컨트랙트의 보안정책을 더 포함할 수도 있다.

[0047] 예를 들어, 스마트 컨트랙트 sc1의 계정 ID가 sid1, 배포자 ID가 did1, 생성자 ID가 cid1인 경우, 스마트 컨트랙트 식별정보는 'F<sid1, did1, cid1>'형태일 수 있다. 만약, 스마트 컨트랙트에 계정 ID와 배포자 ID만이 존재하는 경우, 가상머신은 생성자 ID를 배포자 ID와 동일하게 하여, 컨트랙트 식별정보를 생성할 수 있다.

[0048] 스마트 컨트랙트의 보안정책은 각 스마트 컨트랙트 별로 설정될 수 있고, 모든 스마트컨트랙트에 일괄적으로 적용되어 설정될 수도 있다.

[0049] 단계 S240이 수행되면, 가상머신은 콜스택을 분석하여 호출 주체를 찾는다(S250). 콜 스택의 각 스택에는 호출 함수를 가지는 스마트 컨트랙트의 식별정보가 함께 저장되므로, 가상머신은 콜스택에 저장된 각 스마트 컨트랙트 식별정보를 호출 주체로 찾을 수 있다. 이때, 가상머신은 스택 트레이스상에 나타나는 call chain의 스마트 컨트랙트 식별정보들의 리스트를 가공하여 호출 시점의 크리덴셜(credential)로 생성하고, 호출시점의 크리덴셜을 이용하여 호출 주체의 권한 여부를 판단할 수 있다.

[0050] 단계 S250의 실행 후, 가상머신은 찾은 호출 주체가 해당 함수를 호출할 권한이 있는지 판단한다(S260). 즉, 가상머신은 해당 함수의 스마트 컨트랙트에 설정된 보안정책에 기초하여, 호출 시점의 크리덴셜에 포함된 호출 주체가 해당 함수를 호출할 권한이 있는지 판단할 수 있다. 다시 말하면, 가상머신은 호출시점의 크리덴셜을 해당 스마트 컨트랙트에 설정된 보안정책과 비교하여 호출 주체가 호출 권한이 있는지를 판단할 수 있다.

[0051] 단계 S260의 판단결과, 호출 주체가 해당 함수를 호출할 권한이 있는 경우 가상머신은 해당 함수를 실행시키고(S270), 호출 권한이 없는 경우 해당 함수의 실행을 중지하고 오류 발생을 관리자 또는 개발자에게 알린다(S280).

[0052] 상술한 바와 같이 가상머신이 각 스마트 컨트랙트에 설정된 보안정책에 기초하여, 호출 주체의 함수 호출 권한 여부를 판단하고, 그 판단결과에 따라 함수를 실행시키는 방법에 대해 도 3 및 도 4를 참조하여 설명하기로 한다.

[0053] 도 3은 스마트 컨트랙트 sc1의 함수 A에서 스마트 컨트랙트 sc2의 함수 B가 호출되고, sc2의 함수 B에서 스마트 컨트랙트 sc3의 함수 C가 호출되는 경우를 나타낸다. 이때, 스마트 컨트랙트 sc1, sc2, sc3는 서로 다르거나 동일할 수 있다. 또한, 스마트 컨트랙트 sc1의 계정 ID는 sid1, 배포자 ID는 did1, 생성자 ID는 cid1이고, 스마트 컨트랙트 sc2의 계정 ID는 sid2, 배포자 ID는 did2, 생성자 ID는 cid2이며, 스마트 컨트랙트 sc3의 계정 ID는 sid3, 배포자 ID는 did3, 생성자 ID는 cid3이다. 그리고, 스마트 컨트랙트 sc1에는 "스마트 컨트랙트 sc4는 함수 A 호출 불가"라는 보안정책이 설정되고, 스마트 컨트랙트 sc2에는 "스마트 컨트랙트 sc7는 함수 B 호출 불가"라는 보안정책이 설정되어 있으며, 스마트 컨트랙트 sc3에는 "스마트 컨트랙트 sc1은 함수 C 호출 불가"라는 보안정책이 설정되었다고 가정하여 설명하기로 한다.

[0054] 먼저, 도 3과 같이 sc1의 함수 A가 호출되면, 가상머신은 sc1의 함수 A에 대한 정보를 콜스택에 저장한다. 이때, 가상머신은 도 4의 (a)와 같이 콜 스택에 SC1에 대한 정보를 저장하고, SC1에 대한 정보에는 SC1 식별정보인 F<sid1, did1, cid1>을 포함할 수 있다.

[0055] 다음으로, sc1의 함수 A에서 sc2의 함수 B가 호출되면, 가상머신은 sc2의 함수 B에 대한 정보를 콜스택에 저장한다. 이때, 가상머신은 도 4의 (b)와 같이 콜 스택에 SC2에 대한 정보를 저장하고, SC2에 대한 정보에는 SC2 식별정보인 F<sid2, did2, cid2>를 포함한다. 따라서, 가상머신은 스택 트레이스상에 나타나는 call chain의 SC1 식별정보와 SC2 식별정보를 가공하여, sc2의 함수 B 호출 시점의 크리덴셜(credential)을 생성한다. 이때 생성된 크리덴셜은 sc2의 함수 B의 호출 주체로, 예컨대, "credential = G(F<sid1, did1, cid1>, F<sid2, did2, cid2>)"로 표현될 수 있다. 가상머신은 sc2의 함수 B 호출 시점의 크리덴셜을 이용하여 호출 주체가 sc2의 함수 B의 호출 권한이 있는지를 판단한다. 즉, sc2에는 "스마트 컨트랙트 sc7는 함수 B 호출 불가"라는 보안정책이 설정되어 있으므로, 가상머신은 함수 B 호출 시점의 크리덴셜에 포함된 호출 주체들 중 sc2의 보안정책에 반하는 주체가 있는지를 판단하여, 함수 B 호출 권한 소유 여부를 판단할 수 있다. 함수 B 호출 시점의 크리덴셜에는 호출 권한이 있는 주체만 있으므로, 가상머신은 함수 B를 실행시킨다.

[0056] 마지막으로, sc2의 함수 B에서 sc3의 함수 C가 호출되면, 가상머신은 sc3의 함수 C에 대한 정보를 콜스택에 저

장한다. 이때, 가상머신은 도 4의 (c)와 같이 콜 스택에 SC3에 대한 정보를 저장하고, SC3에 대한 정보에는 SC3 식별정보인 $F<sid3, did3, cid3>$ 를 포함한다. 따라서, 가상머신은 스택 트레이스상에 나타나는 call chain 의 SC1 식별정보, SC2 식별정보 및 SC3 식별정보를 가공하여 sc3의 함수 C 호출 시점의 크리덴셜(credential)을 생성한다. 이때 생성된 크리덴셜은 sc3의 함수 C의 호출 주체로, 예컨대, "credential = G($F<sid1, did1, cid1>$, $F<sid2, did2, cid2>$, $F<sid3, did3, cid3>$)"로 표현될 수 있다. 가상머신은 sc3의 함수 C 호출 시점의 크리덴셜을 이용하여 호출 주체가 sc3의 함수 C의 호출 권한이 있는지를 판단한다. 즉, sc3에는 "스마트 컨트랙트 sc1은 함수 C 호출 불가"라는 보안정책이 설정되어 있으므로, 가상머신은 함수 C 호출 시점의 크리덴셜에 포함된 호출 주체들 중 sc3의 보안정책에 반하는 주체가 있는지를 판단하여, 함수 C 호출 권한 소유 여부를 판단할 수 있다. 함수 C 호출 시점의 크리덴셜에는 호출 권한이 없는 " $F<sid1, did1, cid1>$ "가 있으므로, 가상머신은 호출 권한이 없다고 판단하여 실행을 중지시킨다.

[0058]

한편, 본 발명의 실시예들은 다양한 컴퓨터 수단을 통하여 수행될 수 있는 프로그램 명령 형태로 구현되어 컴퓨터 판독 가능 매체에 기록될 수 있다. 상기 컴퓨터 판독 가능 매체는 프로그램 명령, 데이터 파일, 데이터 구조 등을 단독으로 또는 조합하여 포함할 수 있다. 상기 매체에 기록되는 프로그램 명령은 본 발명을 위하여 특별히 설계되고 구성된 것들이거나 컴퓨터 소프트웨어 당업자에게 공지되어 사용 가능한 것일 수도 있다. 컴퓨터 판독 가능 기록 매체의 예에는 하드 디스크, 플로피 디스크 및 자기 테이프와 같은 자기 매체(magnetic media), CD-ROM, DVD와 같은 광기록 매체(optical media), 플롭티컬 디스크(floptical disk)와 같은 자기-광매체(magneto-optical), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령의 예에는 컴파일러에 의해 만들어지는 것과 같은 기계어 코드뿐만 아니라 인터프리터 등을 사용해서 컴퓨터에 의해서 실행될 수 있는 고급 언어 코드를 포함한다. 상기된 하드웨어 장치는 본 발명의 일 실시예들의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

[0059]

이제까지 본 발명에 대하여 그 바람직한 실시예들을 중심으로 살펴보았다. 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자는 본 발명이 본 발명의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로 개시된 실시예들은 한정적인 관점이 아니라 설명적인 관점에서 고려되어야 한다. 본 발명의 범위는 전술한 설명이 아니라 특허청구범위에 나타나 있으며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명에 포함된 것으로 해석되어야 할 것이다.

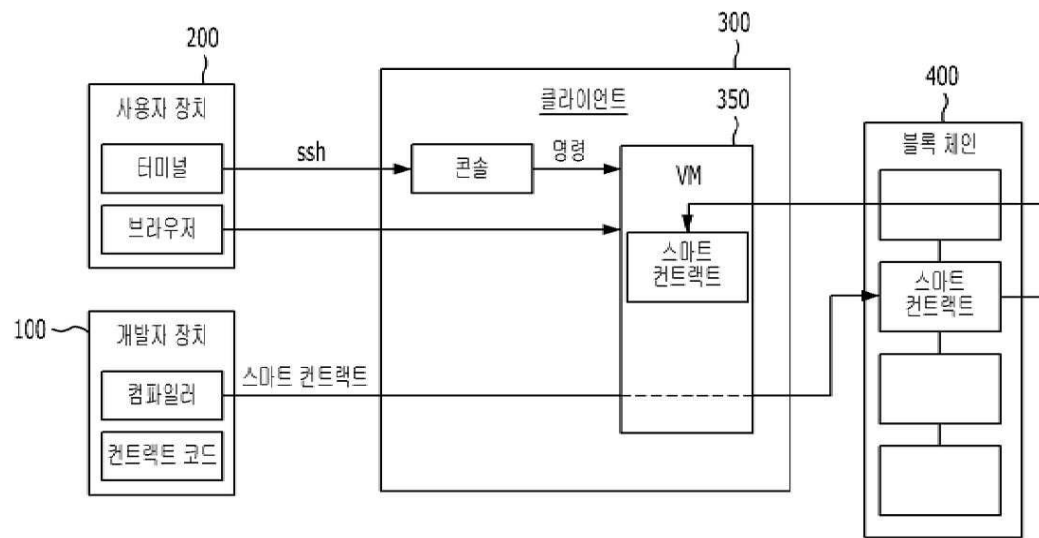
부호의 설명

[0061]

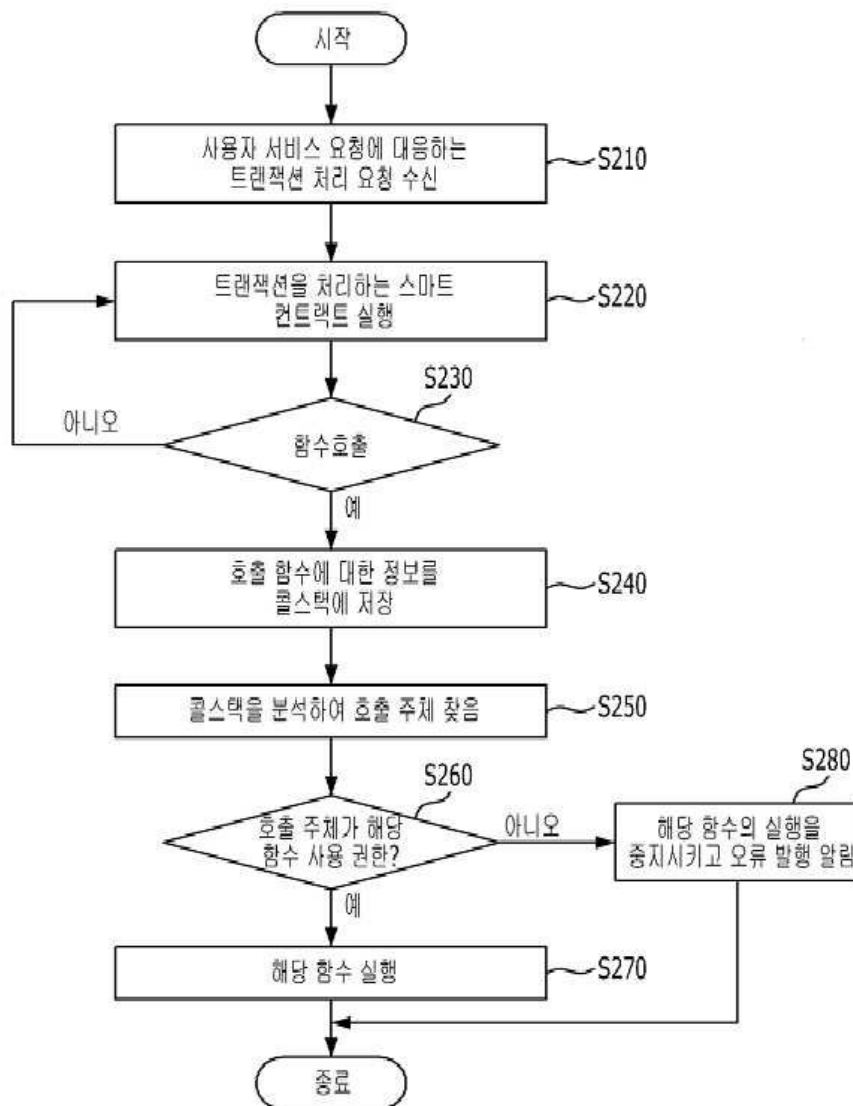
100 : 개발자 장치
200 : 사용자 장치
300 : 클라이언트
350 : 가상머신
400 : 블록체인

도면

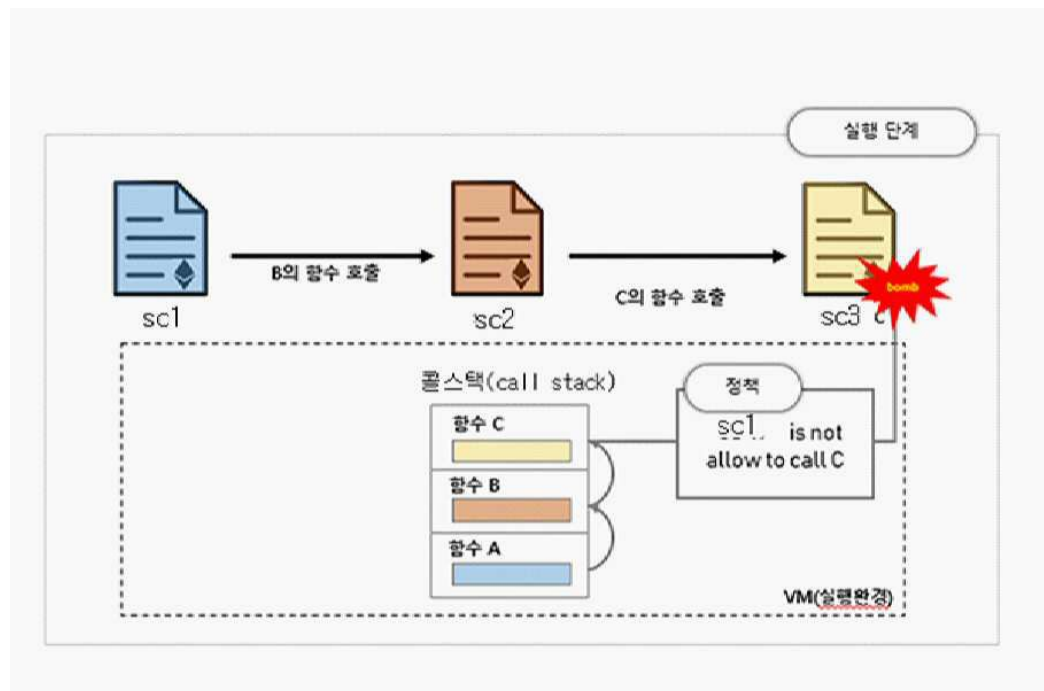
도면1



도면2



도면3



도면4

