(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2021년06월14일
(11) 등록번호 10-2263692
(24) 등록일자 2021년06월04일

(51) 국제특허분류(Int. Cl.)
G06F 8/41 (2018.01) *G06F 21/44* (2013.01)
G06F 21/52 (2013.01) *G06Q 10/10* (2012.01)
G06Q 20/38 (2012.01)
(52) CPC특허분류
G06F 8/447 (2013.01)
G06F 21/445 (2013.01)
(21) 출원번호 10-2019-0177987
(22) 출원일자 2019년12월30일
심사청구일자 2019년12월30일
(56) 선행기술조사문헌
KR100529416 B1*
KR101878869 B1*
KR1020190132938 A*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
충남대학교 산학협력단
대전광역시 유성구 대학로 99 (궁동, 충남대학교)
(72) 발명자
조은선
대전광역시 유성구 어은로 57, 110동 1504호
류재철
대전광역시 유성구 어은로 57 132동 801호 (어은동,한빛아파트)
(74) 대리인
이은철, 김재문

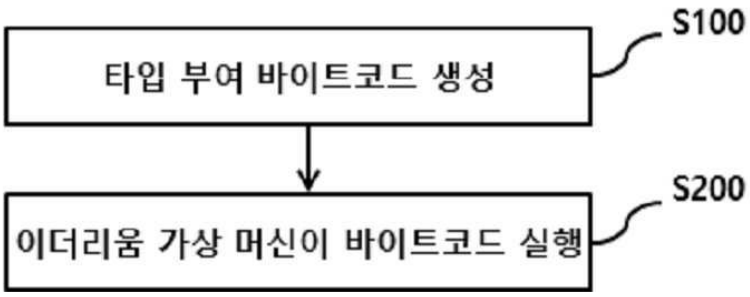
전체 청구항 수 : 총 2 항 심사관 : 지정훈

(54) 발명의 명칭 **이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법**

(57) 요 약

본 기술은 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법에 관한 것이다. 본 기술의 구현 예에 따르면, 이더리움 스마트계약 바이트 코드에 타입을 부여함으로써 바이트 코드의 보안성을 향상시킬 수 있는 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법을 제공할 수 있는 이점이 있다.

대 표 도 - 도2



(52) CPC특허분류
G06F 21/52 (2013.01)
G06Q 10/10 (2013.01)
G06Q 20/382 (2013.01)

이 발명을 지원한 국가연구개발사업

과제고유번호	2018000251
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	정보보호핵심원천기술개발(R&D)
연구과제명	스마트 컨트랙트 프라이버시 보호 및 취약점 분석 기술 개발
기 여 율	1/1
과제수행기관명	충남대학교산학협력단
연구기간	2019.01.01 ~ 2019.12.31

명 세 서

청구범위

청구항 1

이더리움 가상 머신 기반 바이트코드 생성 및 실행에 있어서,

컴파일러가 솔리디티 코드로부터 타입이 부여된 바이트코드를 생성하는 단계; 및

이더리움 가상 머신이 상기 타입이 부여된 바이트코드를 실행하여 스마트 계약을 수행하는 단계를 포함하고,

상기 타입은 데이터의 크기, 데이터의 종류, 데이터의 성격 중 적어도 하나를 포함하며,

상기 가상 머신 기반 바이트코드의 보안성을 향상하기 위해 상기 컴파일러는 디코딩이 필요한 스택 엔트리에 대해 타입 규칙을 적용하는 것을 특징으로 하는 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법.

청구항 2

이더리움 가상 머신 기반 바이트코드 실행에 있어서,

컴파일러가 솔리디티 코드로부터 바이트코드를 생성하는 단계; 및

타입 검사를 하는 이더리움 가상 머신이 상기 바이트코드를 실행하여 스마트 계약을 수행하는 단계를 포함하고,

상기 타입은 데이터의 크기, 데이터의 종류, 데이터의 성격 중 적어도 하나를 포함하며,

상기 가상 머신 기반 바이트코드의 보안성을 향상하기 위해 상기 컴파일러는 디코딩이 필요한 스택 엔트리에 대해 타입 규칙을 적용하는 것을 특징으로 하는 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법.

청구항 3

삭제

발명의 설명

기 술 분 야

[0001] 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법에 관한 것으로, 더욱 상세하게는 타입을 부여하여 이더리움 스마트 계약 바이트 코드의 보안성을 향상하는 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법에 관한 것이다.

배 경 기 술

[0003] 이더리움과 같은 블록체인 2.0은 계약에 프로그램이 들어가 있는 형태인 스마트 계약으로 구성되는 블록체인 플랫폼이다. 이더리움 스마트 계약 코드를 수행하는 가상머신인 EVM(Etherium Virtual Machine)은 저수준의 바이트코드를 수행한다. 바이트 코드는 고수준의 솔리디티(Solidity) 언어를 solc 컴파일러를 통해 변환시킨 것인데, 기존의 솔리디티나 solc 컴파일러는 보안성이 부족하다.

[0004] 따라서 기존의 이더리움 스마트 계약 코드의 보안성을 향상시키기 위한 방법을 도입할 필요성이 있다.

선행기술문헌

특허문헌

[0006] (특허문헌 0001) 1. 한국공개특허공보 10-2019-0105774

발명의 내용

해결하려는 과제

[0007] 본 발명은 상기와 같은 문제를 해결하기 위한 것으로서, 이더리움 바이트코드의 보안성을 향상시킬 수 있는 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법을 제공하는데 그 목적이 있다.

과제의 해결 수단

[0009] 상기와 같은 목적을 달성하기 위한 본 발명은 이더리움 가상 머신 기반 바이트코드 생성 및 실행에 있어서, 컴파일러가 솔리디티 코드로부터 타입이 부여된 바이트코드를 생성하는 단계; 및 이더리움 가상 머신이 상기 타입이 부여된 바이트코드를 실행하여 스마트 계약을 수행하는 단계를 포함하는 것을 특징으로 한다.

[0010] 바람직하게는, 상기 컴파일러는 디코딩이 필요한 스택 엔트리에 대해 타입 규칙을 적용하는 것이다.

[0011] 다른 실시예에 따른 본 발명은 이더리움 가상 머신 기반 바이트코드 생성 및 실행에 있어서, 컴파일러가 솔리디티 코드로부터 바이트코드를 생성하는 단계; 및 타입 검사를 하는 이더리움 가상 머신이 상기 바이트코드를 실행하여 스마트 계약을 수행하는 단계를 포함하는 것을 특징으로 한다.

[0012] 바람직하게는, 상기 컴파일러는 디코딩이 필요한 스택 엔트리에 대해 타입 규칙을 적용하는 것이다.

발명의 효과

[0014] 전술한 바와 같은 본 발명에 따르면, 스마트계약을 수행하는 이더리움 바이트코드에 타입을 부여하여 보안성을 향상시킬 수 있는 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법을 제공할 수 있는 이점이 있다.

도면의 간단한 설명

[0016] 도 1은 컴파일러를 개선하여 타입 시스템을 지원하는 방법의 개념도를 도시한다.

도 2는 본 발명의 실시예에 따른 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법의 흐름도를 도시한다.

도 3은 컴파일러의 도움 없이 타입 시스템을 지원하는 방법의 개념도를 도시한다.

도 4는 본 발명의 다른 실시예에 따른 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법의 흐름도를 도시한다.

발명을 실시하기 위한 구체적인 내용

[0017] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 후술되어 있는 실시예들을 참조하면 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 이에 앞서 본 발명에 관련된 공지 기능 및 그 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 구체적인 설명을 생략하였음에 유의해야 할 것이다.

[0018] 이더리움 바이트 코드는 명령과 인자를 가지며 스택을 가진다는 측면이 자바 가상머신과 유사하다. 하지만 다음과 같은 점들에서 차이가 있다.

[0019] (1) 스택: 스택의 한 칸은 바이트가 아니라 32 byte로 상당히 길다. 따라서 원하는 내용을 얻기위해서는 32byte에 저장되어 있는 내용을 파싱하여 추출하는 경우가 있다. 즉, 인코딩된 정보를 디코딩해서 쓰는 경우인데, 예를 들어 데이터는 메소드 id를 나타내는 해쉬 코드와 함께 여러 가지가 같이 인코딩되어 있고 이중 몇 바이트만 함수 시그니처가 되어 함수 호출을 위해 사용된다.

[0020] (2) 지속성 데이터: Storage라는 특수한 데이터가 있어 이 데이터는 저장장치에 저장되는 것을 가정하며, 이 storage는 map으로 이용되며 동시에 프로그램 내에 일반 변수처럼 접근된다. 갱신은 지속성을 가지고 반영된다.

[0021] (3) 함수 호출: selector를 사용하여 함수가 불리며 함수 호출시 인자 전달을 가상머신 코드에서 직접 지원해야 한다. 한 계약에 여러 함수가 있는데, 각 함수를 호출하기 위해서는 계약 시작부분에 존재하는 selector문(switch 문과 유사함) 을 통해 호출된다. 즉, 함수 호출이 추상화되어있는 Java 바이트코드에 비해 매우 저수준이다.

[0022] (4) 수행을 위해 처음 프로그램이 로딩될 때 relocation이 되므로 바이트코드 텍스트를 디코딩 했을 때 분기 타겟 주소를 파악하는데에 직관적이지 않다.

[0023] 이와 같은 측면은 EVM 바이트 코드가 매우 저수준임을 알 수있게 해준다. 그러나 솔리디티 컴파일러인 solc를 제작하는 측에서 이를 모두 고려하여 완벽하게 만들기란 쉽지 않으며, 특히 계속 버전업을 해나가는 상황에서 품질을 유지하기란 더욱 쉽지 않다.

[0024] 본 발명은 이더리움 컴파일러 solc의 보안이 안전하지 못하거나 안전성이 부족할 경우 가상머신 코드 단계에서 보안성을 보완해줄 수 있는 타입시스템을 제시한다.

[0025] 예를 들어, 개인정보가 유출되는 것을 막기위한 타입시스템과 calldata와 같이 디코딩이 필요한 스택 엔트리에 대해 타입 규칙을 적용하여 가상머신 코드 단계에서 보안성을 향상할 수 있다.

[0026] 타입시스템을 지원하는 방법에는 컴파일러를 개선하여 도움을 받는 방법과 컴파일러의 도움 없이 타입 추론으로 타입을 지원하는 방법이 있다.

[0027] 도 1은 컴파일러를 개선하여 타입 시스템을 지원하는 방법의 개념도를 도시한다.

[0028] 도 1을 참조하면, 왼쪽은 일반 솔리디티 컴파일러로 컴파일된 바이트코드를 보여주며 오른쪽은 컴파일러가 보안성이 향상된 바이트코드를 생성하여 이더리움 가상머신에 정보를 전달하는 것을 보여준다.

[0029] 도 2는 본 발명의 실시예에 따른 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법의 흐름도를 도시한다.

[0030] 도 2를 참조하면, 먼저, 컴파일러가 솔리디티 코드로부터 타입이 부여된 스마트 계약 바이트 코드를 생성한다(S100). 이어서, 이더리움 가상 머신이 타입이 부여된 바이트 코드를 실행하여 스마트 계약을 수행한다(S200).

[0031] 이때, 컴파일러는 디코딩이 필요한 스택 엔트리에 대해 타입 규칙을 적용하는 것일 수 있다. 그러나 바이트코드의 타입 검사를 위해 컴파일러의 도움을 받는 것은 가급적 지양할 필요가 있다.

[0032] 도 3은 컴파일러의 도움 없이 타입 시스템을 지원하는 방법의 개념도를 도시한다.

[0033] 도 3을 참조하면, 이더리움 가상 머신은 일반적인 솔리디티 컴파일러로 만들어진 바이트 코드에 대해서 타입 검사를 지원한다. 이를 위하여 타입을 지원하는 이더리움 가상머신 바이트코드 언어를 필요로 한다.

[0034] 이더리움 가상 머신의 타입 검사는 수행전 검증과 런타임 타입 검사 두 가지가 가능하다. 검증은 논리적 검증 도구로 수행될 수 있고, 런타임 타입 검사를 위해서는 별도의 스택 등의 추가 메모리와 모니터링이 수반된다.

[0035] 도 4는 본 발명의 다른 실시예에 따른 이더리움 스마트계약 가상머신 바이트코드 보안성 향상 방법의 흐름도를 도시한다.

[0036] 도 4를 참조하면, 먼저, 컴파일러가 솔리디티 코드로부터 바이트 코드를 생성한다(S110). 이어서, 타입 검사를 하는 이더리움 가상 머신이 상기 스마트 계약 바이트 코드를 실행하여 스마트 계약을 수행한다(S220).

[0037] 이때, 컴파일러는 디코딩이 필요한 스택 엔트리에 대해 타입 규칙을 적용하는 것일 수 있다.

[0038] 본 발명에 따라 지원되는 보안성 향상을 위한 타입 시스템은 다음과 같이 이루어질 수 있다.

[0039] (1) 타입의 내용

[0040] 데이터의 크기, 데이터의 종류, 데이터의 성격에 대한 내용이 타입이 된다. 스택에 대한 각종 CURD (create/update/read/write) 연산에 대해 확장한다.

[0041] (2) 배치(align)의 적정성

[0042] 각 타입에서 포함하고 있는 크기 정보를 활용하여 여러 기억장치에 대해 배치의 적정성을 지원하여 일관성(Integrity)를 유지하도록 한다.

[0043] (3) 타입의 추론

[0044] 타입 규칙을 정의하여, 각 타입이 프로그램의 다른 부분에 미치는 영향을 추가적으로 규명해나간다. 이 과정은 타입 검증 과정에 주로 포함되나 경우에 따라서는 타입 정의 과정에도 포함된다.

[0045] (4) 타입 검증

[0046] 개인정보의 유출여부, 지속성을 가진 데이터의 사용 형태, 데이터 접근 및 제어 흐름의 일관성(integrity)를 유지하는지 여부 등이 타입으로 검사된다.

[0047] 스마트 계약은 금전적인 거래가 가능한 응용 프로그램이므로 신뢰성 보안성이 중요하다.

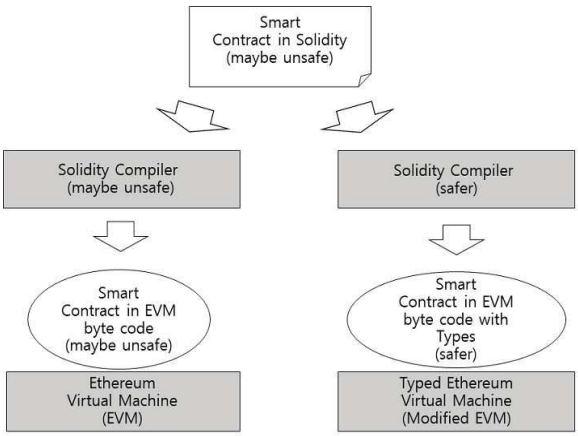
[0048] 본 발명에 따라 이더리움 가상머신 바이트코드에 타입을 부여하고 타입 검사를 실행함으로써 이더리움 스마트 계약의 신뢰성과 안전성을 높일 수 있다.

[0049] 본 발명의 실시예에 따른 방법은 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 실시예를 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-광 매체(magneto-optical media), 및 롬(ROM), 램(RAM), 플래시 메모리 등과 같은 프로그램 명령을 저장하고 수행하도록 특별히 구성된 하드웨어 장치가 포함된다. 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 실시예의 동작을 수행하기 위해 하나 이상의 소프트웨어 모듈로서 작동하도록 구성될 수 있으며, 그 역도 마찬가지이다.

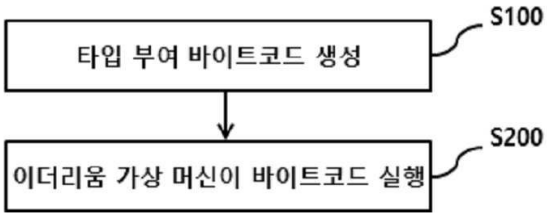
[0050] 이상으로 본 발명의 기술적 사상을 예시하기 위한 바람직한 실시예와 관련하여 설명하고 도시하였지만, 본 발명은 이와 같이 도시되고 설명된 그대로의 구성 및 작용에만 국한되는 것이 아니며, 기술적 사상의 범주를 일탈함이 없이 본 발명에 대해 다수의 변경 및 수정이 가능함을 잘 이해할 수 있을 것이다. 따라서, 그러한 모든 적절한 변경 및 수정과 균등물들도 본 발명의 범위에 속하는 것으로 간주되어야 할 것이다.

도면

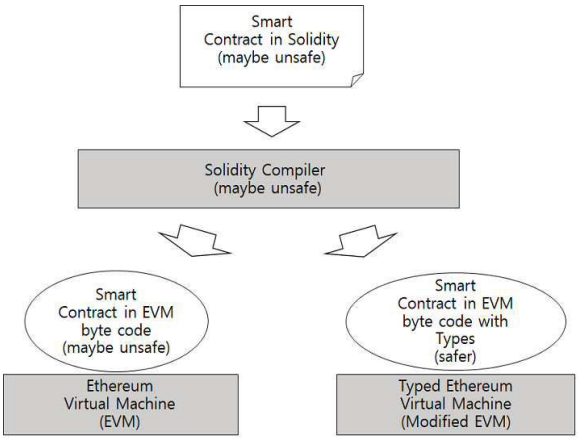
도면1



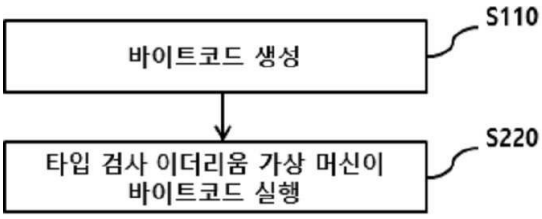
도면2


타입 부여 바이트코드 생성
S100
이더리움 가상 머신이 바이트코드 실행
S200

도면3


Smart
Contract in Solidity
(maybe unsafe)
Solidity Compiler
(maybe unsafe)
Smart
Contract in EVM
byte code
(maybe unsafe)
Ethereum
Virtual Machine
(EVM)
Smart
Contract in EVM
byte code with
Types
(safer)
Typed Ethereum
Virtual Machine
(Modified EVM)

도면4


바이트코드 생성
S110
타입 검사 이더리움 가상 머신이
바이트코드 실행
S220