(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2024년07월24일
(11) 등록번호 10-2687781
(24) 등록일자 2024년07월19일

(51) 국제특허분류(Int. Cl.)
H04L 9/32 (2006.01) *H04L 9/40* (2022.01)
(52) CPC특허분류
H04L 9/3239 (2013.01)
H04L 63/12 (2013.01)
(21) 출원번호 10-2020-7018346
(22) 출원일자(국제) 2018년12월12일
심사청구일자 2021년12월07일
(85) 번역문제출일자 2020년06월24일
(65) 공개번호 10-2020-0096790
(43) 공개일자 2020년08월13일
(86) 국제출원번호 PCT/IB2018/059920
(87) 국제공개번호 WO 2019/116248
국제공개일자 2019년06월20일
(30) 우선권주장
1720946.1 2017년12월15일 영국(GB)
(56) 선행기술조사문헌
JP2013506369 A
JP2016521403 A
WO2016155804 A1

(73) 특허권자
엔체인 홀딩스 리미티드
안티구아바부다 세인트존스, 처치 스트리트 44, 피츠제럴드 하우스
(72) 발명자
코바시 알렉산드라
영국 카디프 씨에프10 2에이치에이치 처칠 웨이 처칠 하우스 7 플로어 어커트-다이크 앤드 로드 엘엘피
마데오 시몬
영국 카디프 씨에프10 2에이치에이치 처칠 웨이 처칠 하우스 7 플로어 어커트-다이크 앤드 로드 엘엘피
(뒷면에 계속)
(74) 대리인
제일특허법인(유)

전체 청구항 수 : 총 15 항 심사관 : 양종필

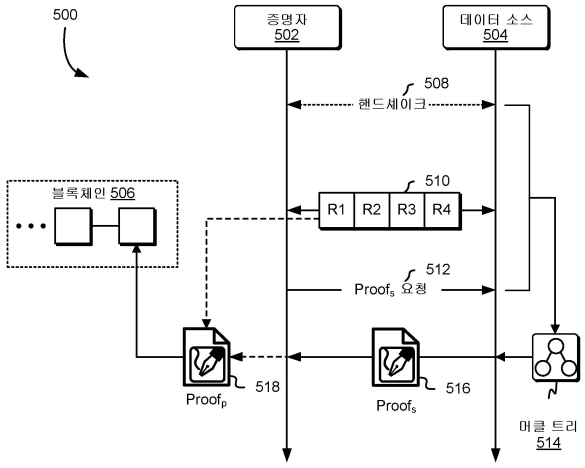
(54) 발명의 명칭 **증명 검증에 기초하여 오프-체인 데이터를 인증하기 위한 시스템 및 방법**

(57) 요 약

블록체인 네트워크에 게시된 프로그램 또는 스크립트를 실행하는 시기 및/또는 방법을 결정하는 것은 실세계 상태 및 이벤트에 관한 데이터와 같은 블록체인 외부의 데이터(즉, 외부 데이터)에 의존할 수 있다. 증명자(예를 들어, 블록체인 네트워크의 노드)는 클라이언트를 대신하여 프로그램의 실행과 같은 하나 이상의 계산을 수행할

(뒷면에 계속)

대 표 도



수 있다. 프로그램을 적절하게 실행하기 위해, 증명자는, 증명자가 클라이언트와 신뢰 관계가 있는 데이터 공급자로부터 획득할 수 있는 (예를 들어, 클라이언트가 신뢰할 수 있는 데이터 공급자에 의해 제공된 유효한 데이터로서 수락하는) 외부 데이터에 의존할 수 있다. 본 명세서에 설명된 시스템 및 방법은 증명자에 의해 데이터 제공자로부터 의도적으로 획득된 입력 데이터의 진위성에 관한 암호로 검증 가능한 보증을 제공하는데 이용될 수 있으며, 입력 데이터는 블록체인 네트워크에 게시된 프로그램 또는 스크립트의 실행에 이용된다. 블록체인 네트워크의 예는 비트코인 기반 네트워크이다.

(52) CPC특허분류

H04L 63/166 (2013.01)

H04L 9/50 (2022.05)

(72) 발명자

모틸린스키 패트릭

영국 카디프 씨에프10 2에이치에이치 처칠 웨이 처칠 하우스 7 플로어 어커트-다이크 앤드 로드 엘엘피

빈센트 스테판

영국 카디프 씨에프10 2에이치에이치 처칠 웨이 처칠 하우스 7 플로어 어커트-다이크 앤드 로드 엘엘피

명 세 서

청구범위

청구항 1

컴퓨터 구현된 방법으로서,

컴퓨팅 엔티티와 암호로 보호된 통신 세션을 수립하는 단계와,

상기 암호로 보호된 통신 세션을 통해, 블록체인 네트워크에 게시된 프로그램의 실행을 제어하는 입력 데이터를 포함하는 제 1 통신을 수신하는 단계와,

상기 암호로 보호된 통신 세션을 통해 통신 세트가 발생했다는 제 1 입증(attestation)을 수신하는 단계 - 상기 통신 세트는 상기 제 1 통신을 포함함 - 와,

상기 입력 데이터를 사용하여 상기 프로그램을 실행하는 단계 - 상기 프로그램의 실행은 상기 프로그램의 올바른 실행의 증명(proof)을 생성함 - 와,

상기 입력 데이터에 적어도 부분적으로 기초하여, 상기 입력 데이터가 상기 제 1 통신에 포함되었다는 제 2 입증을 생성하는 단계와,

상기 프로그램의 올바른 실행의 증명을 다른 컴퓨터 시스템에 제공하는 단계를 포함하는

컴퓨터 구현된 방법.

청구항 2

제 1 항에 있어서,

상기 제 1 입증은 머클 트리(Merkle tree)의 루트 노드(root node)에 적어도 부분적으로 기초한 값을 가지며, 상기 머클 트리는 상기 통신 세트로부터 결정된 한 세트의 리프 노드(leaf node) 및 한 세트의 솔트 값(salt value)을 포함하는

컴퓨터 구현된 방법.

청구항 3

제 2 항에 있어서,

상기 통신 세트의 각각의 통신은 상기 통신이 수신되었는지 또는 송신되었는지에 기초하여 결정된 대응하는 중간 노드를 갖는

컴퓨터 구현된 방법.

청구항 4

제 2 항에 있어서,

상기 제 1 입증의 값은 또한, 적어도 상기 머클 트리의 상기 루트 노드로부터 생성된 암호 해시 출력 및 상기 통신 세트의 시간 간격에 적어도 부분적으로 기초하는

컴퓨터 구현된 방법.

청구항 5

제 2 항에 있어서,

상기 제 2 입증은 상기 머클 트리의 머클 경로에 적어도 부분적으로 기초하고, 상기 머클 경로는 상기 머클 트리의 노드 세트의 값을 포함하고, 상기 노드 세트의 값은 상기 머클 트리의 상기 루트 노드 값을 계산하기에 충분한

컴퓨터 구현된 방법.

청구항 6

제 5 항에 있어서,

상기 머클 경로의 상기 노드 세트는 상기 머클 트리의 각각의 비 리프(non-leaf) 및 비 루트(non-root) 깊이에서 정확히 하나의 노드를 포함하는

컴퓨터 구현된 방법.

청구항 7

제 1 항에 있어서,

상기 프로그램은 2 이상의 당사자(party)에 의해 합의된 룰 세트를 포함하고, 상기 방법은 상기 2 이상의 당사자 중 적어도 하나의 당사자에 의해 신뢰되는 하나 이상의 컴퓨팅 엔티티로부터 상기 컴퓨팅 엔티티를 선택하는 단계를 더 포함하는

컴퓨터 구현된 방법.

청구항 8

제 1 항에 있어서,

블록체인 트랜잭션을 검출하는 단계 - 상기 블록체인 트랜잭션은,

제 1 잠금 스크립트(locking script)를 포함하는 제 1 트랜잭션 출력, 및 상기 올바른 실행의 증명이 유효하다는 표시를 인코딩하는 제 2 트랜잭션 출력을 포함하고, 상기 제 1 트랜잭션 출력과 연관된 제 1 디지털 자산은 잠금해제 스크립트(unlocking script)에 의해 잠금해제 가능하며, 상기 잠금해제 스크립트는,

상기 컴퓨팅 엔티티와 연관된 공개 키;

기대 값을 인코딩하는 디지털 서명; 및

상기 기대 값을 생성하는데 사용 가능한 인증 정보를 인코딩하고, 상기 디지털 서명의 진위성(authenticity)은 상기 공개 키를 사용하여 암호로 검증 가능함 - 와,

적어도 상기 공개 키, 상기 디지털 서명 및 상기 인증 정보를 제공함으로써 상기 제 1 디지털 자산을 잠금해제하는 단계를 더 포함하는

컴퓨터 구현된 방법.

청구항 9

제 8 항에 있어서,

상기 블록체인 트랜잭션은,

상기 다른 컴퓨터 시스템과 연관된 개인 키를 사용하여 디지털 서명된 트랜잭션 입력과,

제 2 잠금해제 스크립트를 포함하는 제 3 트랜잭션 출력 - 상기 제 3 트랜잭션 출력과 연관된 제 2 디지털 자산은 상기 개인 키를 사용하여 잠금해제될 수 있음 - 을 더 포함하며,

상기 제 2 트랜잭션 출력은 또한 상기 다른 컴퓨터 시스템과 연관된 식별자를 인코딩하는

컴퓨터 구현된 방법.

청구항 10

제 8 항에 있어서,

상기 인증 정보는 머클 트리의 머클 경로를 포함하고 상기 기대 값은 상기 머클 트리의 루트 노드에 적어도 부분적으로 기초하는

컴퓨터 구현된 방법.

청구항 11

제 1 항에 있어서,

상기 입력 데이터는 이벤트가 발생했는지를 표시하는 이진 데이터를 포함하는

컴퓨터 구현된 방법.

청구항 12

제 1 항에 있어서,

상기 입력 데이터는 블록체인 상의 다른 데이터에 기초하여 검증할 수 없는 정보를 포함하는 데이터인

컴퓨터 구현된 방법.

청구항 13

제 1 항에 있어서,

상기 제 1 입증은 디지털 서명이며, 상기 디지털 서명의 진위성은 상기 컴퓨팅 엔티티와 연관된 암호화 공개 키를 사용하여 검증 가능한

컴퓨터 구현된 방법.

청구항 14

시스템으로서,

프로세서와,

상기 프로세서에 의한 실행의 결과로서, 상기 시스템으로 하여금 제 1 항 내지 제 13 항 중 어느 한 항에 따른 컴퓨터 구현된 방법을 수행하게 하는 실행 가능한 명령어를 포함하는 메모리를 포함하는

시스템.

청구항 15

컴퓨터 시스템의 프로세서에 의해 실행되는 결과로서, 상기 컴퓨터 시스템으로 하여금 적어도, 제 1 항 내지 제 13 항 중 어느 한 항에 따른 컴퓨터 구현된 방법을 수행하게 하는 실행 가능한 명령어가 저장된 비일시적 컴퓨터 판독 가능 저장 매체.

발명의 설명

기 술 분 야

[0001] 본 발명은 일반적으로 블록체인 기술에 관한 것으로, 보다 상세하게는 비트코인 스크립트(Bitcoin script)에서 검증될 수 있는, 또한 상환 청구권(recourse)의 사례에서 추가로 이용될 수 있는 정보를 제공하는, 대화의 증명의 생성에 관한 것이다. 대화 중에 특정 통신이 발생했다는 증명은 머클 트리(Merkle trees)를 이용할 수 있다. 대화 또는 통신의 데이터는 입력으로서 프로그램 또는 스크립트를 올바르게 실행하는데 이용될 수 있다. 본 발명은 이것으로 제한되는 것은 아니지만 스마트 계약 생성 및 실행에서의 사용에 특히 적합하다.

배 경 기 술

[0002] 본 문서에서 '블록체인(blockchain)'이라는 용어는 여러 유형 중 임의의 전자 컴퓨터 기반의 분산 원장(distributed ledger)을 지칭할 수 있다. 이것은 합의 기반의 블록체인(consensus-based blockchain) 및 트랜잭션 체인(transaction-chain) 기술, 허가된(permissioned) 및 허가되지 않은(unpermissioned) 원장, 공유된 원장(shared ledger) 및 이들의 변형을 포함한다. 다른 블록체인 구현예가 제안되고 개발되었지만, 가장 널리 알려진 블록체인 기술의 애플리케이션은 비트코인 원장(Bitcoin ledger)이다. 편의성 및 예시의 목적을 위해 본 개시내용에서 설명되는 기술의 유용한 애플리케이션으로서 비트코인이 언급될 수 있지만, 비트코인은 본 개

시내용에서 설명되는 기술이 적용될 수 있는 많은 애플리케이션 중 하나에 불과하다. 그러나, 본 발명은 비트코인 블록체인과 함께 사용되는 것으로 제한되지 않는다는 것과, 비상업적 애플리케이션을 비롯한 대안적인 블록체인 구현 및 프로토콜이 또한 본 발명의 범위에 속한다는 것을 유의하여야 한다. 예를 들어, 본 개시내용 내에서 설명된 기술은 블록체인 외부에 있는 데이터에 의존할 수 있는 블록체인 네트워크에 게시된 프로그램 또는 스크립트의 실행에 관해 비트코인과 유사한 제한을 갖는 다른 블록체인 구현을 이용하는 것에 이점을 제공할 것이다.

[0003] 블록체인은, 블록으로 구성되고, 차례로 블록이 트랜잭션 및 다른 정보로 구성될 수 있는 컴퓨터 기반의 탈집중화된(decentralised) 시스템으로서 구현되는 피어 투 피어(peer-to-peer) 전자 원장을 지칭할 수 있다. 일부 예에서, "블록체인 트랜잭션"은 데이터 및 한 세트의 조건을 포함하는 필드 값의 구조화된 모음을 인코딩하는 입력 메시지를 지칭하며, 여기서 필드의 세트가 블록체인 데이터 구조에 기록되기 위해서는 조건의 세트의 이행이 필수이다. 예를 들어, 비트코인에서, 각각의 트랜잭션은 블록체인 시스템의 참가자 사이에서 디지털 자산의 제어권의 양도를 인코딩하는 데이터 구조이고, 적어도 하나의 입력 및 적어도 하나의 출력을 포함한다. 일부 실시예에서, "디지털 자산"은 사용할 권리와 연관된 이진 데이터를 지칭한다. 디지털 자산의 예는 비트코인, 이더(ether), 및 라이트코인(Litecoin)을 포함한다. 용어 "비트코인"은 본 명세서에서 비트코인 프로토콜의 변형인 임의의 프로토콜을 포함하는 것으로 사용된다. 일부 구현예에서, 디지털 자산의 제어권을 양도하는 것은 디지털 자산의 적어도 일부를 제 1 엔티티로부터 제 2 엔티티로 재연관시킴으로써 수행될 수 있다. 블록체인의 각각의 블록은 블록체인의 시작 이후 블록체인에 기록된 모든 트랜잭션의 영구적이고 변경 불가능한 레코드를 생성하기 위해 블록이 함께 체인화되게 하는 이전 블록의 해시를 포함할 수 있다.

[0004] 일부 예에서, "스택 기반의 스크립팅 언어(stack-based scripting language)"는 다양한 스택 기반 또는 스택 지향의 실행 모델 및 동작을 지원하는 프로그래밍 언어를 지칭한다. 즉, 스택 기반의 스크립팅 언어는 스택을 이용할 수 있다. 스택을 사용하여, 값이 스택 최상부(top)에 푸시될 수 있거나 또는 스택의 최상부로부터 팝핑(popped)될 수 있다. 스택을 조작하도록 수행되는 다양한 동작은 결과적으로 하나 이상의 값을 스택의 최상부에 푸시하거나 또는 최상부로부터 또는 팝핑되게 할 수 있다. 예를 들어, OP_EQUAL 연산은 최상부 2 개의 아이템을 스택으로부터 팝핑하고, 이들을 비교하고, 결과(예를 들어, 동일한 경우 1 또는 동일하지 않은 경우 0)를 스택의 최상부에 푸시한다. OP_PICK와 같은, 스택에 대해 수행되는 다른 연산은 아이템이 스택의 최상부 이외의 위치로부터 선택될 수 있게 할 수 있다. 본 실시예 중 일부에 의해 사용되는 일부 스크립팅 언어에서, 적어도 2 개의 스택: 메인 스택 및 대안적인 스택이 있을 수 있다. 스크립팅 언어의 일부 동작은 아이템을 하나의 스택의 최상부로부터 다른 스택의 최상부로 이동시킬 수 있다. 예를 들어, OP_TOALTSTACK는 값을 메인 스택의 최상부로부터 대안 스택의 최상부로 이동시킨다. 스택 기반의 스크립팅 언어는 일부 경우에 엄격한 후입 선출(last-in-first-out)(LIFO) 방식의 동작만으로 전적으로 제한되지 않을 수도 있다는 것을 유의하여야 한다. 예를 들어, 스택 기반의 스크립팅 언어는 스택 내의 n 번째 아이템을 최상부에 복사 또는 이동하는 연산(예를 들어, 비트코인에서, 각각 OP_PICK 및 OP_ROLL)을 지원할 수 있다. 스택 기반의 스크립팅 언어로 작성되는 스크립트는 벡터, 리스트, 또는 스택과 같은 임의의 적절한 데이터 구조를 사용하여 구현될 수 있는 논리적 스택 상으로 푸시될 수 있다.

[0005] 트랜잭션이 블록체인에 기입되게 하기 위해, 트랜잭션은 "유효(validated)"하여야 한다. 네트워크 노드(채굴 노드(mining node))는 네트워크로부터 거절된 무효 트랜잭션을 이용하여, 각각의 트랜잭션이 유효하다는 것을 보장하기 위한 작업을 수행한다. 노드는 유효성에 있어서 다른 노드와는 상이한 표준을 가질 수 있다. 블록체인에서 유효성은 합의 기반이기 때문에, 트랜잭션이 유효하다는 것을 대다수의 노드가 동의하면 트랜잭션은 유효한 것으로 간주된다. 노드 상에 설치된 소프트웨어 클라이언트는 미사용 트랜잭션(Unspent Transaction Output)(UTXO) 잠금 스크립트(locking script) 및 잠금해제 스크립트(unlocking script)를 실행함으로써 UTXO를 부분적으로 참조하는 트랜잭션에 대해 이러한 유효화 작업을 수행한다. 잠금 및 잠금해제 스크립트의 실행이 TRUE로 평가되면, 그리고 해당하는 경우 다른 유효화 조건이 충족되면, 트랜잭션은 노드에 의해 유효화된다. 유효화된 트랜잭션은 다른 네트워크 노드로 전파되고, 그 결과, 채굴 노드는 트랜잭션을 블록체인에 포함시키는 것을 선택할 수 있다. 따라서, 트랜잭션이 블록체인에 기입되게 하기 위해, 트랜잭션은 i) 트랜잭션을 수신하는 제 1 노드에 의해 유효화되어야 하고 - 트랜잭션이 유효화되면, 노드는 이 트랜잭션을 네트워크 내의 다른 노드로 중계함 - ; ii) 채굴 노드에 의해 구축되는 새로운 블록에 추가되어야 하고; iii) 채굴되어야 하고, 즉, 과거 트랜잭션의 공개 원장에 추가되어야 한다. 충분한 수의 블록이 블록체인에 추가되어 트랜잭션을 실제로 비가역적으로 만들 때 트랜잭션은 확인된 것으로 간주된다.

[0006] 블록체인 기술이 암호 화폐 구현의 사용을 위해 가장 널리 알려졌지만, 디지털 엔터프레뉴어(digital

entrepreneur)는 비트코인이 기초로 하는 암호화 보안 시스템의 사용 및 블록체인에 저장되어 새로운 시스템을 구현할 수 있는 데이터의 사용 둘 모두를 모색하기 시작하였다. 블록체인이 암호 화폐의 분야로 제한되지 않는 자동화된 태스크 및 프로세스에 사용될 수 있다면, 상당히 유리할 것이다. 이러한 솔루션은 블록체인의 이득(예를 들어, 이벤트의 영구적인 위조 방지 레코드, 분산 처리 등)을 이용할 수 있으면서 그들의 애플리케이션에서 보다 다목적으로 이용될 것이다.

[0007] 본 개시내용은 하나 이상의 블록체인 기반의 컴퓨터 프로그램의 기술적 양태를 설명한다. 블록체인 기반의 컴퓨터 프로그램은 블록체인 트랜잭션에 기록되는 머신 판독 가능 및 실행 가능 프로그램일 수 있다. 블록체인 기반의 컴퓨터 프로그램은, 결과를 생성하기 위해 입력을 처리할 수 있고 그런 다음 이들 결과에 따라 작용이 수행되게 할 수 있는 룰을 포함할 수 있다. 현재의 연구 분야 중 하나는 "스마트 계약(smart contract)"을 구현하기 위한 블록체인 기반의 컴퓨터 프로그램의 사용이다. 자연어로 작성될 수 있는 종래의 계약과는 달리, 스마트 계약은 머신 판독 가능 계약 또는 협약의 조건 실행을 자동화하도록 설계되는 컴퓨터 프로그램일 수 있다.

[0008] 블록체인과 관련된 다른 관심 영역은 블록체인을 통해 실세계 엔티티를 표현하고 양도하는 '토큰(token)' (또는 '컬러드 코인(coloured coin')의 사용이다. 잠재적으로 민감하거나 은밀한 아이템은 식별 가능한 의미 또는 값을 갖지 않는 토큰으로 표현될 수 있다. 따라서 토큰은 실제 아이템이 블록체인으로부터 참조될 수 있게 하는 식별자의 역할을 한다.

[0009] 실시예에서, 특정 엔티티와의 상호작용이 스마트 계약 내의 특정 단계에서 인코딩될 수 있지만, 다른 방식으로 스마트 계약은 자동으로 실행될 수 있고 자체 시행될 수 있다. 이것은 머신 판독 가능하고 실행 가능하다. 일부 예에서, 자동 실행은 UTXO의 양도를 가능할 수 있도록 성공적으로 수행되는 스마트 계약의 실행을 지칭한다. 이러한 예에서, UTXO의 양도를 야기할 수 있는 "엔티티"는 일부 비밀의 지식을 입증하도록 요구받지 않으면서도 잠금해제 스크립트를 생성할 수 있는 엔티티를 지칭한다는 것을 유의하여야 한다. 다시 말해, 잠금해제 트랜잭션은 데이터의 소스(예를 들어, 잠금해제 트랜잭션을 생성한 엔티티)가 암호화 비밀(예를 들어, 개인 비대칭 키, 대칭 키, 등)로의 액세스를 갖고 있다는 것을 검증하지 않고도 유효화될 수 있다. 또한, 이러한 예에서, 자체 시행은 제약에 따라 블록체인 네트워크의 유효화 노드(validation node)가 잠금해제 트랜잭션을 시행하게 되는 것을 지칭한다. 일부 예에서, UTXO를 "잠금해제하는 것"은 UTXO를 참조하고 유효한 것으로 실행하는 잠금해제 트랜잭션을 생성하는 것을 지칭한다.

[0010] 블록체인 트랜잭션 출력은 비트코인과 같은 디지털 자산의 소유권에 관한 잠금 스크립트 및 정보를 포함한다. 제한부담(encumbrance)으로도 지칭될 수 있는 잠금 스크립트는 UTXO를 양도하기 위해 충족될 필요가 있는 조건을 명시함으로써 디지털 자산을 "잠근한다". 예를 들어, 잠금 스크립트는 특정 데이터가 잠금해제 스크립트에 제공되어 연관된 디지털 자산을 잠금해제할 것을 요구할 수 있다. 잠금 스크립트는 비트코인에서 "scriptPubKey"로도 알려져 있다. 당사자에게 디지털 자산을 잠금해제할 데이터를 제공할 것을 요구하기 위한 기술에는 잠금 스크립트 내에 데이터의 해시를 삽입하는 것이 수반된다.

[0011] 본 발명은 제로 지식 증명(zero-knowledge proof)을 이용할 수 있는 블록체인 상에서 스마트 계약의 실행을 위한 시스템 및 방법으로서 설명될 수 있다. 스마트 계약의 실행은 트랜잭션 유효화의 일부로서 발생할 수 있다. 스마트 계약 실행의 일부로서, 실세계의 상태 및 이벤트에 관한 데이터와 같은 블록체인 외부의 데이터(예를 들어, 오프-체인 데이터)로의 액세스를 획득하는 것이 바람직할 수 있다. 본 명세서에 설명되는 기술은 데이터 소스에 의해 제공되는 데이터가 블록체인 네트워크에 게시된 스마트 계약과 같은 프로그램 또는 스크립트의 실행에서 인증되고 이용되는데 이용될 수 있다. 디지털 자산은 일부 실시예에서 암호 화폐로서 사용될 수 있지만, 실시예에서 디지털 자산은 다른 상황에서 추가적으로 또는 대안적으로 사용될 수 있다고 생각된다. 본 발명은 디지털 자산의 제어권에 적용할 수 있지만, 본질적으로 기술적인 것이며, 디지털 자산의 양도를 필수적으로 포함하지 않고 블록체인 데이터 구조를 이용하는 다른 맥락에서 사용될 수 있다는 것에 유의하여야 한다.

발명의 내용

해결하려는 과제

[0012] 따라서, 이러한 양태 중 하나 이상에서 블록체인 기술을 개선하는 방법 및 시스템을 제공하는 것이 바람직하다. 이러한 개선된 솔루션이 이제 고안되었다. 따라서, 본 발명에 따라 첨부된 청구 범위에 정의된 바와 같은 방법이 제공된다.

과제의 해결 수단

[0013] 이러한 개선된 솔루션이 이제 고안되었다.

[0014] 따라서, 본 발명에 따라 첨부된 청구 범위에 정의된 바와 같은 시스템 및 방법이 제공된다.

[0015] 본 발명에 따르면, 블록체인 네트워크의 노드의 컴퓨터 구현 방법이 제공될 수 있으며, 컴퓨터 구현 방법은: 컴퓨팅 엔티티와 암호로 보호된 통신 세션을 수립하는 단계; 암호로 보호된 통신 세션을 통해, 블록체인 네트워크에 게시된 프로그램의 실행을 제어하는 입력 데이터를 포함하는 통신을 수신하는 단계; 한 세트의 통신이 암호로 보호된 통신 세션을 통해 게시되었다는 제 1 입증(attestation)을 수신하는 단계 - 통신 세트는 데이터를 포함함 - ; 수신된 입력 데이터에 적어도 부분적으로 기초하여, 프로그램의 올바른 실행의 증명(proof) 및 데이터가 데이터 소스로부터 수신되었다는 제 2 입증을 생성하는 단계; 및 프로그램의 올바른 실행의 증명을 다른 컴퓨터 시스템에 제공하는 단계를 포함한다.

[0016] 추가적으로 또는 대안적으로 본 발명은 다음의 단계:

[0017] 컴퓨팅 엔티티와 암호로 보호된 통신 세션을 수립하는 단계;

[0018] 암호로 보호된 통신 세션을 통해, 블록체인 네트워크에 게시된 프로그램의 실행을 제어하는 입력 데이터를 포함하는 통신을 수신하는 단계;

[0019] 한 세트의 통신이 암호로 보호된 통신 세션을 통해 발생했다는 제 1 입증을 수신하는 단계 - 통신 세트는 데이터를 포함함 - ;

[0020] 수신된 입력 데이터에 적어도 부분적으로 기초하여, 프로그램의 올바른 실행의 증명 및 데이터가 데이터 소스로부터 수신되었다는 제 2 입증을 생성하는 단계; 및

[0021] 프로그램의 올바른 실행의 증명을 다른 컴퓨터 시스템에 제공하는 단계를 포함하는 것으로 설명될 수 있다.

[0022] 제 1 입증은 머클 트리의 루트 노드에 적어도 부분적으로 기초한 값을 가질 수 있으며, 머클 트리는 통신 세트로부터 결정된 한 세트의 리프 노드(leaf node)및 한 세트의 솔트 값(salt value)을 포함한다.

[0023] 통신 세트의 통신은 통신이 수신되었는지 또는 송신되었는지에 기초하여 결정된 대응하는 중간 노드를 가질 수 있다. 일부 경우에, 통신 세트의 각각의 통신은 이러한 대응하는 중간 노드를 갖는다.

[0024] 제 1 입증의 값은 통신 세트의 시간 및 적어도 머클 트리의 루트 노드로부터 생성된 암호화 해시 출력에 적어도 부분적으로 기초한다. 암호화 해시 출력을 생성하기에 적합한 암호화 해시 알고리즘의 예는 SHA 256 암호화 해시 알고리즘이다.

[0025] 제 2 입증은 머클 트리의 머클 경로에 적어도 부분적으로 기초할 수 있고, 머클 경로는 머클 트리의 한 세트의 노드의 값을 포함하고, 노드 세트의 값은 머클 트리의 루트 노드 값을 계산하기에 충분하다.

[0026] 바람직하게는, 머클 경로의 노드 세트는 머클 트리의 각각의 비 리프(non-leaf) 및 비 루트(non-root) 깊이에서 최대 하나의 노드를 포함한다. 일부 경우에, 머클 경로는 머클 트리의 각각의 비 리프 및 비 루트 깊이에서 정확히 하나의 노드를 포함한다.

[0027] 프로그램은 2 이상의 당사자에 의해 합의된 한 세트의 룰을 포함할 수 있고, 방법은 두 당사자 중 적어도 하나가 신뢰할 수 있는 하나 이상의 컴퓨팅 엔티티로부터 암호화 보호 통신을 수립하는 컴퓨팅 엔티티를 선택하는 단계를 포함할 수 있다.

[0028] 바람직하게는, 방법은 제 1 트랜잭션 출력 및 제 2 트랜잭션 출력을 포함하는 블록체인 거래를 검출하는 단계 - 제 1 트랜잭션 출력은 제 1 잠금 스크립트를 포함하고, 제 1 트랜잭션 출력과 연관된 제 1 디지털 자산은: 컴퓨팅 엔티티와 연관된 공개 키; 기대 값을 인코딩하는 디지털 서명 - 디지털 서명의 진위성(authenticity)은 공개 키를 사용하여 암호로 검증 가능함 - , 및 기대 값을 생성하는데 사용 가능한 인증 정보;를 인코딩하는 잠금해제 스트립트에 의해 잠금해제 가능하고, 제 2 트랜잭션 출력은 올바른 실행의 증명이 유효하다는 표시를 인코딩함 - ; 및 적어도 공개 키, 디지털 서명 및 인증 정보를 제공함으로써 제 1 디지털 자산을 잠금 해제하는 단계를 포함한다.

[0029] 블록체인 트랜잭션은 다른 컴퓨터 시스템과 연관된 개인 키를 사용하여 디지털 서명된 트랜잭션 입력; 및 제 2 잠금해제 스크립트를 포함하는 제 3 트랜잭션 출력 - 제 3 트랜잭션 출력과 연관된 제 2 디지털 자산은 개인 키

를 사용하여 잠금해제 가능함 - 을 더 포함할 수 있으며; 제 2 트랜잭션 출력은 또한 다른 컴퓨터 시스템과 연관된 식별자를 인코딩한다.

[0030] 바람직하게는, 인증 정보는 머클 트리의 머클 경로를 포함하고 기대 값은 머클 트리의 루트 노드에 적어도 부분적으로 기초한다.

[0031] 데이터는 이벤트가 발생했는지 발생하지 않았는지를 표시하는 이진 데이터를 포함할 수 있다.

[0032] 데이터는 블록체인 상의 다른 데이터에 기초하여 정확성을 검증할 수 없는 정보를 포함하는 데이터일 수 있다.

[0033] 바람직하게는, 제 1 입증은 디지털 서명이며, 디지털 서명의 진위성은 컴퓨팅 엔티티와 연관된 암호화 공개 키를 사용하여 검증 가능하다.

[0034] 또한, 프로세서; 및 프로세서에 의한 실행의 결과로서, 시스템으로 하여금 청구된 바와 같은 방법 중 임의의 방법을 수행하게 하는 실행 가능한 명령어를 포함하는 메모리;를 포함하는 시스템을 제공하는 것이 바람직하다.

[0035] 또한, 컴퓨터 시스템의 하나 이상의 프로세서에 의한 실행의 결과로서, 컴퓨터 시스템으로 하여금 청구된 바와 같은 방법 중 임의의 방법을 적어도 수행하게 하는 실행 가능한 명령어가 저장된 비일시적 컴퓨터 판독 가능 저장 매체를 제공하는 것이 바람직하다.

[0036] 추가적으로 또는 대안적으로, 본 발명은 개선된 블록체인 프로그래밍 툴 또는 보조를 제공할 수 있다. 분산되고 검증 가능한 계산을 용이하게 하거나 가능하게 하는 개선되고 효율적이며 최적화된 장치를 제공할 수 있다.

도면의 간단한 설명

[0037] 본 발명의 이러한 양태 및 다른 양태는 본 명세서에 설명된 실시예로부터 명백해지고 실시예를 참조하여 설명될 것이다. 본 발명의 실시예는 이제 단지 예로서 그리고 첨부 도면을 참조하여 설명될 것이다.

도 1은 다양한 실시예가 구현될 수 있는 블록체인 환경을 예시한다.

도 2는 다양한 실시예에 따른 프로토콜을 구현하는데 이용될 수 있는 컴퓨팅 환경을 예시한다.

도 3은 검증 가능한 계산의 수행에 적합한 환경의 다이어그램을 예시한다.

도 4는 증명자(prover)가 프로그램 또는 스크립트의 실행과 관련하여 데이터 소스로부터 획득된 데이터를 이용하는 환경의 예시적인 다이어그램을 예시한다.

도 5는 프로그램 또는 스크립트의 실행에 이용되는 데이터 소스로부터 데이터를 획득하기 위한 프로토콜을 예시하는 다이어그램을 예시한다.

도 6은 실시예에 따른 머클 트리의 다이어그램을 예시한다.

도 7은 실시예에 따른 머클 경로의 다이어그램을 예시한다.

도 8은 다양한 실시예와 관련하여 이용되는 증명을 생성 및 검증하기 위한 프로토콜의 다이어그램을 예시한다.

도 9는 실시예에 따른 평판 트랜잭션(reputation transaction)의 다이어그램을 예시한다.

도 10은 통신의 증명을 생성하기 위한 프로세스의 예시적인 다이어그램이다.

도 11은 인증된 데이터를 사용하여 프로그램 또는 스크립트를 실행하기 위한 프로세스의 예시적인 다이어그램이다.

도 12는 본 개시내용의 적어도 하나의 실시예를 실시하는데 사용될 수 있는 컴퓨팅 디바이스를 예시한다.

발명을 실시하기 위한 구체적인 내용

[0038] 본 개시내용의 실시예에 따른 블록체인과 연관된 예시적인 블록체인 네트워크(100)를 예시하는 도 1이 먼저 참조될 것이다. 실시예에서, 예시적인 블록체인 네트워크(100)는 피어 투 피어 분산 전자 디바이스로서 구현되는 블록체인 노드를 포함하며, 각각의 디바이스는 적어도 부분적으로 노드(102)의 운영자 사이에 합의되는 블록체인 프로토콜을 따르는 동작을 수행하는 소프트웨어 및/또는 하드웨어의 인스턴스를 실행한다. 일부 예에서, "노드"는 블록체인 네트워크 사이에 분산되는 피어 투 피어 전자 디바이스를 지칭한다. 블록체인 프로토콜의 예는 비트코인 프로토콜이다.

[0039] 일부 실시예에서, 노드(102)는 (예를 들어, 데이터 센터의 서버에 의해, 클라이언트 컴퓨팅 디바이스(예를 들어, 데스크톱 컴퓨터, 랩톱 컴퓨터, 태블릿 컴퓨터, 스마트 폰 등)에 의해, 컴퓨팅 자원 서비스 제공자의 분산 시스템 내의 다수의 컴퓨팅 디바이스에 의해, 또는 도 8의 컴퓨팅 디바이스(800)와 같은 임의의 적절한 전자 클라이언트 디바이스에 의해) 임의의 적합한 컴퓨팅 디바이스로 구성될 수 있다. 일부 실시예에서, 노드(102)는 트랜잭션(104)과 같은 제안된 트랜잭션을 나타내는 데이터 메시지 또는 객체를 수신하는 입력을 갖는다. 일부 실시예에서, 노드는 노드가 보유하는 정보에 대해, 이를테면 트랜잭션(104)의 상태의 정보에 대해 질의 가능하다.

[0040] 도 1에 도시된 바와 같이, 일부 노드(102)는 하나 이상의 다른 노드(102)에 통신 가능하게 결합된다. 이러한 통신 결합은 유선 또는 무선 통신 중 하나 이상을 포함할 수 있다. 실시예에서, 노드(102) 각각은 블록체인 내의 모든 트랜잭션의 "원장"의 적어도 일부를 보유한다. 이러한 방식으로, 원장은 분산 원장일 수 있다. 원장에 영향을 미치는 노드에 의해 처리된 트랜잭션은 원장의 무결성이 유지되도록 하나 이상의 다른 노드에 의해 검증 가능하다.

[0041] 어떤 노드(102)가 어느 다른 노드와 통신할 수 있는지에 관해, 노드 사이에 전달되는 메시지가 블록체인 프로토콜이 전송되어야 한다고 표시하는 메시지인 것으로 가정하면, 그 메시지가 예시적인 블록체인 네트워크(100)(또는 그의 일부 중요 부분)를 통해 전파될 수 있도록 예시적인 블록체인 네트워크(100) 내의 각각의 노드가 하나 이상의 다른 노드(102)와 통신할 수 있는 것이면 충분할 수 있다. 하나의 이러한 메시지는 노드(102A)와 같은 노드(102) 중 하나에 의한 제안된 트랜잭션의 게시일 수 있으며, 이 메시지는 경로(106)와 같은 경로를 따라 전파될 수 있다. 다른 이러한 메시지는 블록체인에 포함시키기 위해 제안된 새로운 블록의 게시일 수 있다.

[0042] 실시예에서, 노드(102) 중 적어도 일부는 암호화 문제 해결과 같은 복잡한 계산을 수행하는 채굴 노드이다. 암호화 문제를 해결하는 채굴 노드는 블록체인에 새로운 블록을 생성하고 그 새로운 블록을 노드(102)의 다른 노드에 브로드캐스트한다. 노드(102)의 다른 노드는 채굴 노드의 작업을 검증하고, 검증시 (예를 들어, 블록체인의 분산 원장에 블록을 추가함으로써) 블록체인 내에 블록을 수락한다. 일부 예에서, 블록은 종종 이전 블록의 타임스탬프 및 "핑거프린트"(예를 들어, 해시)로 마킹된 트랜잭션 그룹이다. 이러한 방식으로, 각각의 블록은 이전 블록에 링크될 수 있고, 이에 의해 블록체인 내의 블록을 링크하는 "체인"을 생성할 수 있다. 실시예에서, 유효 블록이 노드(102)의 합의에 의해 블록체인에 추가된다. 또한 일부 예에서, 블록체인은 유효화된 블록의 리스트를 포함한다.

[0043] 실시예에서, 노드(102) 중 적어도 일부는 본 개시내용에 설명된 바와 같이 트랜잭션을 유효화하는 유효화 노드로서 동작한다. 일부 예에서, 트랜잭션은 디지털 자산(예를 들어, 다수의 비트 코인)의 소유권의 증명을 제공하는 데이터 및 디지털 자산의 소유권/제어권을 수락 또는 양도하기 위한 조건을 포함한다. 일부 예에서, "잠금해제 트랜잭션"은 이전 트랜잭션(previous transaction)의 UTXO로 표시되는 디지털 자산의 적어도 일부를 블록체인 주소와 연관된 실체에 재연관시키는 (예를 들어, 소유권 또는 제어권을 양도하는) 블록체인 트랜잭션을 지칭한다. 일부 예에서, "이전 트랜잭션"은 잠금해제 트랜잭션이 참조하는 UTXO를 포함하는 블록체인 트랜잭션을 지칭한다. 일부 실시예에서, 트랜잭션은 소유권/제어권이 양도("잠금해제")되기 전에 충족되어야 하는 조건으로 트랜잭션을 담보하는 "잠금 스크립트"를 포함한다.

[0044] 일부 실시예에서, 블록체인 주소는 디지털 자산의 적어도 일부의 제어권이 양도/재연관되는 엔티티와 연관된 영숫자 문자의 열이다. 일부 실시예에서 구현된 일부 블록체인 프로토콜에서, 엔티티와 연관된 공개 키와 블록체인 주소 사이에는 일대일 대응 관계가 있다. 일부 실시예에서, 트랜잭션의 유효화에는 잠금 스크립트 및/또는 잠금해제 스크립트에 명시된 하나 이상의 조건을 유효화하는 것이 수반된다. 트랜잭션(104)의 유효화에 성공하면, 유효화 노드는 트랜잭션(104)을 블록체인에 추가하고 이를 노드(102)에 분산시킨다.

[0045] 본 명세서에 설명된 시스템 및 방법은 잠금 스크립트가 검증 키(V_K)를 변경으로부터 보호할 수 있게 하고 증명(π)의 유효성을 체크함으로써, 트랜잭션 검증 동안 블록체인 상에서 제로 지식 프로토콜의 실행을 가능하게 하는 것에 관한 것이다.

[0046] 검증 가능한 계산은 계산의 증명의 생성을 가능하게 하는 기술이다. 실시예에서, 이러한 기술은 클라이언트에 의해, 입력(x) 상의 함수(f)의 평가를 본 명세서에서 증명자로 지칭되는 다른 컴퓨팅 엔티티에 아웃소싱하는데 이용된다. 일부 경우에, 클라이언트는 계산적으로 제한되어, 그럴 필요가 없지만, 클라이언트가 함수의 평가를 수행하는 것을 실행할 수 없고(예를 들어, 클라이언트가 이용할 수 있는 컴퓨팅 자원을 사용한 계산의 예상 런타임이 최대 허용 가능 임계치를 초과하고), 일반적으로 말하면, 클라이언트는 계산 런타임, 계산 비용(예를 들

어, 함수의 평가를 수행하기 위해 컴퓨팅 자원을 할당하는데 드는 재정 비용) 등과 같은 임의의 적절한 기준에 기초하여 입력(x) 상의 함수(f)의 평가를 위임할 수 있다.

[0047] 실시예에서, 증명자는 본 개시내용의 다른 곳에서 더 상세하게 설명되는 바와 같은 블록체인 노드와 같은 임의의 적합한 컴퓨팅 엔티티이다. 실시예에서, 증명자(예를 들어, 블록체인 노드)는 입력(x) 상의 함수(f)를 평가하고, 위에서 설명된 바와 같은 클라이언트 및/또는 블록체인 네트워크의 다른 노드와 같은 다른 컴퓨팅 엔티티에 의해 검증될 수 있는 출력(y) 및 출력(y)의 정확성의 증명(π)을 생성한다. 인수(argument)라고도 지칭될 수 있는 증명은 입력(x) 상의 함수(f)를 다시 계산하는 대신에 증명의 정확성을 검증하여 위에서 설명된 증명자에 의해 생성된 출력의 정확성을 결정함으로써 실제 계산을 수행하는 것보다 빠르게 검증될 수 있다 - 따라서 계산 오버헤드가 감소될 수 있다(예를 들어, 전력 오버헤드 및 컴퓨팅 자원의 전력 공급 및 실행과 연관된 비용을 감소시킬 수 있다). 제로 지식 검증 가능한 계산에서, 증명자는 증명자가 특정 속성을 가진 입력을 알고 있다는 입증을 클라이언트에게 제공한다.

[0048] 지식의 제로 지식 증명(zero-knowledge proof)의 효율적인 변형은 zk_SNARK(Succinct Non-interactive ARgument of Knowledge)이다. 실시예에서, 모든 페어링 기반 zk_SNARK는 증명자가 포괄적인 그룹 연산을 사용하여 그룹 요소의 수를 계산하고 검증자가 다수의 페어링 곱 방정식을 사용하여 증명을 체크하는 프로세스를 포함한다. 실시예에서, 선형의 대화식 증명은 유한 필드에 걸쳐 작동하며, 증명자의 메시지 및 검증자의 메시지는 필드 요소의 벡터를 결정하는데 사용 가능한 정보를 포함하거나, 인코딩하거나, 참조하거나 또는 그렇지 않으면 포함한다.

[0049] 실시예에서, 본 명세서에 설명된 시스템 및 방법은 블록체인의 채굴 노드가 계산(예를 들어, 입력(x)상의 함수(f)의 평가)을 한 번 수행할 수 있게 하고 출력의 정확성을 검증하는데 사용될 수 있는 증명을 생성할 수 있게 하는데, 여기서 증명의 정확성을 평가하는 것은 함수를 평가하는 것보다 계산 비용이 저렴하다. 이러한 맥락에서, 연산 및 작업의 비용(즉, 얼마나 비싼지)은 연산 또는 작업을 수행하는 계산상의 복잡성을 지칭할 수 있다. 실시예에서, 계산상의 복잡성은 소팅 알고리즘(sorting algorithm)을 수행하는 평균 계산 비용 또는 최악의 계산 비용을 지칭하는데 - 예를 들어, 힙소트 알고리즘(heapsort algorithm) 및 퀵소트 알고리즘(quicksort algorithm) 둘 모두는 $O(n \log n)$의 평균 계산 비용을 가지만, 퀵소트는 $O(n^2)$의 최악의 계산 비용을 갖고 반면에 힙소트는 $O(n \log n)$의 최악의 계산 비용을 갖는다. 실시예에서, 입력(x) 상의 함수(f)를 평가하는 평균 계산 비용 및/또는 최악의 계산 비용은 증명의 정확성을 평가하는 것보다 더 나쁘다. 따라서, 본 명세서에 설명된 시스템 및 방법의 사용은 매우 유리하며, 예를 들어, 이러한 계약이 블록체인을 비례적으로 검증하는데 필요한 시간을 증가시키지 않을 것이므로 더 계산적으로 비싼 계약이 실행될 수 있게 할 수 있다. 추가 이점은 검증자 시스템의 전력 소비의 감소를 포함할 수 있고, 그럼으로써 검증자 컴퓨터 시스템의 효율성을 개선할 수 있으며 증명의 정확성을 평가할 때 이러한 검증자 컴퓨터 시스템을 실행하는 것과 연관된 에너지 비용을 감소시킬 수 있다.

[0050] 실시예에서, 검증 키(V_K) 또는 그 일부는 제로 지식 프로토콜(zero-knowledge protocol)의 셋업 단계에서 생성될 수 있고 증명(π)과 함께 사용되는 공개 파라미터 및 증명자에 의해 제공되는 주장된 정확성 증명 계산을 검증하는 입력/출력 데이터로부터 추출될 수 있다. 예를 들어, 위에서 그리고 아래에서 자세히 설명되는 바와 같이, 잠금 스크립트를 허용하는 시스템 및 방법은 검증 키(V_K)를 변경으로부터 보호하고 증명(π)의 유효성을 체크하여, 트랜잭션 유효화 동안 블록체인 상에서 제로 지식 프로토콜의 실행을 가능하게 한다. 따라서, 본 개시내용은 계산의 검증에 사용된 요소를 저장하기 위한 (예를 들어, 비트코인 기반 네트워크에서) 블록체인 스크립트를 사용하여 검증 단계를 실행하는 시스템 및 방법을 제시한다.

[0051] 도 2는 다양한 실시예에 따른 프로토콜을 구현하는데 사용될 수 있는 컴퓨팅 환경(200)을 도시한다. 프로토콜은 정확성 증명(proof-of-correctness)을 저장하고 "구성 별 보정(correct-by-construction)" 암호화 기법을 스마트 계약과 결합하는 블록체인 기술을 사용하여 구현될 수 있다. 실시예에서, 공개적 검증 가능한 계산 방식은 3 개의 단계: 셋업 단계, 계산 단계 및 검증 단계를 포함한다.

[0052] 셋업 단계는 계산 작업의 수행을 아웃소싱하는 프로세스의 일부로서 수행될 수 있다. 이하에서 언급되는 바와 같이, 클라이언트는 계산 작업의 수행을 상이한 컴퓨터 시스템일 수 있는 증명자에게 위임하는 고객 또는 클라이언트 컴퓨터 시스템과 같은 엔티티를 지칭할 수 있다. 일반적으로 말하면, 클라이언트는 이것으로 제한되는 것은 아니지만 제한된 컴퓨팅 자원, 컴퓨팅 자원 부족, 클라이언트 컴퓨터 시스템을 사용하여 작업을 수행하는

것과 연관된 재정 비용, 클라이언트 컴퓨터 시스템을 이용하여 작업을 수행하는 것과 연관된 에너지 비용 등을 비롯한 다양한 이유로 계산 작업의 수행을 위임할 수 있다(예를 들어, 전력을 위해 배터리에 의존하는 모바일 디바이스 또는 랩톱은 증명자를 이용하여 계산 집약적인 작업을 수행할 수 있고, 그럼으로써, 전력을 절약하고 배터리 구동 디바이스의 사용을 연장할 수 있다).

[0053] 실시예에서, 셋업 단계는 클라이언트, 고객, 조직의 직원, 또는 임의의 다른 적절한 엔티티가 올바른 시맨틱을 갖는 공식 언어로 계약을 작성하는 것을 포함한다. 계약은 C 또는 자바(Java)와 같은 고급 프로그래밍 언어로 작성될 수 있다. 일반적으로 말하면, 계약은 컴퓨터 시스템에 의해 조작될 수 있는 포맷이거나 포맷으로 변환될 수 있는 임의의 언어 또는 신택스로 표현될 수 있다. 실시예에서, 제한된 목적을 가진 도메인 특정 언어는 타입-세이프티(type-safety)를 제공할 수 있고 제한된 표현성이 이용될 수 있다. 생성된 소스 코드는 계약의 올바른 설명일 수 있다.

[0054] 컴파일러(202)는, 컴퓨터 시스템의 하나 이상의 프로세서에 의해 실행되면, 시스템이 입력으로서 소스 코드(206)를 취하고 회로를 생성하게 하는 실행 가능 코드를 포함하는 임의의 하드웨어, 소프트웨어 또는 이들의 조합일 수 있다. 컴파일러(202)는 이진 코드와 같은 기계 판독 가능 포맷으로 컴파일된 명령어에 기초하여 명령어를 실행하거나 수행하는 컴퓨터 프로그램을 지칭할 수 있다. 컴파일러(202)가 예시되어 있지만, 인터프리터, 어셈블러 및 다른 적합한 소프트웨어 및/또는 하드웨어 컴포넌트가 소스 코드를 회로로 변환하는데 이용될 수 있다는 것을 유의하여야 한다. 실시예에서, 회로는 필드(F)로부터 값을 운반하고 논리 및/또는 산술 게이트에 연결하는 와이어를 포함하는 산술 회로이다. 실시예에서, 회로($\mathcal{C}$)는 시스템에 의해 원래 회로($\mathcal{C}$)의 완전한 설명을 제공하는 한 세트의 다항식을 포함하는 이차 프로그램 Q(208)를 생성하는데 사용된다.

[0055] 실시예에서, 컴파일러(202)는 이것으로 제한되는 것은 아니지만 프리-프로세서 디렉티브(pre-processor directive), 스태틱 이니셜라이저(static initializer), 글로벌 및 로컬 함수, 블록 범위 변수, 어레이, 데이터 구조, 포인터, 함수 호출, 함수 연산자(예를 들어, 펑터(functor)), 조건부 및 루프, 산술 및 비트와이즈 부울 연산자를 비롯한 C 또는 자바와 같은 프로그래밍 언어의 실질적인 서브세트를 인식할 수 있다. 실시예에서, 컴파일러(202)는 프로그래밍 언어의 표준에 따라 전체 커맨드 세트를 지원하지 않는다(일부 경우에, 이것은 재귀 알고리즘을 금지하는 것과 같이, 특정 유형의 알고리즘이 스마트 계약에서 실행되는 것을 방지하도록 의도될 수 있다). 실시예에서, 컴파일러는 소스 코드의 표현을 산술 게이트 언어로 확장하여 산술 회로를 생성한다. 과거에는 Campanelli, M. 등에 의해 "Zero-Knowledge Contingent Payments Revisited: Attacks and Payments for Services"(2017)에서 그리고 Tillich, S. 및 Smart, B에 의해 "Circuits of Basic Functions Suitable For MPC and FHE"에서 회로 구현이 고려되었다. 산술 회로는 컴파일러(202) 또는 임의의 다른 적절한 하드웨어, 소프트웨어 또는 이들의 조합(예를 들어, 도 2에 도시되지 않은 소프트웨어 모듈)에 의해 이차 산술 문제(Quadratic Arithmetic Problem)(QAP)를 구축하는데 이용될 수 있다. 이차 프로그램은 실시예에 따라 클라이언트의 한 세트의 암호화 루틴(예를 들어, 키 생성 및 검증) 및 검증자의 한 세트의 암호화 루틴(예를 들어, 계산 및 증명 생성)으로 컴파일된다.

[0056] 실시예에서, 키 생성기(204)는, 컴퓨터 시스템의 하나 이상의 프로세서에 의해 실행되면, 시스템이 이차 프로그램으로부터 평가 키 및 검증 키를 생성하게 하는 실행 가능 코드를 포함하는 하드웨어, 소프트웨어 또는 이들의 조합이다. 이차 프로그램으로서 계산을 인코딩하는 기술은 Gennaro, R. 등의 "Quadratic Span Programs and Succinct NIZKs without PCPs"(2013)에서 고려된다. 실시예에서, 이차 산술 문제(QAP)(Q)는 필드(F)를 통해 회로($\mathcal{C}$)를 인코딩하고 한 세트의 m+1 다항식:

[0057] $$V = \{v_k(x)\},\ W = \{w_k(x)\},\ Y = \{y_k(x)\}$$

[0058] 을 포함하고, $0 \le k \le m$ 이다. 타겟 다항식 t(x)이 또한 정의된다. F의 n 개 요소를 입력으로서 취하고 n' 개 요소를 출력하는, N = n + n'인, 함수(f)가 주어지면, $\{c_1, \ldots, c_N\} \in F^N$ 이 입력 그룹의 유효한 할당이고 t(x)가 p(x)를 나눗셈하도록 계수 $\{c_{N+1}, \ldots, c_m\}$ 의 리스트가 존재하면, Q는 f를 계산한다.

$$p(x) = \left(v_0(x) + \sum_{k=1}^{m} c_k \cdot \mathrm{v_k(x)}\right) \cdot \left(w_0(x) + \sum_{k=1}^{m} c_k \cdot \mathrm{w_k(x)}\right) - \left(y_0(x) + \sum_{k=1}^{m} c_k \cdot \mathrm{y_k(x)}\right)$$

[0059]

[0060] 그러므로, 하나의 실시예에서, h(x) · t(x) = p(x)가 되도록 일부 다항식 h(x)가 존재해야 한다. Q의 크기는 m이고, 그 차수(degree)는 t(x)의 차수이다.

[0061] 실시예에서, 산술 회로에 대한 QAP를 구축하는 것은 회로의 각각의 승산 게이트(multiplication gate)(g)마다 임의의 루트 $r_g \in F$ 를 선택하고 타겟 다항식을 $t(x) = \prod_g (x - r_g)$ 로 정의하는 것을 포함한다. 실시예에서, 인덱스 $k \in \{1 \dots m\}$ 는 회로의 각각의 입력 및 승산 게이트로부터의 각각의 출력에 연관된다. V에서 다항식은 좌측 입력을 각각의 게이트에 인코딩하고, W는 우측 입력을 각각의 게이트에 인코딩하고, Y는 출력을 인코딩한다. 예를 들어, k 번째 와이어가 게이트(g)로의 좌측 입력이면 $\mathrm{v_k(r_g)} = 1$ 이며, 그렇지 않으면 $\mathrm{v_k(r_g)} = 0$ 이다. 그러므로, 특정 게이트(g)와 그의 루트(r_g)에 대해, 이전 방정식은 다음과 같이 단순화될 수 있다.

$$\left(\sum_{k=1}^{m} c_k \cdot \mathrm{v_k(r_g)}\right) \cdot \left(\sum_{k=1}^{m} c_k \cdot \mathrm{w_k(r_g)}\right) = \left(\sum_{k \in \mathrm{I_{left}}} c_k\right) \cdot \left(\sum_{k \in \mathrm{I_{right}}} c_k\right) = c_g \mathrm{y_k}(r_g) = c_g$$

[0062]

[0063] 게이트의 출력 값은 입력의 곱과 같다. 가분성 체크(divisibility check)는 $p(r_g)=0$ 이 되도록 t(x)의 각각의 게이트(g) 및 루트(r_g)마다 하나씩 deg(t(x)) 개별 체크로 분해한다. 가산 게이트 및 상수 승산(multiplication-by-constant) 게이트는 QAP의 크기 및 차수에 기여하지 않는다.

[0064] 실시예에서, QAP는 필드(Fp)에 대해 정의되며, 여기서 p는 큰 소수(prime)이다. 실시예에서, Fp에 대한 QAP는 가산 및 승산 모듈로(p)의 항으로 표현될 수 있는 임의의 함수를 효율적으로 계산하는 것이 바람직하다. 산술 분할 게이트(arithmetic split gate)는 $[0, 2^{k-1}]$에 있는 것으로 알려진 입력 $\mathrm{a} \in Fp$를 k 개의 이진 출력 와이어로 변환하도록 설계될 수 있다. 따라서, 부울 함수는 산술 게이트를 사용하여 표현될 수 있다. 예를 들어 $\mathrm{NAND(a,b)} = 1 - \mathrm{ab}$ 이다. 각각의 내장형 부울 게이트는 단 한 번의 승산만을 수행한다. 또한, split와 같은 새로운 게이트는 스탠드얼론으로 정의되고 다른 게이트로 구성될 수 있다. 입력 $a \in F_p$가 $[0, 2^{k-1}]$에 있는 것으로 알려지면, 분할 게이트는 $\sum^{k} 2^{i-1} a_i = a$ 이도록 또한 각각의 a_i가 0 또는 1이도록, a 의 이진 디지트 $a_1, \dots, a_k$ 를 갖는 k 개의 와이어를 출력한다.

[0065] 마지막으로, 모든 증명자 및 검증자에 의해 사용될 공용 파라미터는 셋업 단계의 일부로서 시스템에 의해 생성된다. 평가 키(E_K) 및 검증 키(V_K)는 클라이언트에 의해 선택된 비밀 값을 사용하여 도출된다는 것에 유의하여야 한다. 키 생성기(204)는 키 생성 알고리즘과 관련하여 이차 산술 프로그램(QAP)을 이용하여 평가 키(E_K)(210) 및 검증 키(V_K)(212)를 생성할 수 있다.

[0066] 실시예에서, 계산 작업을 수행하는 데는 증명자에 의한 입력(216) 상의 함수의 계산(즉, f(x)를 평가하는 프로세스)이 수반된다. 실시예에서, 증명자는 클라이언트가 계산 작업을 위임할 수 있는 임의의 적합한 컴퓨터 시스템이다. 실시예에서, 입력(216)은 증명자와 연관된 개인 키를 사용하여 생성된 디지털 서명과 같은 증명자의 신원을 입증하는 정보를 포함한다. 실시예에서, 증명자는 클라이언트가 성공적인 결과로서 디지털 자산을 양도하는 컴퓨터 시스템이다. 실시예에서, 클라이언트는 입력(x) 및 평가 키(E_K)를 증명자에게 제공하고, 증명자는 계산 루틴에 대한 평가 모듈(214)을 사용하여 출력(y)(즉, y=f(x), 여기서 입력은 x이고 함수는 f임)을 계산

하며 또한 평가 키(E_K)를 이용하여 정확성 증명(218)을 생성한다. 실시예에서, 평가 모듈은, 컴퓨터 시스템의 하나 이상의 프로세서에 의해 실행되면, 컴퓨터 시스템이 QAP(208)의 내부 회로 와이어의 값을 평가하게 하고 QAP의 출력(y)을 생성하게 하는 명령어를 포함하는 하드웨어 및/또는 소프트웨어이다.

[0067] 실시예에서, 이차 프로그램의 각각의 다항식 $v_k(x) \in F$ 는 이중 선형 그룹의 요소 $g^{v_k(s)}$ 에 매핑되며, 여기서 s는 클라이언트에 의해 선택된 비밀 값이고, g는 그룹의 생성자이며, F는 g의 이산 로그의 필드이다. 실시예에서, 주어진 입력에 대해, 증명자는 회로를 평가하여 이차 프로그램의 계수(c_i)에 대응하는 내부 회로 와이어의 값 및 출력을 획득한다. 따라서, 증명자는 $v(s) = \sum_{k\in\{m\}} c_k \cdot v_k(s)$ 를 평가하여 $g^{v(s)}$ 를 얻을 수 있고; w(s) 및 y(s)를 계산하고; $h(x) = p(x) / t(x) = \sum^{d} h_i \cdot x^i$ 를 계산하고; 평가 키의 h_i 및 $g^{s(i)}$ 항을 사용하여 $g^{h(s)}$ 를 계산할 수 있다. 실시예에서, 정확성 증명(218)은 $(g^{v(s)}, g^{w(s)}, g^{y(s)}, g^{h(s)})$ 를 포함하고 검증자는 이중 선형 맵을 사용하여 $p(s) = h(s) \cdot t(s)$ 를 체크한다. 실시예에서, 증명(π)은 나중에 사용하기 위해 블록체인(222)에 저장되거나 또는 증명자가 이들 각각과 개별적으로 상호작용할 필요없이 다수의 당사자에 의해 검증될 수 있다. 실시예에서, 정확성 증명의 회로 저장의 평가는 트랜잭션의 잠금 스크립트에 의해 담보되는 자원(예를 들어, 디지털 자산)을 잠금해제하기 위해 수행될 수 있다.

[0068] 실시예에서, 증명(π)은 블록체인 네트워크에 브로드캐스트되고 검증자(220)는 증명을 검증하는데 사용된다. 실시예에서, 검증자(220)는 블록체인 상의 노드와 같은 임의의 적합한 컴퓨팅 엔티티이다. 일부 경우에, 평가 키(E_K) 및 검증 키(V_K)를 생성하는 동일한 컴퓨팅 엔티티가 또한 증명을 검증한다는 점에 유의하여야 한다. 하나의 실시예에서, 블록체인의 노드는 검증 키(V_K) 및 증명(π)을 사용하여 잠금 트랜잭션을 유효화할 수 있으므로, 검증이 성공하면 계약을 유효화한다. 프로토콜의 하나의 요건은 검증자가 검증 키(V_K)를 알고 있을 때조차 잘못된 증명을 제공할 수 없다는 것이다. 따라서, 이 프로토콜에서, 공통 참조 스트링(common reference string)(CRS)은 클라이언트에 의해 또는 적어도 평가 키(E_K) 및 검증 키(V_K)를 발행하는 신뢰할 수 있는 제3 자에 의해 생성된다. 하나의 실시예에서, 발행된 검증 키(V_K)는 임의의 컴퓨팅 엔티티에 의해 계산을 검증하는데 사용될 수 있다.

[0069] 본 명세서에 설명된 기술을 사용하면, 클라이언트는 블록체인 트랜잭션의 수신자의 신원과 같은 트랜잭션 데이터를 부분적으로 난독화(obfuscate)할 수 있다. 실시예에서, 잠금해제 스크립트는 수신자의 주소 및 수신자의 공개 키를 노출시키지 않는다. 그러나 일부의 경우, 트랜잭션의 값(예를 들어, 양도된 디지털 자산의 양)이 블록체인 네트워크의 노드에게 보일 수 있다. 실시예에서, 위에서 및 아래에서 설명되는 바와 같은 암호화 기술은 클라이언트에 의해 잠금 스크립트를 이차 산술 프로그램 및 증명자에 은닉하여 산술 프로그램을 풀어 증명을 생성하는데 이용된다.

[0070] 일반적으로 말하면, 클라이언트는 상대방 또는 증명자에게 지불하기 위해 P2PK 및 P2PKH와 같은 표준 트랜잭션(예를 들어, 비트코인 기반 블록체인 네트워크에 정의된 바와 같은 표준 트랜잭션)을 사용할 수 있다. 예를 들어, 실시예에서, 클라이언트는 P2PK 잠금 스크립트를 산술 회로로 변환하고 그 회로로부터 도출된 퍼즐을 포함하는 잠금 트랜잭션을 브로드캐스트한다. 상대방 또는 증명자는 회로를 수신하고, 적절한 입력(예를 들어, 클라이언트와 증명자 사이의 공유 비밀 또는 증명자의 개인 키를 사용하여 생성된 디지털 서명과 같은 증명자의 신원을 입증하는 정보)을 제공하고, 회로를 실행하여 정확성 증명(π)을 생성한다. 실시예에서, 증명은 자원(예를 들어, 디지털 자산)을 잠금해제하는데 사용되며, 또한 상대방 또는 증명자를 식별하는 정보(예를 들어, 상대방 또는 증명자와 연관된 공개 키 및/또는 디지털 서명)가 난독화되지 않은 포맷으로 블록체인에 기록되지 않는 경우일 수 있다.

[0071] 실시예에서, 검증 키 및 대응하는 증명은 위에서 및/또는 아래에서 설명되는 기술에 따라 생성된다. 따라서, 검증자가 복수의 타원 곡선 승산(elliptic curve multiplication)(예를 들어, 각각의 공개 입력 변수마다 하나씩) 및 5 개의 페어 체크(pair check) - 그 중 하나가 추가 페어링 승산(additional pairing multiplication)

을 포함함 - 를 계산하도록, 검증자에게는 다음과 같은 검증 키(V_K) 및 증명(π)이 주어진다:

$$V_K = \left\{ \begin{pmatrix} \mathcal{P} \\ Q \\ \alpha_v Q \\ \alpha_w Q \\ \alpha_w \mathcal{P} \\ \alpha_y Q \\ \beta \mathcal{P} \\ \beta Q \\ r_y t(s)\mathcal{P} \\ r_v v_i(s)\mathcal{P} \\ r_w w_i(s) Q \\ r_y y_i(s)\mathcal{P} \end{pmatrix}_{i=0..N} \right.$$

[0072]

$$Proof\ \pi = \left\{ \begin{array}{c} \sum_{i=N+1}^{m} a_i r_v v_i(s)\mathcal{P} \\ \sum_{i=N+1}^{m} a_i \alpha_v r_v v_i(s)\mathcal{P} \\ \sum_{i=N+1}^{m} a_i r_w w_i(s) Q \\ \sum_{i=N+1}^{m} a_i \alpha_w r_w w_i(s)\mathcal{P} \\ \sum_{i=N+1}^{m} a_i r_y y_i(s)\mathcal{P} \\ \sum_{i=N+1}^{m} a_i \alpha_y r_y y_i(s)\mathcal{P} \\ \sum_{i=N+1}^{m} a_i (r_v \beta v_i(s) + r_w \beta w_i(s) + r_y \beta y_i(s))\mathcal{P} \\ \sum_{i=0}^{d} h_i s^i Q \end{array} \right\}$$

[0073]

[0074] t(x)가 p(x)를 제산하여 $(x_{N+1}, \ldots, x_m) = f(x_0, \ldots, x_N)$ 로 되는 것을 증명하기 위해, 검증 키(V_K), 증명(π), 및 $(a_1, a_2, \ldots, a_N)$ 이 주어지면, 검증자는 아래와 같이 진행한다. 먼저 다음의 3 개의 α항을 모두 체크하며:

$$e(\alpha_v r_v V_{\text{mid}}(s)\mathcal{P}, Q) = e(r_v V_{\text{mid}}(s)\mathcal{P}, \alpha_v Q)$$
$$e(\alpha_w r_w W_{\text{mid}}(s)\mathcal{P}, Q) = e(\alpha_w \mathcal{P}, r_w W_{\text{mid}}(s) Q)$$

[0075] $$e(\alpha_y r_y Y_{\text{mid}}(s)\mathcal{P}, Q) = e(r_y Y_{\text{mid}}(s)\mathcal{P}, \alpha_y Q)$$

[0076] 여기서, $V_{\text{mid}}(s) = \sum_{i=N+1}^{m} a_i v_i(s)$, $W_{\text{mid}}(s) = \sum_{i=N+1}^{m} a_i w_i(s)$ 및 $Y_{\text{mid}}(s) = \sum_{i=N+1}^{m} a_i y_i(s)$ 이다. 그 다음, 검증자는 다음과 같은 항(β)을 체크한다:

[0077] $e\left(r_v V_{\text{mid}}(s)\mathcal{P} + r_y Y_{\text{mid}}(s)\mathcal{P}, \beta Q\right) \cdot e\left(\beta\mathcal{P}, r_w W_{\text{mid}}(s)Q\right) = e\left(Z_{\text{mid}}(s)\mathcal{P}, Q\right)$ 및 $Z_{\text{mid}}(s) = \sum_{i=N+1}^{m} a_i \left(r_v \beta v_i(s) + r_w \beta w_i(s) + r_y \beta y_i(s)\right)$. 마지막으로, 검증자는 다음과 같이 가분성 요건을 체크하고:

[0078] $$e(r_v V(s)\mathcal{P}, r_w W(s)Q) = e\left(r_y Y(s)\mathcal{P}, Q\right) \cdot e\left(r_y t(s)\mathcal{P}, h(s)Q\right),$$

[0079] 여기서 $r_v V(s)\mathcal{P} = \sum_{i=0}^{m} r_v a_i v_i(s)\mathcal{P}$, $r_w W(s)Q = \sum_{i=0}^{m} r_w a_i w(s)Q$, $r_y Y(s)\mathcal{P} = \sum_{i=0}^{m} r_y a_i y(s)\mathcal{P}$, 및 $h(s)Q = \sum_{i=0}^{d} h_i Q$ 이다.

[0080] 따라서, 위에서 설명한 섹션 및 본 개시내용에서 설명된 예로부터의 표기를 고려할 때, 검증은 하나의 실시예에 따라 다음과 같은 요소의 한 세트의 쌍의 체크를 포함한다:

$$e(\pi_2, V_K^2) = e(\pi_1, V_K^3)$$
$$e(\pi_4, V_K^2) = e(V_K^5, \pi_3,)$$
$$e(\pi_6, V_K^2) = e(\pi_5, V_K^6)$$
$$e((\pi_1 + \pi_6), V_K^2) = e(\pi_7, V_K^2)$$

[0081]
$$e\left((a_0 V_K^{10} + a_1 V_K^{11} + a_2 V_K^{12} + a_3 V_K^{13} + a_4 V_K^{14} + \pi_2 + a_7 V_K^{15}), (a_0 V_K^{16} + a_1 V_K^{17} + a_2 V_K^{18} + a_3 V_K^{19} + a_4 V_K^{20} + \pi_4 + a_7 V_K^{21})\right)$$
$$= e\left((a_0 V_K^{22} + a_1 V_K^{23} + a_2 V_K^{24} + a_3 V_K^{25} + a_4 V_K^{26} + \pi_6 + a_7 V_K^{15}), V_K^2\right)$$
$$* e(V_K^9, \pi_8)$$

[0082] 도 3은 검증 가능한 계산의 수행을 조정하기 위한 다이어그램(300)을 예시한다. 클라이언트(302), 증명자(304) 및 검증자(306)는 블록체인 네트워크의 노드일 수 있다. 클라이언트(302)는 컴퓨터 시스템의 하나 이상의 프로세서에 의해 실행될 때 컴퓨터 시스템이 스마트 계약(308)을 수신하게 하는 실행 가능 코드를 포함할 수 있는 임의의 적합한 컴퓨터 시스템일 수 있다. 실시예에서, 스마트 계약(308)은 C, C++ 또는 자바와 같은 소스 코드로서 고급 프로그래밍 언어로 인코딩된다. 실시예에서, 컴파일러, 인터프리터 및/또는 어셈블러와 같은 소프트웨어는 스마트 계약(308)을 필드($\mathbb{F}$)로부터 값을 운반하고 가산 및 승산 게이트에 연결하는 "와이어"로 구성된 산술 회로(310)로 변환하는데 이용될 수 있다. 산술 회로는 물리 와이어에 의해 연결된 (예를 들어, 7400 시리즈 게이트, 플립 플롭, 버퍼, 디코더, 멀티플렉서 등과 같은 트랜지스터-트랜지스터 로직(transistor-transistor logic)(TTL) 집적 회로를 사용하는) 일련의 물리 게이트를 포함하는 물리 회로에 의해 구현될 수 있는 논리 회로를 지칭할 수 있다는 것에 유의하여야 한다. 스마트 계약(308)의 실행이 도 3 및 다른 곳에서 설명되지만, 스마트 계약의 사용은 산술 회로로 변환될 수 있는 소스 코드의 비제한적인 예일 뿐이다. 실시예에서 (예를 들어, 단독으로 또는 다른 엔티티와 함께) 클라이언트(302)는 한 세트의 동작에 의해 정의된 작업을 수행하기 위한 소스 코드를 결정하며, 여기서 작업의 실행은 증명자(304)에게 위임된다. 일반적으로 말하면, 검증자(306)는, 예를 들어, 증명자(304)에 의해 생성된 정확성 증명의 유효성을 검증함으로써, 증명자(304)가 작업을 올바르게 실행했다고 결정하는 것과 연관된 작업을 수행할 수 있다.

[0083] 실시예에서 클라이언트(302)는 산술 회로(310)를 증명자(304)에게 제공하고 데이터 제공자(318)는 회로로의 입력(312)을 증명자에 제공한다. 일부 경우에, 입력(312)은 실세계 상태 및 이벤트에 관한 데이터와 같은 데이터일 수 있다. 회로(310)는 원래 회로의 완전한 설명을 제공하는 한 세트의 다항식을 포함하는 이차 프로그램

($\mathcal{Q}$)을 생성하는데 사용될 수 있다. 어느 경우이든, 증명자(304)는 회로($\mathcal{C}$) 또는 입력(312) 상의 이차 프로그램($\mathcal{Q}$)을 실행하여 하나 이상의 출력 중간 출력 및 하나의 최종 출력을 생성할 수 있다. 일부 실시예에서, 증명자는, 입력 와이어에 할당된 값이 x의 값이고, 중간 값이 $\mathcal{C}$의 각각의 게이트의 올바른 동작에 대응하고, 출력 와이어(들)에 할당된 값이 y이도록 회로 와이어에 값을 할당하는 $\{\mathcal{C}, x, y\}$에 대한 유효 트랜스크립트(valid transcript)를 출력으로서 획득하는 것으로 예상되며; 청구된 출력이 올바르지 않으면(즉, $y \neq \mathcal{P}(x)$), $\{\mathcal{C}, x, y\}$에 대한 유효 트랜스크립트가 존재하지 않는다. 실시예에서 증명자는 회로 와이어의 값의 서브세트를 제공할 것으로 예상되며, 여기서 회로 와이어의 값의 선택된 서브세트는 선험적으로 증명자에게 알려지지 않는다.

[0084] 실시예에서, 출력(y), 내부 회로 와이어의 값(또는 그 서브세트), 및 평가 키(E_K)는 정확성 증명(316)을 생성하는데 사용된다. 증명(π)은 블록체인에 저장될 수 있고, 증명자(304)가 다수의 당사자와 개별적으로 상호작용할 필요없이 다수의 당사자에 의해 검증될 수 있다. 이러한 방식으로, 검증자(306)는 공개 검증 키(V_K) 및 증명(π)을 사용하여 잠금 트랜잭션을 유효화할 수 있고, 그럼으로써 계약을 유효화할 수 있다. 일부 경우에, 클라이언트(302)는 검증이 실패하면 잠금 트랜잭션에 의해 담보되는 디지털 자산을 재청구할 수 있다. 일부 경우에, 검증자(306) 및 클라이언트(302)는 동일한 컴퓨터 시스템이다.

[0085] 도 4는 본 개시내용의 실시예의 다이어그램(400)을 예시한다. 구체적으로, 도 4는 데이터 소스에 의해 제공된 데이터가 블록체인 네트워크에 게시된 스마트 계약과 같은 프로그램 또는 스크립트의 실행시 인증되고 이용되는 환경을 예시한다. 실시예에서, 클라이언트(402), 증명자(404) 및 데이터 제공자(406)는 컴퓨터 시스템이다. 실시예에서, 클라이언트(402)는 본 명세서에서 컴퓨팅 작업의 수행을 증명자라고 지칭되는 다른 컴퓨터 시스템에 아웃소싱 또는 위임하는 컴퓨터 시스템을 지칭한다. 실시예에서, 계산 작업은 함수의 평가, 스마트 계약의 실행 등을 지칭한다. 실시예에서, 계산 작업을 수행하는 데는 회로 및 하나 이상의 입력에 기초한 출력의 계산이 수반되며, 계산은 클라이언트를 대신하여 증명자에 의해 수행된다. 회로는 계약 조건의 공식적인 표현인 소스 코드로부터 생성될 수 있다.

[0086] 본 개시내용의 다른 곳에서 더 상세히 설명된 바와 같이, 블록체인 상에서 스마트 계약을 실행하기 위한 프로토콜은 제로 지식 증명을 이용할 수 있다. 실시예에서, 스마트 계약의 실행은 트랜잭션 유효화의 일부로서 발생할 수 있다. 이를 위해, 프로토콜의 설정 단계에서 생성된 공개 파라미터는 증명 및 입력/출력 데이터와 함께 증명자에 의해 제공되는 주장된 정확성 증명 계산을 검증하고 계약 실행을 유효화하는데 사용된다. 계약의 실행 단계는 일부 실시예에서 블록체인 외부의 입력/출력 데이터에 의존할 수 있다. 일반적으로 말하면, 외부 데이터는 블록체인 상에서 사용할 수 없거나 블록체인을 통해 액세스 가능한 데이터로부터 검증할 수 없는 정보를 지칭한다. 외부 데이터는 데이터 피드 및 애플리케이션 프로그래밍 인터페이스(Application Programming Interface)(API)와 같은 다양한 소스로부터 획득될 수 있다. 외부 데이터는, 이것으로 제한되는 것은 아니지만, 자산 및 금융 애플리케이션에 대한 시장 가격 피드(예를 들어, 금리)에 액세스해야 하는 유가 증권(예를 들어, 이자율, 파생 상품, 채권)에 사용하기 위한 스마트 계약; 외부 데이터(예를 들어, 지연에 대비하여 당사자가 보장하는 항공편 정보; 농작물 보험에 대한 가격 지수 대신에 날씨의 데이터 피드를 사용하는 금융 파생 상품 계약에 대한 날씨 데이터)에 액세스해야 하는 피어 투 피어 보험 스마트 계약; 선적에 관한 GPS 데이터가 필요한 무역 스마트 계약; 난수 생성기에 액세스해야 하는 도박 계약; 등을 비롯한 스마트 계약을 둘러싼 다양한 맥락에서 사용될 수 있다.

[0087] 실시예에서, 스마트 계약을 실행하는 시기 및/또는 방법을 결정하는 것은 실세계 상태 및 이벤트에 관한 데이터와 같은 외부 데이터로의 액세스에 의존한다. 실시예에서, 데이터 제공자(406)는 이러한 데이터를 제공하며, 일부의 경우 스마트 오라클(smart oracle)로 지칭될 수 있다. 오라클에 의해 반환된 데이터는 당사자에 의해 명시된 특정 정보가 명시된 텍스트 또는 명시된 정보를 포함하는지를 표시하는 부울의 형태와 같은 다양한 포맷 및 유형일 수 있다. 실시예에서, 이러한 데이터는 베팅(예를 들어, 복싱 경기의 결과와 같은 이진 이벤트의 결과)을 구성하는데 사용될 수 있다. 부동 소수점 값은 (암호 화폐를 비롯한) 통화 간의 환율의 맥락에서 이용될 수 있으며, 일부 비트코인 거래소의 API로부터 판독될 수 있다. 이러한 유형의 데이터는 옵션을 구성하고 접촉을 헤지(hedge)하는데 유용할 수 있다.

[0088] 데이터 제공자(406)는 또한 실세계 상태 및 이벤트에 관한 데이터와 같은 외부 데이터를 검색 및/또는 제공하고 외부 데이터를 블록체인에 이용 가능하게 하는 오라클로 지칭될 수 있다. 오라클은 컴퓨터 시스템의 하나 이상의 프로세서 상에서 실행될 때 컴퓨터 시스템으로 하여금 블록체인과 네트워크(예를 들어, 인터넷) 사이의 신뢰할 수 있는 링크로서 작용하는 블록체인(예를 들어, 비트코인, 이더리움 등)과 호환 가능한 데이터 및/데이터 피드를 생성하게 하는 실행 가능 코드를 포함하는 컴퓨팅 엔티티를 지칭할 수 있다. 오라클은 웹 사이트, 피드, 데이터베이스 및 다른 데이터 소스로부터 데이터를 획득하여 외부 데이터를 블록체인에 제공할 수 있다. 외부 데이터는 예를 들어 온도 정보, 선적의 GPS 좌표, 항공 정보(예를 들어, 비행 상태 정보) 등을 포함할 수 있다.

[0089] 실시예에서, 증명자(404)는, 데이터 제공자와 암호로 보호된 통신 세션을 수립하고; 암호로 보호된 통신 세션을 통해 요청을 작성함으로써 데이터 제공자(406)로부터 외부 데이터를 획득하고; 데이터 제공자는 요청에 응답하여 데이터를 제공하고; 증명자는 데이터를 수신하고, 데이터 수신에 응답하여 당사자 사이의 통신의 입증을 요청하고; 데이터 제공자는 암호로 보호된 통신 세션 동안 증명자와 데이터 제공자 사이의 암호로 검증 가능한 통신의 증명($\pi_{Communications}$)을 계산하고 데이터 제공자의 개인 키로 입증에 디지털 서명하며; 증명자는 통신의 증명을 수신한다. 일반적으로 말하면, 통신의 증명은 하나 이상의 통신이 클라이언트와 서버 사이에서(예를 들어, 증명자와 데이터 제공자 사이에서) 발생한 암호로 검증 가능한 입증이다. 실시예에서, 일부 경우에 통신의 일부가, 이를테면 개정된 정보를 정보의 난독화(예를 들어, 암호화되거나 해시된 포맷의 정보)로 대체하거나 미리 결정된 디폴트 값으로 대체함으로써, (예를 들어, 공개에 법적 제한이 적용되는 정보로) 개정될 수 있다는 것에 유의해야 하지만, 입증은 클라이언트와 서버 사이의 통신의 내용을 검증하는데 사용될 수 있다. 실시예에서, 입증은 도 6에 따라 설명된 바와 같은 머클 트리의 루트 값에 적어도 부분적으로 기초하여 결정된다. 실시예에서, 입증(예를 들어, 통신의 증명)은 데이터 제공자(406)에 액세스 가능한 암호화 개인 키를 사용하여 디지털 서명된다. 인증 기관과 같은 엔티티는 암호화 공개 키가 개인 키에 대응한다는 것을 인증하는 디지털 인증서를 발행할 수 있다. 본 개시내용의 범위에서, 통신의 증명은 일반적으로 표기 $\pi_{Communications}$을 사용하여 지칭되는 반면, 올바른 실행의 증명은 π_{Prover} 또는 더 간단히 π로 지칭될 수 있음에 유의하여야 한다.

[0090] 인증 기관(410)은 특정 암호화 키가 특정 엔티티에 속한다는 것을 인증하는 디지털 인증서를 생성 및/또는 배포하는 컴퓨팅 엔티티를 지칭할 수 있다. 예를 들어, 데이터 제공자의 공개 키를 간직하는 디지털 인증서는 컴퓨팅 엔티티에 의해 획득될 수 있으며, 이러한 공개 키는 다양한 맥락에서 - 예를 들어, 데이터 제공자와의 안전한 통신을 위해 데이터를 암호화하고, 데이터 제공자가 서명한 디지털 서명의 진위성을 검증하는 등에 이용될 수 있다. 도 4는 클라이언트(402)가 인증 기관(410)과의 신뢰 관계를 갖는(예를 들어, 클라이언트는 인증 기관에 의해 적절하게 서명 및/또는 발행된 디지털 인증서를 인증된 것으로 취급하는) 실시예를 예시하지만, 키 관리 시스템 또는 키 레지스트리와 같은 임의의 적합한 신뢰할 수 있는 컴퓨팅 엔티티가 이용될 수 있다.

[0091] 실시예에서, 증명자(404)는 입력 데이터를 이용하여, 예를 들어 도 2와 관련하여 위에서 설명한 방식으로 π_{Prover}를 생성한다. 실시예에서, π_{Prover}는 추후 사용을 위해 블록체인(408)에 저장되거나, 또는 증명자가 이들 당사자 각각과 개별적으로 상호작용할 필요없이 다수의 당사자에 의해 검증될 수 있다. 예를 들어, 비행 지연에 대비하여 당사자를 보장하기 위한 스마트 계약의 실행은 공식 항공 당국(예를 들어, 미국 연방 항공국 또는 영국 민간 항공 당국)으로부터 입력 데이터로서 획득하는 공식 비행 데이터에 의존할 수 있다. 일반적으로 말하면, 입력 데이터가 정부 소스로부터 받을 필요는 없으며 - 스마트 계약의 당사자가 입력 데이터를 획득할 수 있는 엔티티 리스트에 동의할 수 있다 - 따라서, 데이터 소스는 클라이언트가 데이터를 획득하는데 신뢰할 수 있다는 것을 나타내는 임의의 적합한 컴퓨팅 엔티티일 수 있다. 클라이언트는 상이한 입력에 대해 신뢰할 수 있는 엔티티의 상이한 리스트를 명시할 수 있다. 신뢰할 수 있는 엔티티의 리스트는 계약의 가치에 따라 선택 및/또는 정리될 수 있다.

[0092] 증명자(404)는 통신의 증명을 클라이언트(402)에 송신할 수 있다. 일부 경우에, 데이터 제공자(406)는 - 예를 들어, 증명자로부터의 커맨드에 기초하여 통신의 증명을 클라이언트(402)에 송신할 수 있다. 이전에 설명한 바와 같이, 통신의 증명은 머클 트리의 루트 값에 적어도 부분적으로 기초하여 도출될 수 있다. 예를 들어, 통신의 증명은 또한 통신이 발생한 시간 간격을 명시하는 정보를 인코딩할 수 있다. 실시예에서:

[0093] $$W_{Time} = [T_{SessionStart}, T_{ProofRequest}]$$

[0094] $$\pi_{Communications} = H(h'_{final}, W_{Time})$$

[0095] 다시 말해, 통신의 증명은 암호화 해시 함수를 사용하여 계산되며, 여기서 머클 트리의 루트 값(h'_{final}) 및 통신의 시간 간격(실시예에서, 암호화 통신 세션의 시작 시간 및 통신의 증명이 요청된 시간을 인코딩하는 W_{Time})은 암호화 해시 함수에 입력된다. 실시예에서, 암호화 해시 함수의 입력은 연결되거나 그렇지 않으면 결합된다. 머클 트리는 W_{Time} 에 의해 표시된 시간 간격 동안 당사자 사이에 교환된 레코드에 적어도 부분적으로 기초하여 생성될 수 있다. 통신의 증명은 데이터 소스의 암호화 개인 키를 사용하여 디지털 서명될 수 있다. 실시예에서, 대화가 발생하는 시간은 $W_{Time} = [T_{SessionStart}, T_{ProofRequest}]$ 로 정의되고 서버(예를 들어, 데이터 소스(504))에 의해 생성된 통신의 증명의 일부로서 포함된다. 통신의 증명을 클라이언트(402)에 송신하는 것에 부가하여 및/또는 대안으로, 증명자(404)는 보안 서버에 또는 컴퓨팅 자원 서비스 제공자의 데이터 저장 서비스에 (예를 들어, 클라이언트(402)와 연관된 공개 키에 의해 암호화된) 증명을 저장할 수 있다는 것에 유의하여야 한다. 실시예에서, 클라이언트(402)는 데이터 저장소(예를 들어, 데이터 저장 서비스)에 통신의 증명을 저장하고 클라이언트(404)에게 통합 자원 식별자(Uniform Resource Identifier)(URI) 또는 데이터 저장소로부터 통신의 증명을 획득하기 위해 사용 가능한 다른 적절한 기준을 제공한다. 예를 들어, (예를 들어, 며칠에 걸쳐 하나 이상의 경로에 대한 비행 데이터를 캡처하는) 반복된 작업을 수행하는 증명자(402)는, 클라이언트가 시간 간격에 따라 (예를 들어, 특정 경로에 대한) 이력 비행 데이터를 획득할 수 있도록, 캡처된 데이터를 데이터 저장소의 컨테이너에 집계된다.

[0096] 실시예에서, 클라이언트(402)는 통신의 증명 및 통신의 증명에 걸쳐진 디지털 서명을 수신한다. 클라이언트는 디지털 서명이 데이터 제공자에 의해 의도적으로 서명된 것으로 결정하고, (예를 들어, 인증 기관(410)에 의해 발행된 디지털 인증서로부터) 데이터 제공자와 연관된 공개 키를 획득하고, 공개 키를 사용하여 디지털 서명의 진위를 검증할 수 있다. 디지털 서명이 진본인 것으로 결정되면, 클라이언트(402)는 통신의 증명이 데이터 제공자에 의해 생성된 것으로 신뢰하고; 그렇지 않으면, 통신의 증명 및 디지털 서명은 폐기될 수 있다. 통신의 증명을 검증한 후에, 클라이언트(402)는 예를 들어 도 2 및 도 3과 관련하여 논의된 방식으로 정확성의 증명을 검증할 수 있다.

[0097] 일부 실시예에서, 클라이언트(402)는 특정 증명자(404)로부터 획득된 외부 데이터가 정확하다(예를 들어, 데이터가 데이터 제공자에 의해 제공된 동일한 데이터이다)는 것을 표시하는 평판 트랜잭션을 생성한다. 평판 트랜잭션은 증명자(404)가 계산 작업을 올바르게 수행했는지의 여부 및/또는 증명자(404)가 데이터 제공자로부터 수신된 외부 데이터를 사용하여 π_{Prover} 의 계산을 수행하는 것에 대해 정직한 지의 여부에 대한 레코드로서 블록체인(408)에 채굴될 수 있다. 스마트 계약이 실행된 후에 클라이언트는 π_{Prover}를 생성하는데 사용된 입력 데이터가 데이터 공급자에 의해 제공된 입력 데이터인지를 검증하기 위해 $\pi_{Communications}$ 를 사용할 수 있다. 평판 트랜잭션은 증명자에 의해 제공된 서비스에 대한 검토를 표시하는 블록체인에 채굴될 수 있으며 - 긍정적 결과는 증명자가 정직하게 행동하도록 장려할 수 있으며 후속 스마트 계약의 실행을 위한 증명자를 선택하는데 사용될 수 있다. 평판 트랜잭션은 π_{Prover} 에 사용된 입력 데이터가 데이터 제공자로부터 제공되었다는 충분한 증명을 제공할 때 증명자가 평판을 잠금해제할 수 있게 하는 제 1 잠금해제 스크립트 및 클라이언트가 검토를 취소할 수 있는 제 2 잠금해제 스크립트를 포함할 수 있다. 도 4에 예시된 평판 트랜잭션은 도 9와 관련하여 아래에 설명되는 바와 같은 본 개시내용의 다른 곳에서 설명된 것에 따를 수 있다.

[0098] 일반적으로 말하면, 도 4의 맥락에서, 클라이언트는 타겟 소스(src), 시간(τ) 및 질의(q)를 명시하는 요청($Req(src, \tau, q)$)을 작성한다. src 가 웹 서버이면, TLS 세션과 같은 암호로 보호된 통신 세션이 필요할 수 있다. 실시예에서, 진위성의 암호로 검증 가능한 보증을 제공하는 임의의 암호로 보호된 통신 세션이 이용될

수 있으며, 여기서 당사자 사이의 메시지는 동일한 비밀 키를 사용하여 생성 및 검증되는 메시지 인증 코드(message authentication code)(MAC)를 통해 인증된다. 이것은 서버 및 메시지의 수신자가 동일한 키에 동의해야 하므로, MAC을 검증할 수 있는 모든 사용자는 다른 메시지에 대해서도 MAC을 생성할 수 있다는 것을 암시한다. 대조적으로, 디지털 서명은 특정 메시지가 특정 개인 키의 소유자에 의해 서명되었다는 것을 제공할 수 있다. Hajjeh 및 M. Badra에 의해, 서명된 데이터의 전송을 시작하거나 중지할 때 클라이언트와 서버가 피어에게 통지하는 TLSSignOnOff라 불리는 새로운 하위 프로토콜을 정의하는 "TLS Sign"에서 통신을 공증하는 접근 방식이 고려되었다. 중지 메시지 이후, 서버는 대화의 해시를 수집하여 서명한다. R. Housley 및 M. Brown의 "Transport Layer Security(TLS) Evidence Extensions"와, H. Ritzdorf, K. Wust, A. Gervais, G. Felley 및 S. Capkun의 "TLS-N: Non-repudiation over TLS Enabling Ubiquitous Content Signing for Disintermediation"에 의해 설명된 기술 - 클라이언트가 요청을 작성할 때 그리고 당사자 중 하나가 증명 창을 닫을 때 시작하는 증명 창을 정의하고, 메시지의 해시 및 증명 생성의 타임스탬프가 서버에 의해 서명되며, 선택적으로 민감한 레코드는 숨겨짐 - 과 같이, 암호로 보호된 통신 세션의 레코드 데이터를 공증하기 위한 (예를 들어, 세션의 레코드를 저장하고, 기록된 대화의 진위성 및/또는 무결성의 암호로 검증 가능한 증명으로 대화에 서명하기 위한) 다양한 다른 접근법이 사용될 수 있다.

[0099] 도 5는 증명자(502)가 데이터 소스(504)로부터 데이터를 획득하는 실시예의 예시적인 도면(500)이다. 실시예에서, 암호로 보호된 통신 세션은 증명자(502)와 데이터 소스(504) 사이에 수립된다. 암호로 보호된 통신 세션의 예는 전송 계층 보안(Transport Layer Security)(TLS) 및 보안 소켓 계층(Secure Sockets Layer)(SSL) 세션을 포함한다. 암호로 보호된 통신 세션은 당사자 사이에 보안 연결을 제공하는데 이용될 수 있다. 이러한 맥락에서, 보안 연결은 진위성(authenticity), 무결성(integrity), 프라이버시(privacy) 또는 이들의 임의의 조합의 암호로 검증 가능한 보증을 지칭할 수 있다. 진위성은 메시지 작성자라고 칭하는 당사자에 의해 메시지가 작성되었다는 보증을 지칭할 수 있다. 무결성은 수신된 메시지가 메시지가 송신될 때의 원래 형태로부터 (예를 들어, 악의적인 당사자에 의해) 의도적으로 또는 (예를 들어, 송신 동안의 신호 손실의 결과로) 의도하지 않게 수정되지 않았다는 보증을 지칭할 수 있다. 비밀성(confidentiality)은 송신 전 메시지의 일부 또는 전부를 암호화하는 것을 지칭할 수 있다. 실시예에서, 클라이언트(예를 들어, 증명자(502)) 및 서버(예를 들어, 데이터 소스(504))는 핸드셰이크(508)로 지칭되는 제 1 단계 동안 정보를 교환하며, 여기서 보안성을 보장하기 위해 비대칭 및 대칭 암호화의 하이브리드가 발행된다. 도 5에 예시된 핸드셰이크는 TLS 핸드셰이크 프로토콜에 따른 4-방향 핸드셰이크일 수 있다. 실시예에서, 클라이언트(예를 들어, 증명자(502))는 클라이언트 버전, 세션 ID, 추가된 확장을 통한 추가적인 기능성, 또는 이들의 일부 조합과 같은 파라미터를 포함하는 ClientHello 메시지를 송신한다. 실시예에서, 핸드셰이크는 서버(예를 들어, 데이터 소스(504))를 향한 신호를 메시지로 시작하여 대화에 대한 증명의 기록을 시작한다. 결과적으로, 서버(예를 들어, 데이터 소스(504))는 후속 요청을 수신하여 통신의 증명을 생성할 때까지 지속되는 증명 생성과 연관된 내부 상태에 진입할 것이다. 이러한 내부 상태에 있는 동안, 데이터 소스(504)는 암호로 보호된 통신 세션을 통해 교환된 데이터 소스(504)와 증명자(502) 사이의 통신의 내용(또는 그 일부)을 기록할 수 있다.

[0100] 핸드셰이크 이후(예를 들어, 레코드 프로토콜 동안), 발신 메시지는 블록으로 분할되는 반면에, 착신 메시지는 재조립된다. 실시예에서, 암호로 보호된 통신 세션의 당사자 사이의 대화는 복수의 레코드($R_1 \dots R_i$)를 포함한다. 이들 레코드는, 이를테면 레코드 $R_1 \dots R_i$ 를 포함하는 한 세트의 리프 노드(leaf node)를 포함하는 머클 트리를 생성함으로써, 통신의 증명을 생성하는데 이용될 수 있다. 도 5와 관련하여 예시되고 설명된 레코드는 도 6과 관련하여 설명된 것과 같은 머클 트리를 생성하는데 사용될 수 있다.

[0101] 실시예에서, 레코드는 암호로 보호된 통신 세션의 일부로서 증명자(502)와 데이터 소스(504) 사이의 대화를 형성하는 요청 및 응답을 지칭한다. 일부 경우에, 요청 및 응답은 필요한 것보다 많은 정보(예를 들어, 스마트 계약의 실행과 관련하여 증명자(502)가 필요로 하는 입력 데이터를 결정하는 것과 관련이 없는 정보)를 포함한다. 실시예에서, 서버(예를 들어, 데이터 소스(504))는 (예를 들어, 특정 필드를 갖는 JSON 노드의 형태로) 템플릿을 채운다. 템플릿은 핸드셰이크 동안 명시될 수 있다. 예로서, 날씨 데이터를 획득하기 위한 템플릿은 위치, 날짜 및 온도에 관한 필드를 가질 수 있다. 실시예에서, 비행 데이터에 대한 요청은 예를 들어, 다음과 같은 포맷으로, 비행 상태, 비행 식별자 및 날짜를 포함할 수 있다:

{"flightInfo":{"data":{"id":"BA886":[{"validDate":"2017-11-01T07:00:00+0000"},"status

[0102] ":"On time"}]}.

[0103] 핸드셰이크가 완료되면, 당사자(예를 들어, 증명자(502) 및 데이터 제공자(504))는 데이터를 포함하는 레코드(510)를 서로 간에 송신할 수 있다. 예를 들어, 증명자(502)는 데이터 제공자(504)로부터, 입력으로서 회로를 평가하는데 사용될 수 있고 스마트 계약의 실행의 일부로서 출력을 생성할 수 있는 데이터를 요청할 수 있다. 증명자(502)가 회로를 평가하기에 충분한 데이터를 수신한 후에, 증명자는 데이터 제공자(504)로부터 세션의 증명(512)을 요청할 수 있다. 데이터 제공자(504)는 클라이언트(예를 들어, 도 3 및 도 4와 연관되어 설명된 클라이언트)와 신뢰 관계를 갖는 컴퓨터 시스템일 수 있으며, 여기서 클라이언트는 데이터 제공자에 의해 생성 및/또는 제공되는 데이터를 신뢰한다 - 이러한 맥락에서, 다른 엔티티로부터의 데이터를 신뢰하는 클라이언트는 엔티티로부터 획득된 데이터를 올바른 것으로서 수락하는 클라이언트를 지칭하며 및/또는 유효한 보안 인증서를 가진 신뢰할 수 있는 소스로부터 게시된 데이터는 사실이고 정확하다. 실시예에서, 서버(예를 들어, 데이터 소스(504))는 핸드셰이크 프로토콜의 완료로부터 $\pi_{Communications}$ 를 생성하라는 요청이 수신될 때까지 레코드 프로토콜 동안 당사자 사이에 교환된 모든 레코드의 해시 루트(514)로 구성되는 통신의 증명(516)을 생성한다. 실시예에서, W_{Time} 은 또한 데이터의 실행이 특정 시간에 수집되는 입력(예를 들어, 특정 날짜의 특정 항공편 번호의 지연을 방지하는 항공 보험에 대한 스마트 계약)에 의해 좌우되는 경우에서와 같이, 레코드로 해시된다.

[0104] 실시예에서, 클라이언트(예를 들어, 증명자(502))는 $\pi_{Communications}$ 를 요청하고, 데이터 제공자는 데이터 소스의 개인 키와 연결되고, 해시되고 디지털 서명된 W_{Time} 및 모든 레코드의 커밋(commitment)으로부터 획득된 해시 체인의 최종 값을 반환한다. 따라서, 당사자 사이에서 통신된 레코드에 포함된 데이터는 스마트 계약의 실행에 이용될 수 있다. 이러한 통신 증명($\pi_{Communications}$)는 또한 초기 스마트 계약에 전달되고 전용 기능을 통해 검증될 수 있으며, 따라서 반쯤 신뢰할 수 있는 증명자에게는 필요하지 않다. 실시예에서, 레코드는 도 6과 관련하여 설명된 방식과 같은 머클 구조 내에 함께 해시된다.

[0105] 세션 클라이언트가 세션 서버로부터 증명을 요청하는 실시예가 도 5와 관련하여 설명되지만, 그러한 경우일 필요는 없다는 점에 유의하여야 하며 - 본 명세서에 설명된 기술은 TLS 세션의 서버가 TLS 세션의 클라이언트로부터 통신의 증명을 요청하도록 적응될 수 있다.

[0106] 도 6은 본 개시내용에서 설명된 다양한 실시예에 따라 구성 및/또는 이용될 수 있는 머클 트리의 예시적인 다이어그램(600)이다. 예를 들어, 이러한 머클 트리는 도 2 내지 도 5와 연관되어 설명된 실시예와 관련하여 구성될 수 있다.

[0107] 실시예에서, 클라이언트 컴퓨터 시스템 및 서버 컴퓨터 시스템은 TLS 세션과 같은 암호화 보호 통신 세션을 수립하고 메시지(예를 들어, 레코드)를 교환한다. 세션의 당사자 사이에서 통신된 하나 이상의 레코드($R_1 \dots R_i$)는 지정된 기간 동안 - 예를 들어, 핸드셰이크 프로토콜 완료부터 서버가 클라이언트로부터 통신의 증명을 생성하기 위한 요청을 수신할 때까지 서버에 의해 기록될 수 있다. 실시예에서, 서버는 서버의 하나 이상의 프로세서에 의해 실행될 때, 서버로 하여금: 클라이언트와 서버 사이의 통신을 기록하기 시작한다는 표시를 검출하고; 암호로 보호된 통신을 통해 송신 및 수신된 레코드를 저장하고; 통신의 증명을 생성하기 위한 표시를 검출하고; 저장된 레코드에 적어도 부분적으로 기초하여 머클 트리를 생성하게 하는 실행 가능 코드를 포함한다. 위에서 설명한 프로세스에 대한 다양한 대안 및 확장이 고려된다. 머클 트리는 머클 트리의 각각의 리프 노드가 데이터 레코드의 해시이며 각각의 비 리프 노드(non-leaf node)는 적어도 암호화 해시 함수를 사용하여 비 리프 노드의 자식 노드(child node)로부터 적어도 일부에 기초하여 도출되는 데이터 구조를 지칭할 수 있다. 그러나, 본 명세서에 포함된 기술은 머클 트리와 유사한 구조를 이용하여 특정 데이터가 머클 트리의 일부라는 것을 검증할 수 있다는 점에 유의해야 하며 - 예를 들어, 실시예에서, 본 명세서에 포함된 기술은, 트리의 리프 노드가 데이터 레코드이고 비 리프 노드가 적어도 암호화 해시 함수를 사용하여 비 리프 노드의 자식 노드로부터 적어도 일부에 기초하여 도출되는 구조에 적용된다. 실시예에서, 본 명세서에 설명된 트리 데이터 구조는 이진 트리이다. 도 6은 불균형 머클 트리의 예를 예시하지만, 일부 실시예에서, 머클 트리의 리프 노드는 트리를 균형을 맞추거나 실질적으로 균형을 맞추는 방식으로 배치된다(예를 들어, 실질적으로 균형이 맞추어진 트리는 트리의 특정 깊이에 모든 리프를 가질 수 있다).

[0108] 실시예에서, 머클 트리는 저장되는(예를 들어, 서버에 의해 RAM과 같은 단기 메모리에 캐시되는) 하나 이상의

레코드($R_1 \dots R_i$)로부터 생성되고, 통신의 증명을 생성하라는 표시를 검출하는 즉시, 레코드는 머클 트리의 리프 노드로서 시간 순서대로 선택된다. 각각의 레코드는 TLS 프리-마스터 시크릿(pre-master secret) 또는 마스터 시크릿과 같은 핸드셰이크 트래픽 시크릿(handshake traffic secret)으로부터 파생될 수 있는 랜덤 솔트 값(random salt vaule)으로 암호로 해시된다. 솔트는 임의의 적합한 확률론적 프로세스를 사용하여 생성된 난수 또는 의사 난수일 수 있다. 솔트는 솔트가 특정 머클 트리의 맥락 내에서 재사용되지 않도록 선택될 수 있거나, 또는 임의의 2 개의 솔트가 동일한 값일 확률이 임계 확률 미만이 되는 방식으로 선택될 수 있다.

[0109] 제 1 솔트 및 제 1 레코드는 암호화 해시 함수로의 입력일 수 있으며, 여기서 제 1 솔트 및 제 1 레코드는 (예를 들어, SHA 256 암호화 해시 알고리즘을 사용하여) 연결되고 해시된다. 일반적으로 말하면, 암호화 해시 함수가 예시된 경우 일방향 함수와 같은 프리-이미지 저항성 함수(pre-image resistant function)가 이용될 수 있다. 도 6에 도시된 예시적인 예에서 솔트가 이용되지만, 레코드는 일반적으로 말하면, 논스(nonce), 초기화 벡터 및 임의의 다른 적절한 암호화 프리미티브(cryptographic primitive)로 증분될 수 있다. 제 1 솔트 및 제 1 레코드를 입력으로서 사용하여 암호화 해시 알고리즘을 수행함으로써 제 1 출력이 생성된다. 제 1 출력은 제 1 레코드가 세션의 클라이언트 또는 서비스에 의해 송신되었는지를 표시하는 정보로 해시될 수 있다. 예를 들어, 도 6은 클라이언트와 연관된 정보가 해시 출력에 접두(prepend)되는 실시예를 예시하며, 여기서 클라이언트는 레코드의 전송자이고 서비스와 연관된 정보는 해시 출력에 추가되고, 여기서 서비스는 레코드의 전송자이다. 접두된/추가된 정보는 임의의 적절한 정보일 수 있고 - 고정된 클라이언트 값(예를 들어, 0) 및 고정된 서버 값(예를 들어, 1)이 이용될 수 있다:

[0110] $$h_{R_1} = H(0, H(salt_{C1}, R_1))$$

[0111] $$h_{R_2} = H(H(salt_{C2}, R_2), 1)$$

[0112] 실시예에서, 트리 구조는 고정되고, 클라이언트 또는 서비스와 연관된 값(예를 들어, 전송자와 연관된 IP 주소 또는 MAC 주소)이 머클 트리에 접두/추가된다. (도 6에 예시되지 않은) 이러한 하나의 예에서:

[0113] $$h_{R_1} = H("client", H(salt_{C1}, R_1))$$

[0114] $$h_{R_2} = H("server", H(salt_{C2}, R_2))$$

[0115] 암호로 보호된 통신 세션을 통해 송신된 추가 레코드는 위에서 설명한 방식으로 작성되고 순차적으로 함께 해시되어 루트 노드 값을 생성할 수 있다. 도 6에 예시된 예에서, 각각의 노드는 중간 노드 값을 생성한다:

[0116] $$h'_{R_1} = H(h_{R_1})$$

[0117] $$h'_{R_i} = H(h_{R_{i-1}}, h_{R_i}) \ for \ i > 1$$

[0118] (도 6에 예시되지 않은) 실시예에서, $h'_{R_1} = h_{R_1}$.

[0119] 실시예에서, 최종 해시 값은 $h'_{R_{last}} = H(h_{R_{last-1}}, h_{R_{last}})$로서 계산되고 머클 트리의 루트이다. 실시예에서, $h'_{R_{last}}$는 통신의 증명이다. 실시예에서, $h'_{R_{last}}$는 통신의 증명을 형성하기 위해 함께 해시될 수 있는 시간 간격(예를 들어, W_{Time}, R_1 및 R_{last}에 대한 타임스탬프)과 같은 추가 정보로 증분된다. 통신의 증명은 암호화 통신 세션의 서비스와 연관된 개인 키에 의해 디지털 서명될 수 있으며, 여기서 서비스는 (예를 들어, 도 3 및 도 4와 연관되어 설명된 바와 같은) 스마트 계약 클라이언트가 신뢰할 수 있는 엔티티이다.

[0120] 도 7은 본 개시내용에서 설명된 다양한 실시예에 따른 머클 경로의 예시적인 다이어그램(700)이다. 머클 경로는 암호로 보호된 통신 세션의 클라이언트와 서비스 사이의 대화 동안 특정 레코드(또는 보다 일반적으로 특정 정보 또는 특정 데이터)가 통신되었다는 암호로 검증 가능한 보증을 제공하는데 이용될 수 있다. 머클 경로는 도 6과 관련하여 설명된 머클 트리로부터 구성될 수 있고 도 2 내지 도 5와 연관되어 설명된 실시예와 관련하여 이용될 수 있다.

[0121] 실시예에서, 증명자는 데이터 소스로부터 데이터를 수신하고 이 데이터를 입력으로서 사용하여 산술 회로를 해

결하고 회로의 출력에 적어도 부분적으로 기초하여 올바른 실행의 증명을 생성한다. 또한, 증명자는 산술 회로를 해결하는데 사용된 데이터가 증명자와 데이터 소스 사이의 한 세트의 통신 세트 내에 포함되었다는 입증을 생성할 수 있다. 실시예에서, 입증은 머클 경로이며, 머클 경로는 머클 트리의 루트 값을 계산하기에 충분한 머클 트리의 한 세트의 노드 값을 포함한다. 머클 경로는 머클 트리의 리프 노드 값 중 일부 또는 모두를 포함하거나 전혀 포함하지 않을 수 있다. 머클 경로는 머클 트리 내의 노드 값의 위치에 관한 정보를 포함할 수 있다. 루트 노드 값은 더 큰 머클 트리의 서브트리의 루트 노드 값일 수 있다는 점을 유의하여야 한다. 입증은 정확성의 증명(π_{Prover})로 인코딩될 수 있다.

[0122] 실시예에서, 컴퓨팅 엔티티는: 신뢰할 수 있는 엔티티로부터, 신뢰할 수 있는 엔티티의 개인 키를 사용하여 생성된 루트 노드 값에 대한 머클 트리의 루트 노드 값 및 디지털 서명을 수신함으로써; 신뢰할 수 있는 엔티티의 공개 키를 사용하여 디지털 서명의 진위성을 검증함으로써; 한 세트의 노드 값을 포함하는 머클 경로를 수신함으로써 - 노드 세트 중 하나의 노드는 통신된 것으로 알려진 특정 통신에 대응함 -; 노드 값 세트로부터의 추출된 루트 값을 계산함으로써; 및 추출된 루트 값을 검증된 루트 노드 값과 비교함으로써, 특정 통신의 내용이 2 개의 다른 컴퓨팅 엔티티 사이의 한 세트의 통신 내에 포함되었다는 것을 검증할 수 있다.

[0123] 실시예에서, 도 6과 관련하여 설명된 머클 트리에 기초하여 도출되는 도 7에 예시된 머클 경로(702)를 고려한다. 도 7에 예시된 머클 경로(702)는 특정 레코드(R_3)가 (예를 들어, TLS 세션을 통해) 클라이언트와 서버 사이의 통신 세트의 일부로서 포함되었다는 것을 검증하는데 이용될 수 있다. 머클 경로(702)의 값 세트는 함께 해시되어 최종 해시 값을 생성할 수 있다. 실시예에서, 최종 해시 값은 값이 함께 해시되는 순서를 나타내는 머클 경로에서 인코딩되거나 또는 머클 경로와 연관되어 인코딩된 추가 정보에 기초하여 생성된다. 예를 들어, 도 7에 예시된 머클 경로(702)는 도 7에 예시된 최종 값 $h_{R_4} = H\left(H\left(h'_{R_2}, H\left(0, H(salt_{C3}, R_3)\right)\right), h_{R_4}\right)$을 계산하는데 사용될 수 있다. 최종 값이 루트 값(예를 들어, 도 6의 $h'_{R_{last}}$)과 일치하면, 컴퓨팅 엔티티는 레코드(R_3)가 대화의 당사자 사이의 통신 세트 내에 통신으로서 포함되었다고 결정한다.

[0124] 일반적으로 말하면, 머클 경로는 루트 노드의 값을 계산하기에 충분한 머클 트리의 한 세트의 노드(또는 이들 노드의 값)를 지칭할 수 있다. 루트 노드는 도 7에 예시된 머클 트리의 루트 노드 또는 이러한 트리의 서브트리일 수 있다. 도 7에 예시된 머클 경로와 같은 머클 경로는 2 개의 리프 노드의 값과, 리프 노드와 루트 노드 사이의 머클 트리의 각각의 깊이에서 머클 트리의 정확히 하나의 노드에 대응하는 정확히 하나의 값을 인코딩할 수 있다. 그러나, 그럴 필요가 없고: 다른 적합한 머클 경로가 머클 트리의 각각의 리프 노드의 값을 포함한다 - 그러나, 이러한 머클 경로는 도 7에 예시된 머클 경로와 같은 다른 머클 경로보다 더 큰 저장 요건을 가질 수 있다.

[0125] 도 8은 다양한 실시예에 따른 프로토콜을 구현하는데 사용될 수 있는 컴퓨팅 환경의 다이어그램(800)을 예시한다. 프로토콜은 블록체인 기술을 사용하여 구현되어 블록체인 외부의 데이터에 기초하여 생성된 정확성 증명을 저장할 수 있다. 실시예에서, 증명자(802), 데이터 소스(804), 클라이언트(806), 인증 기관(808) 및 블록체인(810)은 본 개시내용의 다른 곳에서 설명된 것을 따르며 - 이들 컴포넌트는 컴퓨팅 시스템으로서 구현될 수 있다.

[0126] 실시예에서, 증명자(802)는 컴퓨터 시스템의 하나 이상의 프로세서에 의해 실행될 때 컴퓨터 시스템이 데이터 소스(804)로부터 데이터를 요청하게 하는 실행 가능 코드를 포함하는 컴퓨터 시스템이다. 데이터 소스(804)는 클라이언트(806)와 신뢰 관계를 갖는 임의의 적합한 컴퓨팅 엔티티일 수 있다. 데이터는 실제 이벤트 또는 보다 일반적으로 블록체인 상에서 사용할 수 없는 데이터와 관련될 수 있다. 요청될 수 있는 데이터의 예는 특정 날짜의 항공 비행 상태(예를 들어, 지연, 취소, 정시 도착 등)이다. 데이터 소스(804)는 요청에 응답하여, 데이터그램, 웹 페이지, JSON 포맷 등과 같은 임의의 적합한 포맷으로 제공될 수 있는 요청된 데이터를 제공할 수 있다. 실시예에서, 증명자는 데이터를 수신하고, 데이터가 수신되었다는 결정에 응답하여, 통신의 증명을 위해 제 2 요청을 데이터 소스(804)에 제출한다. 데이터 소스(804)는 데이터가 데이터 소스에 의해 증명자에게 제공되었다는 암호로 검증 가능한 보증을 제공하는 통신의 증명을 생성한다(812). 통신의 증명은 데이터 소스의 개인 키에 의해 디지털 서명될 수 있고, 디지털 서명의 진위성은 데이터 소스와 연관된 공개 키를 사용하여 검증될 수 있다. 공개 키를 간직하는 디지털 인증서는 인증 기관(808)에 의해 발행될 수 있다. 증명자(802) 및 데이터 소스(804)는 진위성, 무결성 및 비밀성의 암호로 검증 가능한 보장을 제공하는 TLS 세션과 같은 암호로 보

호된 통신 세션(814)을 통해 통신할 수 있다.

[0127] 실시예에서, 증명자(802)는 클라이언트(8__)로부터 스마트 계약의 조건을 캡슐화하는 산술 회로를 수신한다. 증명자(802)는 계산을 실행할 수 있고, 여기서 데이터의 계산은 데이터 소스(804)로부터 획득된 데이터에 적어도 부분적으로 기초하여 수행된다. 증명자(802)는 출력을 획득하고, 블록체인 네트워크(810) 상에 브로드캐스트되어 기록(816)된 회로의 올바른 실행의 증명을 생성(814)함으로써, 블록체인의 노드가 스마트 계약을 유효화할 수 있게 한다.

[0128] 오프-체인 데이터(예를 들어, 실세계 데이터)를 포함하는 스마트 계약의 실행은 당사자 사이의 신뢰의 상이한 레벨에 의존한다. 신뢰를 얻기 위해, 증명자는 클라이언트(또는 증명자가 정직한 지를 증명하는 임의의 다른 시스템)에게 증명자와 데이터를 생성하는 것에 신뢰할 수 있는 데이터 소스 사이에 특정 대화가 발생했다는 것을 입증할 수 있다. 실시예에서, 증명자는 클라이언트에게 산술 회로 및 통신 증명을 평가하는데 사용된 입력 데이터를 비롯한 레코드를 포함하는 (예를 들어, 도 7과 관련하여 설명된 바와 같이) 머클 경로를 제공한다(818). 클라이언트는 머클 경로 및 통신의 증명을 수신하고, 특정 레코드(또는 레코드의 데이터)가 통신의 증명을 통해 생성된 디지털 서명의 진위성을 검증하는 (예를 들어, 인증 기관(808)으로부터 데이터 소스의 공개 키를 간직한 디지털 인증서를 요청함으로써) 신뢰할 수 있는 데이터 소스와 연관된 공개 키를 획득함으로써 신뢰할 수 있는 데이터 소스와의 통신의 일부이었다는 것을 검증하고(820), 머클 경로로부터 해시 출력을 생성하고, 생성된 해시 출력이 수신된 통신의 증명과 일치한다는 것을 검증한다. 이러한 방식으로, 클라이언트는 스마트 계약의 입력이 신뢰할 수 있는 소스에 의해 제공되었다고 결정할 수 있다.

[0129] 실시예에서, 클라이언트는, 이를테면 본 개시내용의 다른 곳에 설명된 방식으로 및/또는 U.K. 특허 출원 No. 1719998.5에 기재된 기술을 사용하여, 정확성의 증명을 검증한다(822). 클라이언트는, 또한 블록체인에 기록(826)되고 증명자가 위에서 설명된 특정 스마트 계약의 이행에서 정직하게 행동했는지를 표시하는 클라이언트의 검토를 블록체인에 브로드캐스트하는 평판 트랜잭션을 생성(824)할 수 있다. 이러한 맥락에서는, 일부 실시예에서, 상이한 입력을 사용하여 스마트 계약을 실행하고, 그럼으로써 스마트 계약의 실행의 결과를 변경하는 부정한 엔티티와는 대조적으로, 데이터 소스에 의해 제공된 동일 데이터를 사용하여 스마트 계약을 실행하는 증명자를 정직하게 지칭할 수 있다.

[0130] 도 9는 도 4 및 도 8과 관련하여 설명된 실시예와 같은 다양한 맥락에서 이용될 수 있는 평판 트랜잭션(902)의 예시적인 다이어그램(900)이다. 실시예에서, 평판 트랜잭션은 증명자가 정직하다는 가정 하의 계약 후 실행 검증(post-contract execution verification)의 하나의 유형이다. 실시예에서, 이러한 유형의 검증은 계약의 실행에 영향을 미치지 않으며, 오히려 증명자의 평판에 기여하는 메트릭으로서 이용된다. 스마트 계약이 실행되면, 클라이언트는 잘 행동하는 증명자를 홍보함으로써 스마트 계약 시장의 성과를 높이는 인센티브로 증명자의 서비스를 검토할 기회를 갖는다. 이와 반대로, 증명자는 그들의 평판과 그들의 미래의 잠재 고객에게 직접적인 영향을 미치므로, 올바르게 행동하도록 장려되며 - 클라이언트는 그들의 평판(또는 평판의 결여)에 기초하여 증명자를 선택할 수 있다.

[0131] 실시예에서, (예를 들어, 도 4와 관련하여 설명된 바와 같은) 클라이언트는 검토자(904) 입력과의 트랜잭션을 생성한다. 검토자 입력은 클라이언트의 개인 키를 사용하여 서명된 트랜잭션 입력일 수 있다. 일반적으로 말하면, 검토자 입력은 클라이언트가 평판 트랜잭션의 검토자라는 공공연하게 검증 가능한 입증을 제공하는 임의의 정보일 수 있다. 평판 트랜잭션은 3 개의 출력을 포함한다. 검증(906) 출력은 공개 키, 서명, 및 머클 경로를 입력으로서 받는 연관된 잠금해제 스크립트를 가질 수 있다. 검증시에, 머클 경로는 레코드가 신뢰할 수 있는 엔티티(예를 들어, 도 4와 관련하여 설명된 데이터 제공자)에 의해 서명되고 전송된 대화 증명에 속한다는 것을 증명하는데 사용될 수 있다. 일반적으로 말하면, 검증(906) 출력은 기대 값 및 기대 값을 생성하는데 사용 가능한 인증 정보를 포함하는 트랜잭션 출력이며 - 기대 값은 위에서 설명한 바와 같은 공개 키를 사용하여 검증 가능한 디지털 서명의 디지털 서명된 진위성일 수 있다. 실시예에서, 인증 정보는 루트 해시 값을 생성하는데 사용 가능한 머클 경로이다. 제 2 출력은 클라이언트에 속하는 주소를 포함하는 폐기(revocation)(908) 출력일 수 있다. 실시예에서, 폐기 출력을 잠금해제하는 것은 검토가 취소되거나 철회되는 것의 상징이다. 평판 트랜잭션은 또한 검토 메타데이터(910)를 증명자의 신원(예를 들어, 증명자의 주소), 증명자가 실행한 스마트 계약, 및 긍정적인 검토의 표시에 관한 정보를 포함하는 제 3 트랜잭션 출력으로서 인코딩할 수 있다. 비트코인 기반 실시예에서, 평판 메타데이터(910)는 OP_RETURN 출력으로서 구현된다. 실시예에서, 출력은 디지털 자산의 명목적인 양(예를 들어, 더스트(dust) 값)을 갖거나 또는 (소비될 수 없는 OP_RETURN 출력의 경우에는) 전혀 값을 갖지 않을 수 있다. 실시예에서, 비트코인 기반 시스템은 다음에 따라 구현된 평판 트랜잭션을 가질

수 있다:

입력 주소	입력량	잠금 해제 스크립트	출력량
클라이언트	2d + miner fee (ClientSig)	검증 잠금해제 스크립트: OP_8 OP_ROLL (OP_CAT OP_SHA256 OP_SWAP OP_IF OP_SWAP OP_ENDIF) * 4 <h'final> OP_EQUAL ProverPubKey OP_CHECKSIG 해지 잠금해제 스크립트: ClientPubKey OP_CHECKSIG 검토 메타데이터 잠금해제 스크립트: OP_RETURN SCMReview 1 proverID contractID	d d 0

[0132]

[0133] OP_CAT은 2 개의 문자열을 연결하는데 적합한 임의의 opcode, 루틴, 커맨드 또는 함수를 지칭할 수 있다는 것을 또한 유의하여야 한다.도 10은 통신의 증명을 생성하기 위한 프로세스(1000)의 예시적인 다이어그램이다. 실시예에서, 프로세스(1000)는 하드웨어, 소프트웨어 또는 이들의 조합을 사용하여 구현된다. 프로세스를 수행하기에 적합한 시스템은 도 4와 관련하여 설명된 것과 같은 데이터 제공자를 포함한다. 일반적으로 말하면, 프로세스(1000)를 수행하는 시스템은 스마트 계약 또는 다른 계산 작업의 실행을 다른 컴퓨팅 엔티티에 위임하는 컴퓨팅 엔티티에 의해 신뢰되는 임의의 적절한 시스템일 수 있다. 이러한 맥락에서, 신뢰는 신뢰할 수 있는 엔티티로부터 발신하는 및/또는 브로드캐스팅된 데이터가 올바른 것으로 수락하고 및/또는 유효한 보안 인증서를 갖는 신뢰할 수 있는 소스로부터 게시된 데이터가 진실하고 정확하다고 수락하는 하나의 컴퓨팅 엔티티로 지칭할 수 있다.

[0134] 실시예에서, 프로세스(1000)를 수행하는 시스템은 시스템의 하나 이상의 프로세서에 의해 실행될 때, 시스템으로 하여금 증명자와 같은 컴퓨팅 엔티티와 암호로 보호된 통신 세션을 수립(1002)하게 하는 실행 가능 코드를 포함한다. 클라이언트는 스마트 계약의 실행에 이용되는 데이터에 대해 신뢰할 수 있는 소스의 리스트를 가질 수 있으며 - 리스트는 (예를 들어, 블록체인 상의 스마트 계약 또는 다른 곳의 일부로서 또는 그와 연관되어) 게시될 수 있다. 암호로 보호된 통신 세션의 예는 TLS 세션이다.

[0135] 시스템은 시스템과 상대방 사이의 통신 인증을 시작하는 표시를 검출할 수 있다(1004). 표시는 프로토콜의 일부로서 암시적으로 정의될 수 있으며 - 예를 들어, 시스템은 4-방향 핸드셰이크 프로토콜의 완료시에 통신 인증을 시작할 수 있다. 실시예에서, 표시는 레코드 프로토콜에 따라 메시지로서 시스템에 의해 수신되며, 메시지는 통신 인증을 시작한다는 것을 표시한다. 통신 인증은 또한 2 이상의 컴퓨팅 엔티티 사이의 통신 및/또는 대화를 공증하는 것으로 지칭될 수 있다는 것을 유의하여야 한다. 통신 인증을 시작한다는 표시의 일부로서, 시스템은 시스템이 통신 인증을 시작한 때의 타임스탬프를 기록할 수 있다.

[0136] 통신 인증을 시작한다는 표시에 후속하여, 시스템은 레코드 프로토콜의 일부로서 데이터를 수신 및/또는 송신(1006)할 수 있고, 통신(또는 그에 포함된 데이터)은 데이터 구조로서 단기 메모리와 같은 임의의 적절한 포맷으로 기록될 수 있다. 데이터의 수신 및 송신과 관련하여 예시된 적층된 박스는 다수의 레코드가 수신 및 송신될 수 있다는 것을 표시한다는 것을 주목하여야 하지만, 그렇게 할 필요가 없고 - 실시예에서 단일 레코드가 수신 및/또는 송신된다.

[0137] 실시예에서, 시스템은 통신 인증을 시작한다는 표시를 수신한 후에, 상대방과의 통신 인증을 종료한다는 표시를 수신할 수 있다(1008). 실시예에서, 시스템은 통신 인증의 종료하는 요청이 검출된 시간의 타임스탬프를 기록한다. 일부 경우에, 예를 들어, 종료는 암호로 보호된 통신 세션의 종료에 의해 암시적으로 검출된다.

[0138] 실시예에서, 시스템은 인증 기간(즉, 시스템이 인증 시작을 검출한 때와 인증 종료를 검출한 때 사이의 시간) 동안 송신된 및/또는 수신된 레코드(또는 그 일부)에 기초하여 통신의 증명을 생성한다(1010). 시스템은 도 7과 관련하여 설명된 방식으로 머클 트리를 생성함으로써 통신의 증명을 생성할 수 있다. 머클 트리의 루트는 통신의 증명일 수 있다. 실시예에서, 통신의 증명은 인증된 통신이 발생한 시간 간격을 인코딩하는 머클 트리의 루트 및 W_time에 기초하여 결정된다. 통신의 증명은 개인 키를 사용하여 시스템에 의해 디지털 서명될 수 있으며, 여기서 대응하는 공개 키는 디지털 인증서를 발행하는 인증 기관을 통해 액세스 가능하다. 실시예에서, 시스템은 도 3 내지 도 5 및 도 8과 관련하여 설명된 데이터 제공자이다. 통신의 증명을 생성할 때, 시스템은 통신의 증명을 세션의 상대방(예를 들어, 증명자) 및/또는 다른 엔티티(예를 들어, 세션 동안 증명자에 의해 명시된 클라이언트)에게 송신할 수 있다(1014).

[0139] 도 11은 블록체인 네트워크에 게시된 스마트 계약과 같은 인증된 데이터를 사용하여 프로그램 또는 스크립트를

실행하기 위한 프로세스(1100)의 예시적인 다이어그램이다. 실시예에서, 외부 데이터(예를 들어, 실세계 이벤트에 관한 데이터)는 인증되는 반면, 블록체인으로부터 획득 가능한 데이터는 인증되지 않는다. 이러한 맥락과 관련하여 언급된 데이터는 스마트 계약에 사용되는 출력을 생성하기 위해 산술 회로의 평가의 일부로서 사용되는 입력 데이터를 지칭할 수 있다. 실시예에서, 프로세스(1100)는 하드웨어, 소프트웨어 또는 이들의 조합을 사용하여 구현된다. 프로세스를 수행하기에 적합한 시스템은 도 2 내지 도4 및 도 8과 관련하여 설명된 것과 같은 증명자를 포함한다.

[0140] 실시예에서, 시스템은 클라이언트를 대신하여 스마트 계약을 실행하기 위해 외부 데이터가 획득되어야 한다고 결정한다. 시스템은 데이터가 획득될 수 있는 컴퓨팅 엔티티를 식별하며(1102), 여기서 컴퓨팅 엔티티는 클라이언트에 의해 신뢰되는 엔티티이다. 클라이언트는 신뢰할 수 있는 엔티티의 리스트를, 이를테면 스마트 계약에서 또는 스마트 계약과 관련하여 또는 블록체인의 다른 곳에서 게시할 수 있고, 시스템은 외부 데이터를 획득할 수 있는 적합한 신뢰할 수 있는 엔티티를 선택한다. 실시예에서, 시스템은 데이터 소스와 암호로 보호된 통신 세션을 수립한다(1104). 암호로 보호된 통신 세션의 예는 TLS 또는 SSL 세션이다.

[0141] 실시예에서, 시스템은 소스로부터 외부 데이터를 요청하고 암호로 보호된 통신 세션을 통해 데이터 소스로부터 데이터를 수신한다(1106). 외부 데이터는, 이것으로 제한되는 것은 아니지만 자산 및 금융 애플리케이션에 대한 시장 가격 피드(예를 들어, 금리)에 액세스해야 하는 유가 증권(예를 들어, 이자율, 파생 상품, 채권)에 사용하기 위한 스마트 계약; 외부 데이터(예를 들어, 지연에 대비하여 당사자가 보장하는 항공편 정보; 농작물 보험에 대한 가격 지수 대신에 날씨의 데이터 피드를 사용하는 금융 파생 상품 계약에 대한 날씨 데이터)에 액세스해야 하는 피어 투 피어 보험 스마트 계약; 선적에 관한 GPS 데이터가 필요한 무역 스마트 계약; 난수 생성기에 액세스해야 하는 도박 계약; 등을 비롯한 스마트 계약을 둘러싼 다양한 상황에서 사용될 수 있다.

[0142] 시스템이 외부 데이터를 수신한 후에, 시스템은 한 세트의 통신이 암호로 보호된 통신 세션의 일부로서 발생했다는 제 1 입증을 제공하는 표시를 데이터 소스에 제공할 수 있으며, 여기서 통신 세트는 시스템이 신뢰할 수 있는 데이터 소스로부터 외부 데이터를 수신하는 것을 포함한다. 시스템은 제 1 입증을 수신할 수 있다(1108). 제 1 입증은 머클 트리의 해시 루트인 통신의 증명일 수 있다. 제 1 입증은 디지털 서명될 수 있고, 디지털 서명의 진위성은 신뢰할 수 있는 데이터 소스와 연관된 공개 키에 의해 검증될 수 있다.

[0143] 실시예에서, 시스템은 생성하고(1110), 데이터를 산술 회로($\mathcal{C}$)에 대한 입력(x)으로서 사용하며, 여기서 회로는 하나 이상의 출력(y)을 생성할 이차 프로그램을 생성하는데 사용된다. 실시예에서, 시스템(예를 들어, 증명자)은 입력 와이어에 할당된 값이 x의 값이 되도록 회로 와이어에 값을 할당하는 $\{\mathcal{C}, x, y\}$에 대한 유효 트랜스크립트를 출력으로서 획득한다. 시스템은 또한 산술 회로로의 입력으로서 사용된 데이터가 신뢰할 수 있는 소스로부터 획득된 데이터라는 입증을 생성할 수 있다. 입증은 도 6 및 도 7과 관련하여 본 개시내용의 다른 곳에서 설명된 바와 같은 머클 경로일 수 있다. 시스템은 스마트 계약의 올바른 실행의 증명을 블록체인에 브로드캐스트할 수 있다(1112). 시스템은 제 2 입증(예를 들어, 머클 경로), 제 1 입증(즉, 통신의 증명과 같은 데이터 소스로부터 수신된 입증), 신뢰할 수 있는 엔티티와 연관된 공개 키, 또는 이들의 일부 조합을 송신할 수 있다.

[0144] 본 개시내용의 맥락에서, 프리-이미지 저항성 함수는, 현재 값에 대해 계산하기 어렵지만 현재의 값으로부터 이전의 값을 결정하기 위해 계산적으로 사소하지 않을 수 있는 함수를 지칭하는 일방향 함수를 포함한다는 것을 유의하여야 한다. 일부 경우에, 일방향 멤버십 함수가 수학적으로 일방향으로 증명되지 않고/증명 가능하지 않지만, 그럼에도 불구하고 함수가 프리-이미지 저항성을 갖게 하는 계산 복잡도 속성을 갖는다. 일방향 함수 - 또한 "효과적 일방향 함수(effectively one-way function)"라고도 함 - 는, 이것으로 제한되는 것은 아니지만 메시지 인증 코드(예를 들어, 해시 기반 메시지 인증 코드(hash based message authentication code)(HMAC))와 같은 암호화 해시 함수, (예를 들어, 패스워드가 평문 및 암호화 키에 적어도 부분적으로 기초하는) PBKDF2 및 bcrypt와 같은 키 파생 함수, 및 그들의 범위(가능한 출력)보다 큰 도메인(가능한 입력 세트)를 가질 수 있지만 반드시 필요한 것은 아닐 수 있는 다른 보안 랜덤화 함수를 포함한다. 다양한 실시예에 대한 다른 적절한 함수는, 이것으로 제한되는 것은 아니지만 적어도 평문 및 암호 키를 입력으로서 수용하고 입력을 랜덤으로 생성할 확률이 명시된 임계치 미만이 되도록 프리-이미지 저항, (입력(x1)이 주어지면, f(x1)=f(x2)이 되도록 상이한 입력(x2)을 랜덤으로 생성할 확률이 명시된 임계치 미만인) 제 2 프리-이미지 저항성 및/또는 (예를 들어, 결과적으로 동일한 출력이 되는 2 개의 상이한 입력의 확률이 명시된 임계치 미만인) 충돌 저항성의 특성을 갖는 함

수를 포함한다.

[0145] 도 12는 본 개시내용의 적어도 하나의 실시예를 실시하는데 사용될 수 있는 컴퓨팅 디바이스(1200)의 예시적인 단순화된 블록도이다. 다양한 실시예에서, 컴퓨팅 디바이스(1200)는 위에서 예시되고 설명된 시스템 중 임의의 것을 구현하는데 사용될 수 있다. 예를 들어, 컴퓨팅 디바이스(1200)는, 데이터 서버, 웹 서버, 휴대용 컴퓨팅 디바이스, 개인용 컴퓨터, 또는 임의의 전자 컴퓨팅 디바이스로서 사용하기 위해 구성될 수 있다. 도 12에 도시된 바와 같이, 컴퓨팅 디바이스(1200)는, 실시예에서, 버스 서브시스템(1204)을 통해 다수의 주변기기 서브시스템과 통신하고 다수의 주변기기 서브시스템에 동작 가능하게 결합된 하나 이상의 프로세서(1202)를 포함할 수 있다. 일부 실시예에서, 이러한 주변기기 서브시스템은 메모리 서브시스템(1208) 및 파일/디스크 저장 서브시스템(1210), 하나 이상의 유저 인터페이스 입력 디바이스(1212), 하나 이상의 유저 인터페이스 출력 디바이스(1214) 및 네트워크 인터페이스 서브시스템(1216)을 포함하는 저장 서브시스템(1206)을 포함한다. 이러한 저장 서브시스템(1206)은 정보의 일시 또는 장기 저장을 위해 사용될 수 있다.

[0146] 일부 실시예에서, 버스 서브시스템(1204)은 컴퓨팅 디바이스(1200)의 다양한 컴포넌트 및 서브시스템이 의도대로 서로 통신할 수 있게 하기 위한 메커니즘을 제공한다. 버스 서브시스템(1204)이 단일의 버스로서 개략적으로 도시되지만, 버스 서브시스템의 대안적인 실시예는 다수의 버스를 이용한다. 일부 실시예에서, 네트워크 인터페이스 서브시스템(1216)은 다른 컴퓨팅 디바이스 및 네트워크와의 인터페이스를 제공한다. 네트워크 인터페이스 서브시스템(1216)은, 일부 실시예에서, 다른 시스템으로부터 데이터를 수신하고 컴퓨팅 디바이스(1200)로부터 다른 시스템으로 데이터를 송신하기 위한 인터페이스로서 동작한다. 일부 실시예에서, 버스 서브시스템(1204)은 세부 사항, 검색 용어 등과 같은 데이터를 통신하는데 이용된다.

[0147] 일부 실시예에서, 사용자 인터페이스 입력 디바이스(1212)는 키보드와 같은 하나 이상의 사용자 입력 디바이스; 통합 마우스, 트랙볼, 터치패드, 또는 그래픽 태블릿과 같은 포인팅 디바이스; 스캐너; 바코드 스캐너; 디스플레이에 통합되는 터치스크린; 음성 인식 시스템과 같은 오디오 입력 디바이스; 및 다른 유형의 입력 디바이스를 포함한다. 일반적으로, 용어 "입력 디바이스"의 사용은 컴퓨팅 디바이스(1200)에 정보를 입력하기 위한 모든 가능한 유형의 디바이스 및 메커니즘을 포함하는 것으로 의도된다. 일부 실시예에서, 하나 이상의 사용자 인터페이스 출력 디바이스(1214)는 디스플레이 서브시스템, 프린터, 또는 오디오 출력 디바이스와 같은 비시각적 디스플레이 등을 포함한다. 일부 실시예에서, 디스플레이 서브시스템은 음극선관(cathode ray tube)(CRT), 액정 디스플레이(liquid crystal display)(LCD), 발광 다이오드(light emitting diode)(LED) 디스플레이와 같은 플랫 패널 디바이스, 또는 투영 또는 다른 디스플레이 디바이스를 포함한다. 일반적으로, 용어 "출력 디바이스"의 사용은 컴퓨팅 디바이스(1200)로부터 정보를 출력하기 위한 모든 가능한 유형의 디바이스 및 메커니즘을 포함하는 것으로 의도된다. 하나 이상의 유저 인터페이스 출력 디바이스(1214)는, 예를 들어, 설명되는 프로세스 및 그 변형예를 수행하는 애플리케이션과의 사용자 상호작용을, 이러한 상호작용이 적절할 수 있을 때, 용이하게 하기 위한 사용자 인터페이스를 제시하는데 사용될 수 있다.

[0148] 일부 실시예에서, 저장 서브시스템(1206)은 본 개시내용의 적어도 하나의 실시예의 기능성을 제공하는 기본 프로그래밍 및 데이터 구성을 저장하기 위한 컴퓨터 판독 가능 저장 매체를 제공한다. 애플리케이션(프로그램, 코드 모듈, 명령어)은 일부 실시예에서 하나 이상의 프로세서에 의해 실행될 때, 본 개시내용의 하나 이상의 실시예의 기능성을 제공하고, 실시예에서, 저장 서브시스템(1206)에 저장된다. 이러한 애플리케이션 모듈 또는 명령어는 하나 이상의 프로세서(1202)에 의해 실행될 수 있다. 다양한 실시예에서, 저장 서브시스템(1206)은 또한 본 개시내용에 따라 사용되는 데이터를 저장하기 위한 저장소를 제공한다. 일부 실시예에서, 저장 서브시스템(1206)은 메모리 서브시스템(1208) 및 파일/디스크 저장 서브시스템(1210)을 포함한다.

[0149] 실시예에서, 메모리 서브시스템(1208)은 프로그램 실행 동안 명령어 및 데이터의 저장을 위한 메인 랜덤 액세스 메모리(random access memory)(RAM)(1218) 및/또는 고정된 명령어가 저장될 수 있는 판독 전용 메모리(read only memory)(ROM)(1220)와 같은 다수의 메모리를 포함한다. 일부 실시예에서, 파일/디스크 저장 서브시스템(1210)은 프로그램 및 데이터 파일의 비일시적인 영구적(비휘발성) 저장소를 제공하고, 하드 디스크 드라이브, 연관된 착탈식 매체를 수반한 플로피 디스크 드라이브, 콤팩트 디스크 판독 전용 메모리(Compact Disk Read Only Memory)(CD-ROM) 드라이브, 광학 드라이브, 착탈식 매체 카트리지, 또는 다른 유사한 저장 매체를 포함할 수 있다.

[0150] 일부 실시예에서, 컴퓨팅 디바이스(1200)는 적어도 하나의 로컬 클록(1224)을 포함한다. 로컬 클록(1224)은, 일부 실시예에서, 특정한 시작 날짜로부터 발생된 틱(tick)의 수를 나타내는 카운터이며, 일부 실시예에서, 컴퓨팅 디바이스(1200) 내부에서 일체로 위치된다. 다양한 실시예에서, 로컬 클록(1224)은 특정 클록 펄스에서

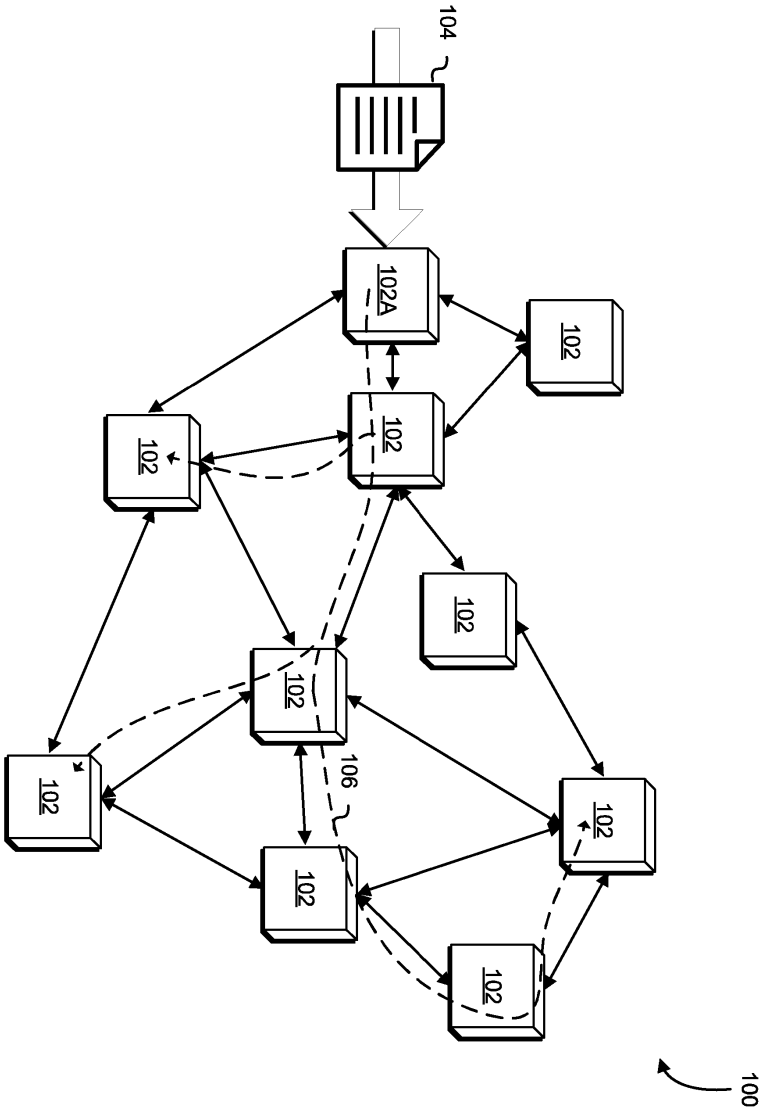
컴퓨팅 디바이스(1200)의 프로세서 및 그 내부에 포함된 서브시스템에서 데이터 전송을 동기화하는데 사용되고, 컴퓨팅 디바이스(1200)와 데이터 센터 내의 다른 시스템 사이의 동기 동작을 조정하는데 사용될 수 있다. 다른 실시예에서, 로컬 클록은 프로그램 가능한 인터벌 타이머(interval timer)이다.

[0151] 컴퓨팅 디바이스(1200)는 휴대용 컴퓨터 디바이스, 태블릿 컴퓨터, 워크스테이션, 또는 아래에서 설명되는 임의의 다른 디바이스를 비롯한, 다양한 유형 중 임의의 유형의 것일 수 있다. 또한, 컴퓨팅 디바이스(1200)는, 일부 실시예에서, 하나 이상의 포트(예를 들어, USB, 헤드폰 잭, 라이트닝 커넥터(Lightning connector) 등)를 통해 컴퓨팅 디바이스(1200)에 연결될 수 있는 다른 디바이스를 포함할 수 있다. 실시예에서, 이러한 디바이스는 광섬유 커넥터를 수용하는 포트를 포함한다. 따라서, 일부 실시예에서, 이러한 디바이스는 광학 신호를 전기 신호로 변환하고, 전기 신호는 디바이스를 처리를 위해 컴퓨팅 디바이스(1200)에 연결하는 포트를 통해 송신된다. 컴퓨터 및 네트워크의 끊임없이 변화하는 성질로 인해, 도 12에 도시된 컴퓨팅 디바이스(1200)에 관한 설명은 디바이스의 바람직한 실시예를 예시하는 목적에 특정한 예로서만 의도된다. 도 12에 예시된 시스템보다 더 많은 또는 더 적은 컴포넌트를 갖는 많은 다른 구성이 가능하다.

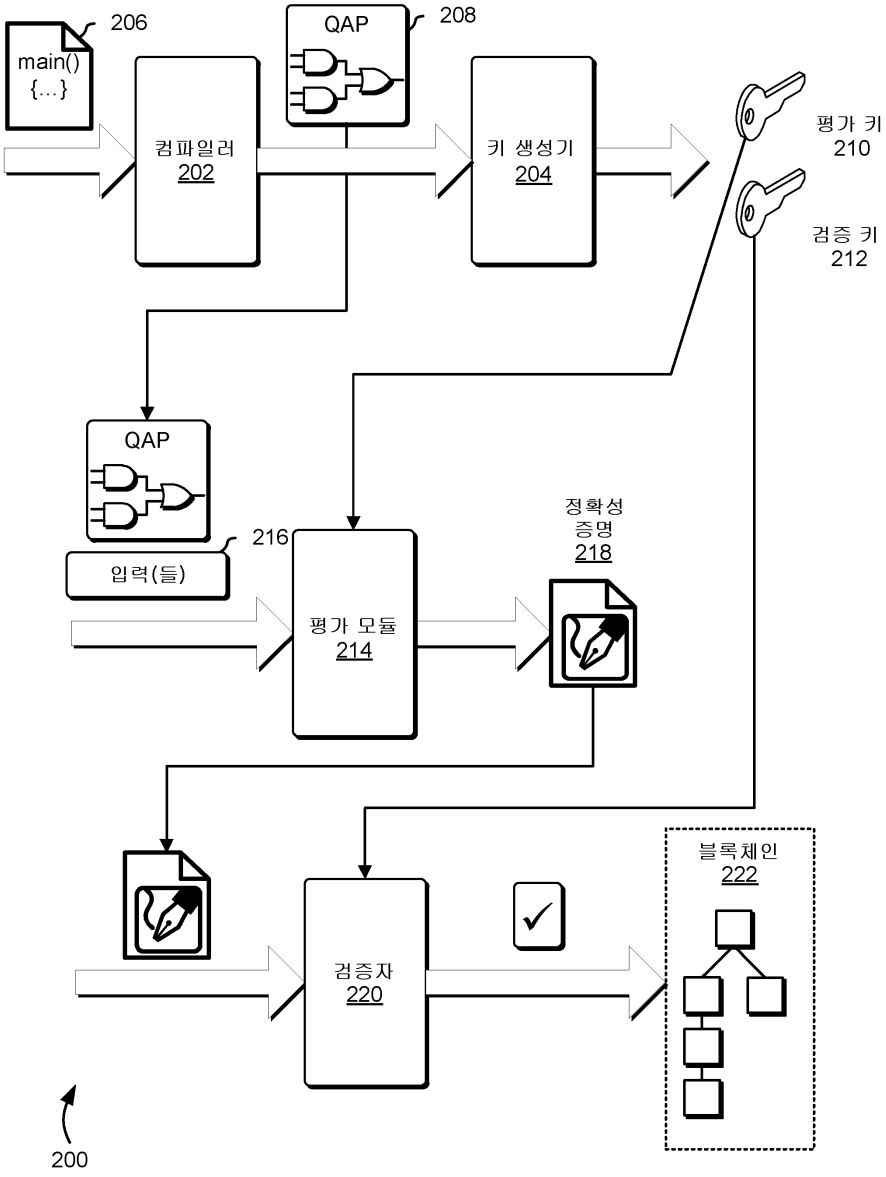
[0152] 위에서 언급한 실시예는 본 발명을 제한하기 보다는 예시하는 것이며, 관련 기술분야의 통상의 기술자는 첨부된 청구범위에 의해 정의된 바와 같은 본 발명의 범위를 벗어나지 않으면서 많은 대안적인 실시예를 설계할 수 있을 것이다는 점을 유의하여야 한다. 청구범위에서, 괄호 안의 임의의 참조 부호는 청구범위를 제한하는 것으로 해석되지 않아야 한다. "포함하는", "포함한다" 등의 단어는 임의의 청구항 또는 명세서 전체에서 열거되는 것 이외의 요소 또는 단계의 존재를 배제하지 않는다. 본 명세서에서, "포함한다"는 "구비하거나 또는 구성된다"를 의미하고 "포함하는"은 "구비하거나 또는 구성되는"을 의미한다. 요소의 단수 참조는 이러한 요소의 복수의 참조를 배제하지는 않으며, 그 반대도 마찬가지이다. 본 발명은 몇몇 별개의 요소를 포함하는 하드웨어에 의해 그리고 적절히 프로그램된 컴퓨터에 의해 구현될 수 있다. 몇몇 수단을 열거하는 장치 청구항에서, 이러한 수단 중 몇몇은 하드웨어의 하나의 아이템 및 동일한 아이템에 의해 구현될 수 있다. 특정 조치가 상이한 종속 청구항에서 인용된다는 단순한 사실은 이들 조치의 조합이 유리하게 사용될 수 없다는 것을 나타내지는 않는다.

도면

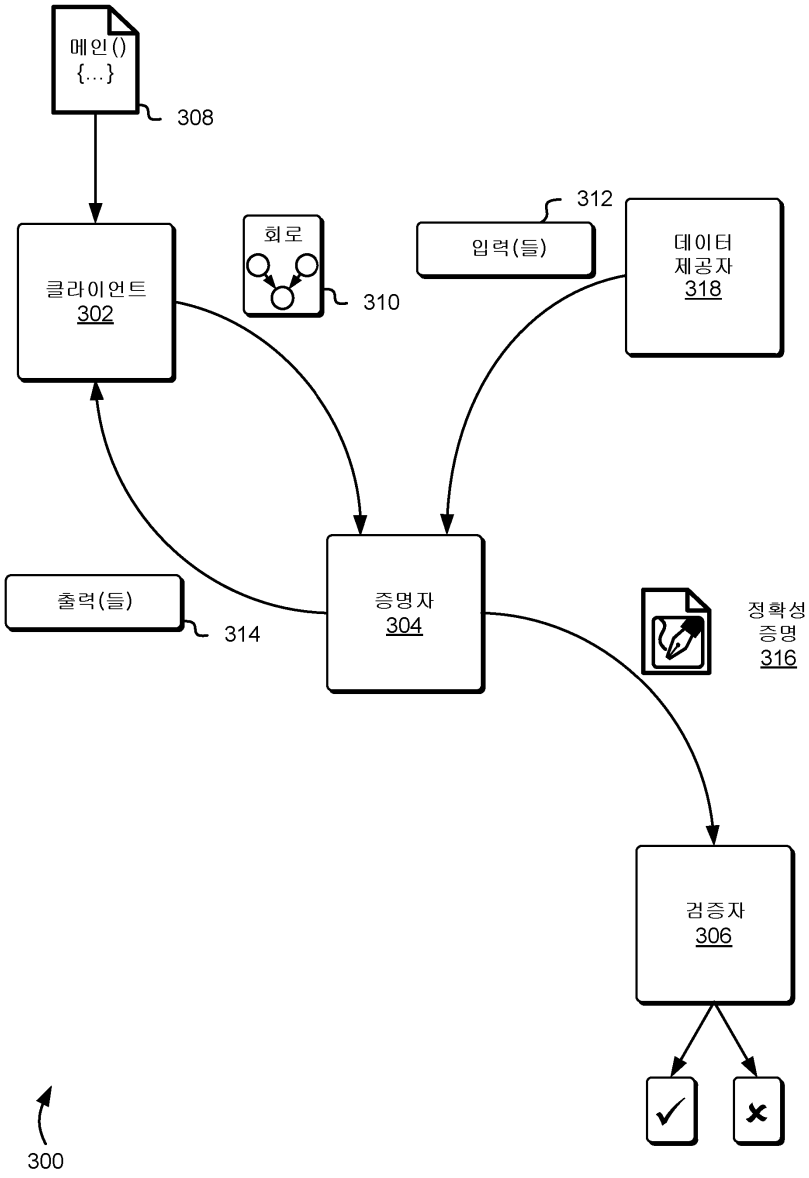
도면1



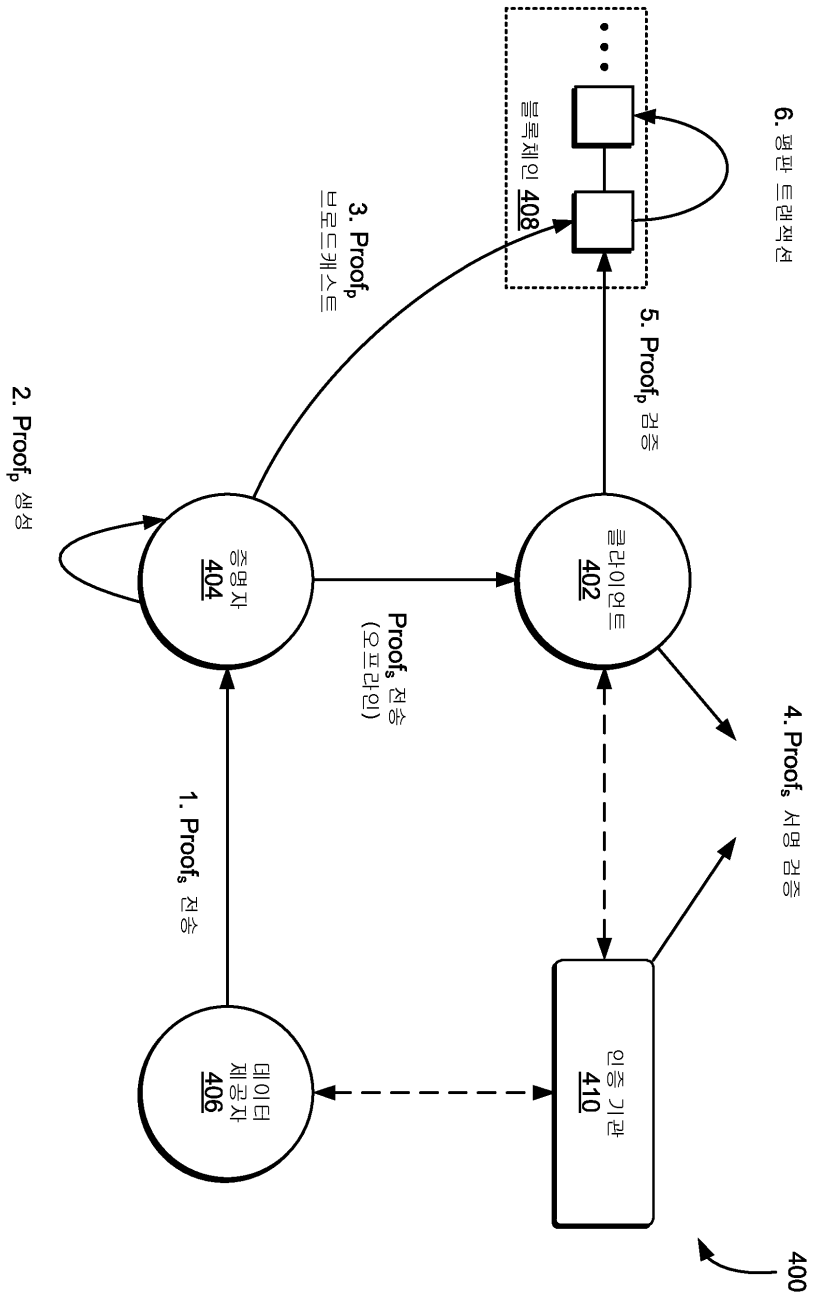
도면2


206
main()
{...}
컴파일러
202
QAP
208
키 생성기
204
평가 키
210
검증 키
212
QAP
입력(들)
216
평가 모듈
214
정확성
증명
218
블록체인
222
검증자
220
200

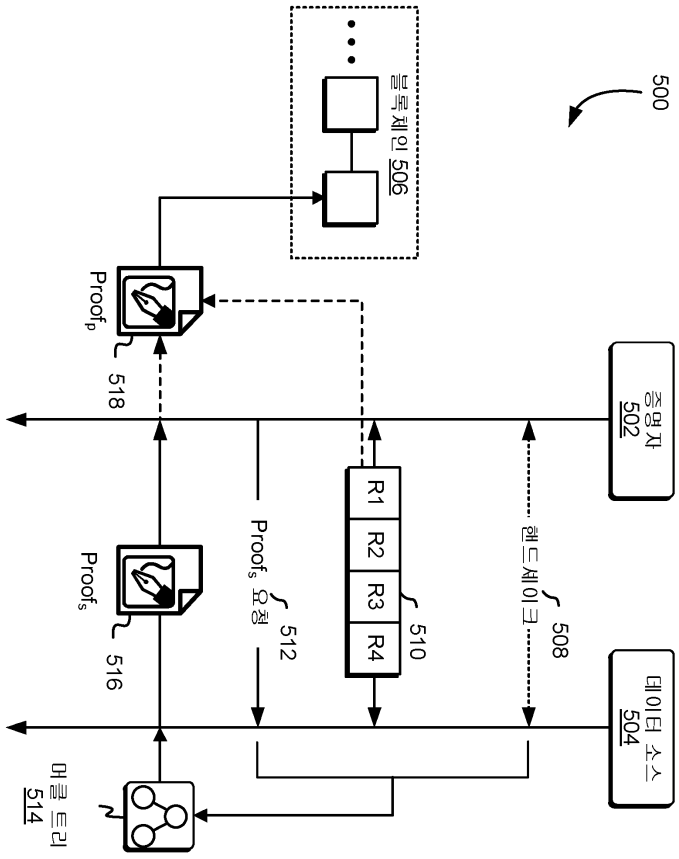
도면3


메인()
{...}
308
클라이언트
302
회로
310
312
입력(들)
데이터
제공자
318
출력(들)
314
증명자
304
정확성
증명
316
검증자
306
300

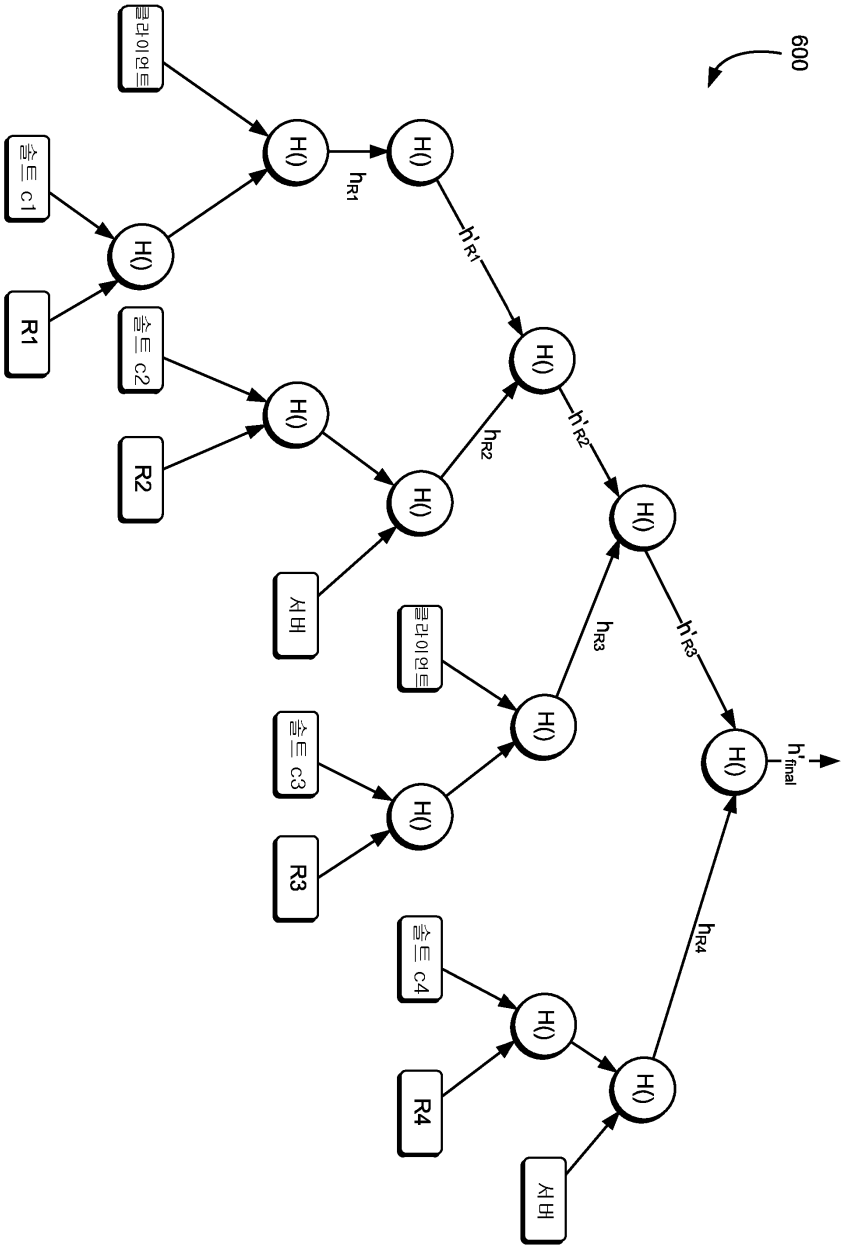
도면4


블록체인 408
3. Proof_p 브로드캐스트
2. Proof_p 생성
증명자 404
클라이언트 402
5. Proof_p 검증
6. 평판 트랜잭션
Proof_s 전송 (오프라인)
1. Proof_s 전송
4. Proof_s 서명 검증
데이터 제공자 406
인증 기관 410
400

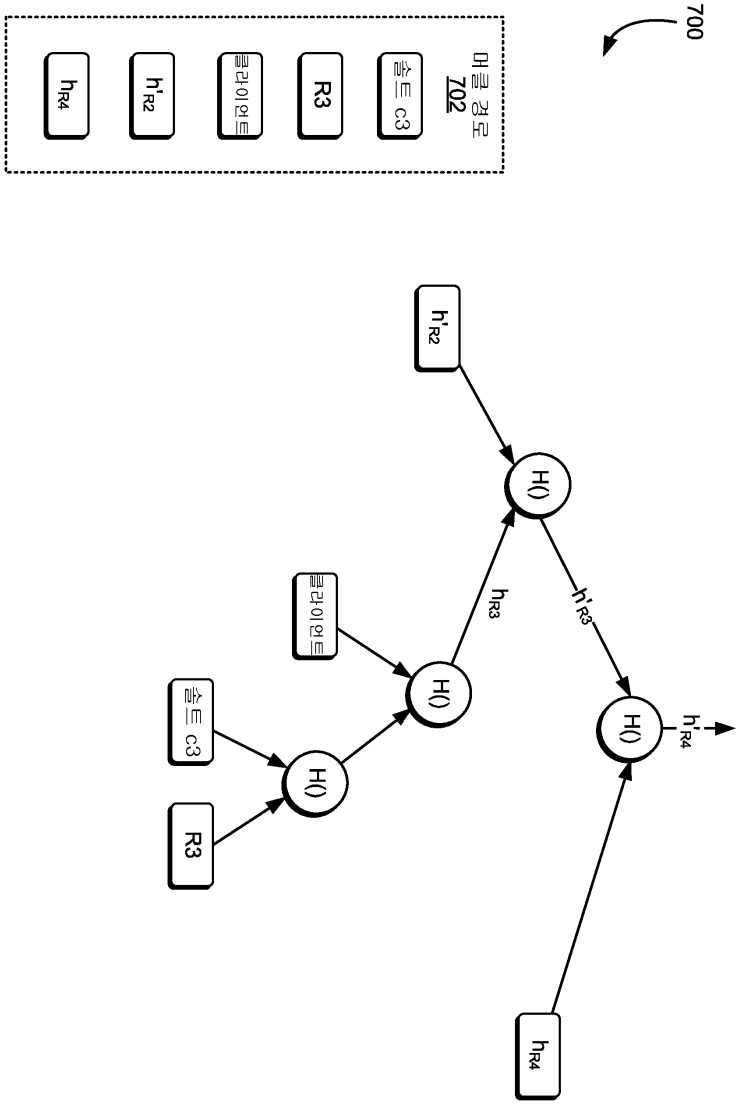
도면5


500
블록체인 506
증명자 502
데이터 소스 504
머클 트리 514
Proof$_P$ 518
Proof$_S$ 516
Proof$_S$ 요청 512
R1
R2
R3
R4
510
핸드셰이크 508

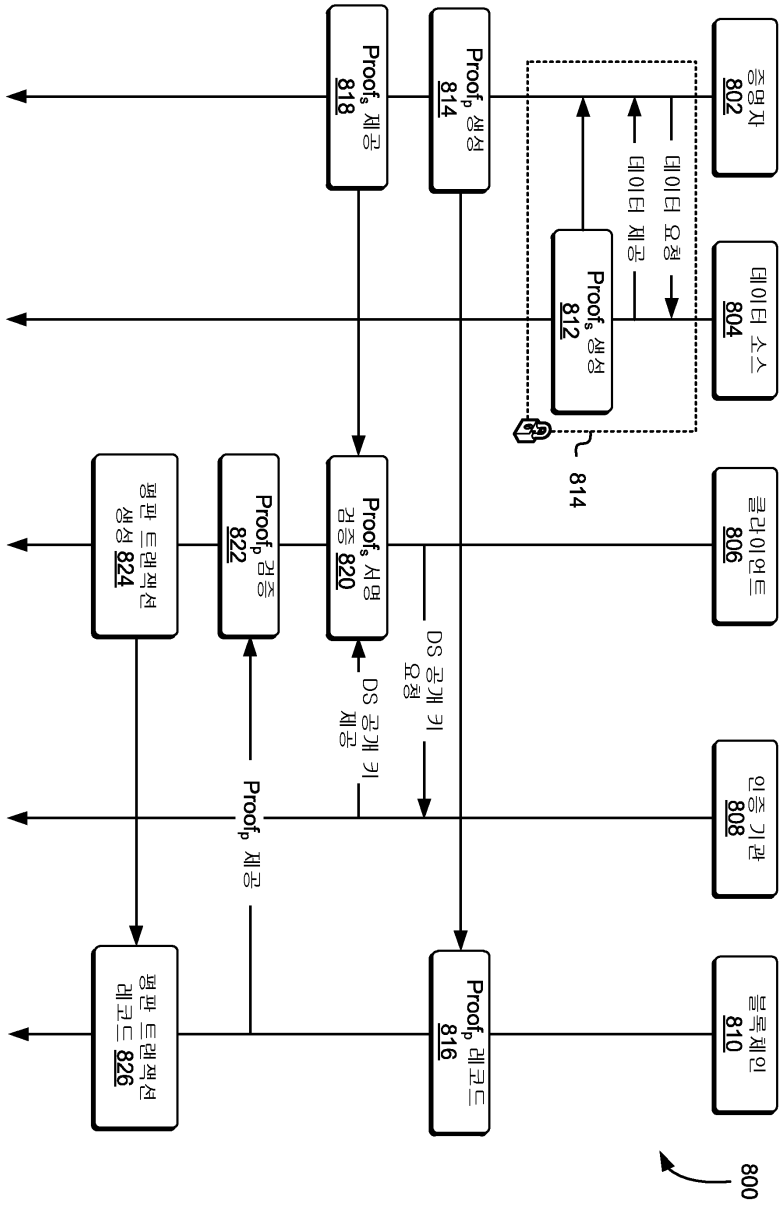
도면6


600
클라이언트
솔트 c1
R1
H()
h_{R1}
h'_{R1}
솔트 c2
R2
h_{R2}
h'_{R2}
서버
클라이언트
h_{R3}
h'_{R3}
솔트 c3
R3
h'_{final}
솔트 c4
R4
h_{R4}
서버

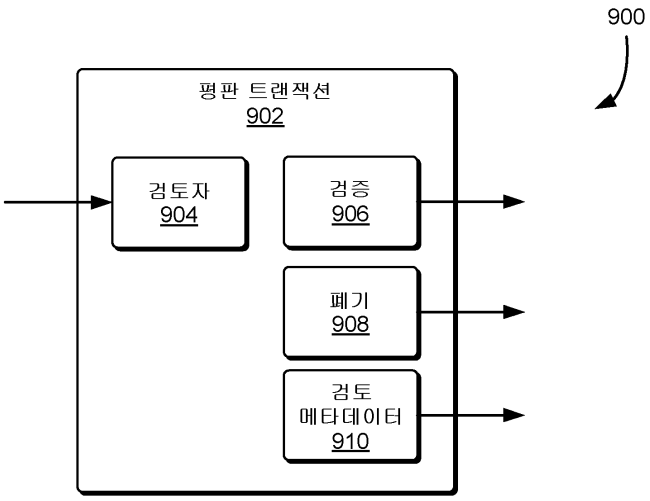
도면7



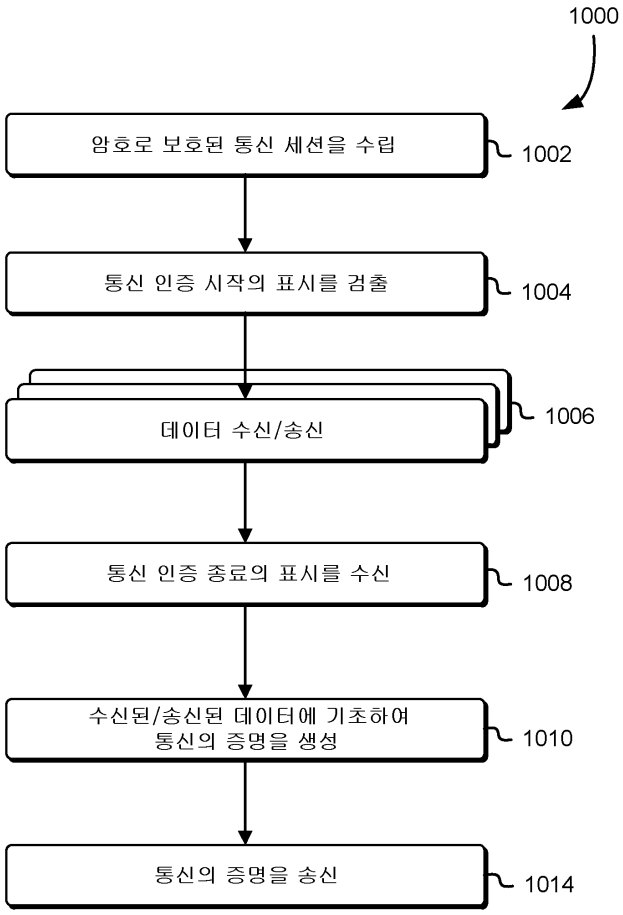
도면8


증명자 802
데이터 소스 804
클라이언트 806
인증기관 808
블록체인 810
Proofs 생성 812
814
Proofp 생성 814
Proofs 제공 818
데이터 요청
데이터 제공
Proofs 서명 검증 820
Proofp 검증 822
평판 트랜잭션 생성 824
평판 트랜잭션 레코드 826
Proofp 레코드 816
DS 공개 키 요청
DS 공개 키 제공
Proofp 제공
800

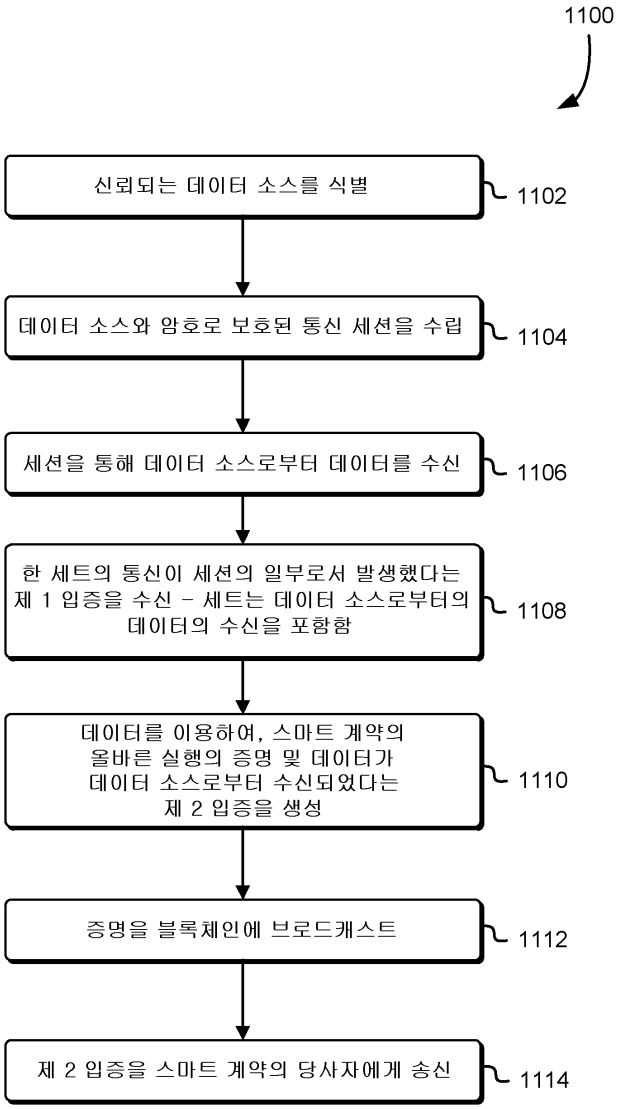
도면9


900
평판 트랜잭션
902
검토자
904
검증
906
폐기
908
검토
메타데이터
910

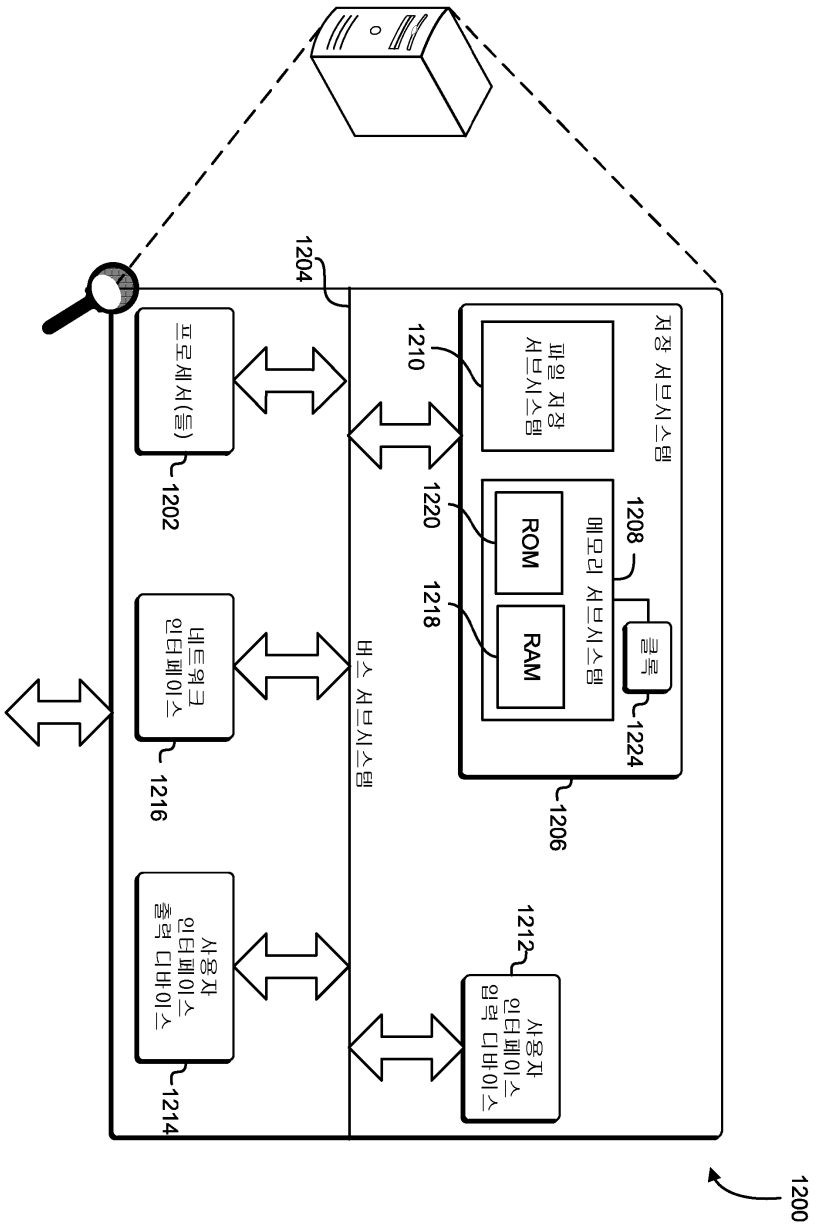
도면10


1000
암호로 보호된 통신 세션을 수립
1002
통신 인증 시작의 표시를 검출
1004
데이터 수신/송신
1006
통신 인증 종료의 표시를 수신
1008
수신된/송신된 데이터에 기초하여
통신의 증명을 생성
1010
통신의 증명을 송신
1014

도면11


1100
신뢰되는 데이터 소스를 식별
1102
데이터 소스와 암호로 보호된 통신 세션을 수립
1104
세션을 통해 데이터 소스로부터 데이터를 수신
1106
한 세트의 통신이 세션의 일부로서 발생했다는 제 1 입증을 수신 - 세트는 데이터 소스로부터의 데이터의 수신을 포함함
1108
데이터를 이용하여, 스마트 계약의 올바른 실행의 증명 및 데이터가 데이터 소스로부터 수신되었다는 제 2 입증을 생성
1110
증명을 블록체인에 브로드캐스트
1112
제 2 입증을 스마트 계약의 당사자에게 송신
1114

도면12


1200
1204
프로세서(들)
1202
네트워크 인터페이스
1216
사용자 인터페이스 출력 디바이스
1214
버스 서브시스템
저장 서브시스템
1210
파일 저장 서브시스템
1220
메모리 서브시스템
ROM
RAM
1218
1208
클록
1224
1206
1212
사용자 인터페이스 입력 디바이스