(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2022-0118405
(43) 공개일자 2022년08월25일

(51) 국제특허분류(Int. Cl.)
H04L 9/32 (2006.01) *H04L 9/00* (2022.01)
(52) CPC특허분류
H04L 9/321 (2013.01)
H04L 9/3247 (2013.01)
(21) 출원번호 10-2022-7017716
(22) 출원일자(국제) 2020년10월05일
심사청구일자 없음
(85) 번역문제출일자 2022년05월25일
(86) 국제출원번호 PCT/IB2020/059319
(87) 국제공개번호 WO 2021/084347
국제공개일자 2021년05월06일
(30) 우선권주장
1915841.9 2019년10월31일 영국(GB)

(71) 출원인
엔체인 라이센싱 아게
스위스 주크 6300 그라펜아우베크 6
(72) 발명자
맥케이, 알렉산더
영국 런던 더블유1더블유 8에이피 마켓 플레이스 30 엔체인 홀딩스 리미티드 내
타르탄, 클로이
영국 런던 더블유1더블유 8에이피 마켓 플레이스 30 엔체인 홀딩스 리미티드 내
(뒷면에 계속)
(74) 대리인
특허법인 광장리앤고

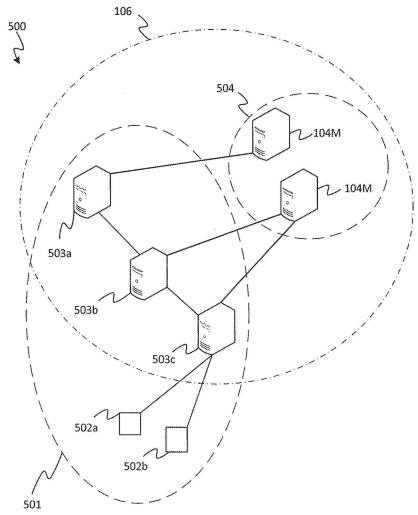
전체 청구항 수 : 총 29 항

(54) 발명의 명칭 **블록체인 트랜잭션들을 이용한 통신 프로토콜**

(57) 요 약

제1 네트워크에 참여하도록 요청자에게 허가를 부여하기 위한 컴퓨터-구현 방법이 개시된다. 제1 네트워크는 브리징 노드들의 세트 및 브리징 노드들의 세트 중 하나 이상에 의해 제어 가능한 디바이스들의 세트를 포함한다. 각각의 브리징 노드는 또한 블록체인 네트워크의 개개의 노드이다. 방법은 등록 기관에 의해 수행되고 제1 블록체인 트랜잭션을 생성하는 단계를 포함한다. 제1 블록체인 트랜잭션은 등록 기관의 제1 공개 키에 링크된 서명을 포함하는 입력을 포함한다. 제1 블록체인 트랜잭션은 또한 제1 인증서를 포함하는 제1 출력을 포함하고, 제1 인증서는 요청자에게 할당된 식별자를 포함한다. 방법은 블록체인에의 포함을 위해 블록체인 네트워크로 제1 블록체인 트랜잭션을 송신하는 단계를 더 포함한다.

대 표 도 - 도5



(52) CPC특허분류
H04L 9/3268 (2013.01)
H04L 9/50 (2022.05)
(72) 발명자
와하브, 자드
영국 런던 더블유1더블유 8에이피 마켓 플레이스 30 엔체인 홀딩스 리미티드 내
세르귀에바, 안토아네타
영국 런던 더블유1더블유 8에이피 마켓 플레이스 30 엔체인 홀딩스 리미티드 내
라이트, 크레이그 스티븐
영국 런던 더블유1더블유 8에이피 마켓 플레이스 30 엔체인 홀딩스 리미티드 내

명 세 서

청구범위

청구항 1

제1 네트워크에 참여하도록 요청자에게 허가를 부여하기 위한 컴퓨터-구현 방법으로서,

상기 제1 네트워크는 브리징(bridging) 노드들의 세트 및 상기 브리징 노드들의 세트 중 하나 이상에 의해 제어 가능한 디바이스들의 세트를 포함하고, 각각의 브리징 노드는 또한 블록체인 네트워크의 개개의 노드이고;

상기 방법은, 등록 기관(registration authority)에 의해 수행되고,

제1 블록체인 트랜잭션을 생성하는 단계 - 상기 제1 블록체인 트랜잭션은 상기 등록 기관의 제1 공개 키에 링크된 서명을 포함하는 입력, 및 제1 인증서를 포함하는 제1 출력을 포함하고, 상기 제1 인증서는 상기 요청자에게 할당된 식별자를 포함함 - ; 및

상기 블록체인에의 포함을 위해 상기 블록체인 네트워크로 상기 제1 블록체인 트랜잭션을 송신하는 단계를 포함하는,

컴퓨터-구현 방법.

청구항 2

제1항에 있어서,

상기 제1 트랜잭션은 상기 등록 기관의 제2 공개 키에 잠긴 제2 출력을 포함하는,

컴퓨터-구현 방법.

청구항 3

제2항에 있어서,

상기 제1 출력은 상기 등록 기관의 제2 공개 키에 대해 시간-잠금되고, 시간 잠금은 미리 결정된 시간 기간 후까지 상기 제1 출력이 잠금해제되는 것을 방지하는,

컴퓨터-구현 방법.

청구항 4

제2항 또는 제3항에 있어서,

상기 제1 출력은 적어도 상기 등록 기관의 제2 공개 키 및 상이한 공개 키에 잠기는,

컴퓨터-구현 방법.

청구항 5

제1항 내지 제4항 중 어느 한 항에 있어서,

상기 제1 블록체인 트랜잭션의 트랜잭션 식별자를 허가 요청자에게 송신하는 단계를 포함하는,

컴퓨터-구현 방법.

청구항 6

제1항 내지 제5항 중 어느 한 항에 있어서,

상기 인증서는 암호화 키로 암호화되며, 상기 암호화 키는 상기 등록 기관에 의해 생성되는,

컴퓨터-구현 방법.

청구항 7

제1항 내지 제6항 중 어느 한 항에 있어서,

상기 네트워크에 참여하기 위한 상기 요청자로부터의 요청을 수신하는 단계 - 상기 요청은 하나 이상의 크리덴셜(credential)들을 포함함 - ; 및

상기 하나 이상의 크리덴셜들에 기초하여 상기 요청을 유효성 검증(validate)하는 단계를 포함하고,

상기 제1 블록체인 트랜잭션의 상기 생성은 상기 요청이 유효한 것을 조건으로 하는,

컴퓨터-구현 방법.

청구항 8

제1항 내지 제7항 중 어느 한 항에 있어서,

상기 브리징 노드들의 세트는 마스터 노드 및 상기 마스터 노드에 의해 제어 가능한 중개 노드들의 세트를 포함하고, 상기 등록 기관은 상기 마스터 노드인,

컴퓨터-구현 방법.

청구항 9

제1항 내지 제8항 중 어느 한 항에 있어서,

상기 브리징 노드들의 세트는 마스터 노드 및 상기 마스터 노드에 의해 제어 가능한 중개 노드들의 세트를 포함하고, 상기 요청자는 상기 마스터 노드인,

컴퓨터-구현 방법.

청구항 10

제1항 내지 제9항 중 어느 한 항에 있어서,

상기 요청자는 상기 블록체인 네트워크의 개개의 노드이고, 상기 인증서는 상기 허가 요청자에게 할당된 공개 키를 포함하는,

컴퓨터-구현 방법.

청구항 11

제1항 내지 제8항 중 어느 한 항에 있어서,

상기 요청자는 상기 제1 네트워크의 하나 이상의 브리징 노드들에 의해 제어 가능한 디바이스이고, 상기 방법은,

인증서들의 세트를 상기 요청자에게 송신하는 단계를 포함하고, 상기 세트의 각각의 인증서는 상기 노드들의 세트 중 개개의 하나에 송신되는,

컴퓨터-구현 방법.

청구항 12

제1항 내지 제11항 중 어느 한 항에 있어서,

상기 브리징 노드들의 세트 중 하나 이상에 상기 제1 인증서를 송신하는 단계를 포함하는,

컴퓨터-구현 방법.

청구항 13

제2항 또는 이에 의존하는 임의의 항에 있어서,

제2 블록체인 트랜잭션을 생성하는 단계 - 상기 제2 블록체인 트랜잭션은 상기 제1 트랜잭션의 제2 출력을 참

조하는 입력을 포함하고 상기 등록 기관의 제2 공개 키에 링크된 서명을 포함함 -- ; 및

상기 블록체인에의 포함을 위해 상기 블록체인 네트워크로 상기 제2 블록체인 트랜잭션을 송신하는 단계를 포함하는,

컴퓨터-구현 방법.

청구항 14

제1 네트워크에 참여하기 위한 허가를 요청하기 위한 컴퓨터-구현 방법으로서,

상기 제1 네트워크는 브리징 노드들의 세트 및 상기 브리징 노드들의 세트 중 하나 이상에 의해 제어 가능한 디바이스들의 세트를 포함하고, 각각의 브리징 노드는 또한 블록체인 네트워크의 개개의 노드이고;

상기 방법은, 요청자에 의해 수행되고,

상기 제1 네트워크 참여하기 위한 요청을 등록 기관에 송신하는 단계; 및

제1 인증서를 획득하는 단계를 포함하고, 상기 인증서는 상기 등록 기관에 의해 발행되고 상기 요청자에게 할당된 식별자를 포함하는,

컴퓨터-구현 방법.

청구항 15

제14항에 있어서,

상기 획득하는 단계는,

상기 제1 인증서를 포함하는 제1 블록체인 트랜잭션의 트랜잭션 식별자를 수신하는 단계; 및

상기 트랜잭션 식별자를 사용하여 상기 블록체인으로부터 상기 제1 블록체인 트랜잭션을 획득하는 단계를 포함하는,

컴퓨터-구현 방법.

청구항 16

제15항에 있어서,

상기 제1 블록체인 트랜잭션은 상기 인증서를 포함하는 제1 입력, 및 상기 등록 기관의 공개 키에 링크된 제2 출력을 포함하고, 상기 방법은,

상기 등록 기관의 공개 키를 식별하는 단계; 및

상기 등록 기관의 공개 키로부터 상기 블록체인으로 송신된 하나 이상의 개개의 트랜잭션들에 포함된 하나 이상의 제2 인증서들을 식별하는 단계를 포함하고, 각각의 제2 인증서는 개개의 브리징 노드 또는 디바이스 또는 상기 네트워크에 발행되는,

컴퓨터-구현 방법.

청구항 17

제16항에 있어서,

상기 제1 인증서는 상기 요청자의 공개 키를 포함하고, 상기 제1 네트워크의 상기 브리징 노드들의 세트의 개개의 하나에 발행된 각각의 제2 인증서는 상기 노드의 개개의 공개 키를 포함하고, 상기 방법은,

상기 브리징 노드들의 세트 중 적어도 하나에 제3 블록체인 트랜잭션을 송신하는 단계를 포함하고, 상기 제3 블록체인 트랜잭션은 상기 적어도 하나의 브리징 노드의 개개의 공개 키에 잠긴 출력을 포함하는,

컴퓨터-구현 방법.

청구항 18

제14항 내지 제17항 중 어느 한 항에 있어서,

상기 제1 인증서를 획득하는 단계는 상기 등록 기관으로부터 상기 제1 인증서를 수신하는 단계를 포함하는,

컴퓨터-구현 방법.

청구항 19

제14항 내지 제18항 중 어느 한 항에 있어서,

상기 등록 기관으로부터 하나 이상의 제2 인증서들을 수신하는 단계를 포함하고, 각각의 제2 인증서는 상기 제1 네트워크의 브리징 노드들 또는 디바이스들의 세트의 개개의 하나에 발행되는,

컴퓨터-구현 방법.

청구항 20

제19항에 있어서,

상기 제1 인증서는 상기 요청자의 네트워크 주소를 포함하고, 상기 제1 네트워크의 개개의 브리징 노드에 발행된 각각의 제2 인증서는 상기 노드의 개개의 네트워크 주소를 포함하고, 상기 방법은,

상기 브리징 노드들의 세트 중 하나 이상에 메시지를 송신하는 단계를 포함하고, 상기 메시지는 상기 요청자의 네트워크 주소로부터, 상기 메시지가 송신되는 하나 이상의 브리징 노드들의 개개의 네트워크 주소로 송신되는,

컴퓨터-구현 방법.

청구항 21

제14항 또는 제18항 내지 제20항 중 어느 한 항에 있어서,

상기 요청자는 상기 제1 네트워크의 상기 디바이스들의 세트 중 하나인,

컴퓨터-구현 방법.

청구항 22

제14항 내지 제21항 중 어느 한 항에 있어서,

상기 요청자는 상기 제1 네트워크의 상기 노드들의 세트 중 하나인,

컴퓨터-구현 방법.

청구항 23

제22항에 있어서,

상기 브리징 노드들의 세트는 마스터 노드 및 상기 마스터 노드에 의해 제어 가능한 하나 이상의 중개 노드들을 포함하고, 상기 요청자는 상기 마스터 노드인,

컴퓨터-구현 방법.

청구항 24

제14항 내지 제23항 중 어느 한 항에 있어서,

상기 요청은 상기 요청자의 하나 이상의 크리덴셜들을 포함하는,

컴퓨터-구현 방법.

청구항 25

제24항에 있어서,

상기 하나 이상의 크리덴셜들은 상기 요청자의 IP 주소를 포함하는,

컴퓨터-구현 방법.

청구항 26

컴퓨터 장비로서,

하나 이상의 메모리 유닛들을 포함하는 메모리; 및

하나 이상의 프로세싱 유닛들을 포함하는 프로세싱 장치를 포함하고,

상기 메모리는 상기 프로세싱 장치 상에서 실행되도록 배열된 코드를 저장하고, 상기 코드는 상기 프로세싱 장치 상에 있을 때, 제1항 내지 제13항 중 어느 한 항의 방법을 수행하도록 구성되는,

컴퓨터 장비.

청구항 27

컴퓨터-판독 가능 저장소 상에서 구체화되고, 제26항의 컴퓨터 장비 상에서 실행될 때 제1항 내지 제13항 중 어느 한 항의 방법을 수행하도록 구성된 컴퓨터 프로그램.

청구항 28

컴퓨터 장비로서,

하나 이상의 메모리 유닛들을 포함하는 메모리; 및

하나 이상의 프로세싱 유닛들을 포함하는 프로세싱 장치를 포함하고,

상기 메모리는 상기 프로세싱 장치 상에서 실행되도록 배열된 코드를 저장하고, 상기 코드는 상기 프로세싱 장치 상에 있을 때, 제14항 내지 제25항 중 어느 한 항의 방법을 수행하도록 구성되는,

컴퓨터 장비.

청구항 29

컴퓨터-판독 가능 저장소 상에서 구체화되고, 컴퓨터 장비 상에서 실행될 때 제14항 내지 제25항 중 어느 한 항의 방법을 수행하도록 구성된 컴퓨터 프로그램.

발명의 설명

기 술 분 야

[0001] 본 개시는 네트워크에 참여하도록 예컨대, 요청 엔티티가 네트워크에 액세스하기 위해 요청 엔티티에 허가를 부여하기 위한 방법들에 관한 것이다.

배 경 기 술

[0002] 블록체인은 분산 데이터 구조의 형태를 지칭하며, 여기에서 블록체인의 복제본이 피어-투-피어(Peer-to-Peer; P2P) 네트워크의 복수의 노드들 각각에서 유지된다. 블록체인은 데이터의 블록들의 체인을 포함하며, 각각의 블록은 하나 이상의 트랜잭션들을 포함한다. 각각의 트랜잭션은 하나 이상 블록들에 걸쳐 있을 수 있는 시퀀스에서 선행 트랜잭션을 뒤로 가리킬 수 있다. 트랜잭션들은 "채굴"로 알려진 프로세스에 의해 생성되는 새로운 블록들에 포함되도록 네트워크에 제출될 수 있으며, 이는 복수의 채굴 노드들 각각이 "작업 증명(proof-of-work)", 즉 블록들에 포함되기를 기다리는 보류 중인 트랜잭션들의 풀에 기초하여 암호화 퍼즐의 해결을 수행하기 위해 경쟁하는 것을 수반한다.

[0003] 종래에는, 블록체인의 트랜잭션들은 디지털 자산, 즉 가치 저장소로서 작용하는 데이터를 전달하는 데 사용된다. 그러나, 블록체인 위에 부가적인 기능성을 쌓기 위해 블록체인이 또한 활용될 수 있다. 예컨대, 블록체인 프로토콜들은 트랜잭션의 출력에의 부가적인 사용자 데이터의 저장을 허용할 수 있다. 현대의 블록체인들은 단일 트랜잭션 내에 저장될 수 있는 최대 데이터 용량을 증가시키고 있어, 보다 복잡한 데이터가 통합되는 것을 가능하게 한다. 예컨대, 이는 블록체인에 전자 문서를 저장하거나, 심지어 오디오 또는 비디오 데이터를 저장하는 데 사용될 수 있다.

[0004] 네트워크의 각각의 노드는 3개의 역할들: 포워딩, 채굴 및 저장 중 임의의 하나, 둘 또는 모두를 가질 수 있다. 포워딩 노드들은 네트워크의 노드들 전반에 걸쳐 트랜잭션들을 전파시킨다. 채굴 노드들은 블록들 내로의 트랜잭션들의 채굴을 수행한다. 저장 노드들은 블록체인의 채굴된 블록들의 그 자체의 사본을 각각 저장한다. 트랜잭션을 블록체인에 기록하기 위해, 당사자는 트랜잭션을 전파될 네트워크의 노드들 중 하나로 전송한다. 트랜잭션을 수신한 채굴 노드들은 트랜잭션을 새로운 블록 내로 채굴하기 위해 경쟁할 수 있다. 각각의 노드는 트랜잭션이 유효하기 위한 하나 이상의 조건들을 포함하는 동일한 노드 프로토콜을 준수하도록 구성된다. 유효하지 않은 트랜잭션들은 블록들 내로 채굴되거나 전파되지 않을 것이다. 트랜잭션이 유효성 검증되고 그리하여 블록체인 상에서 수락된다고 가정하면, 부가적인 사용자 데이터는 이에 따라 변경 불가능한 공개 레코드로서 P2P 네트워크의 노드들 각각에 저장된 채로 유지된다

발명의 내용

[0005] 사물 인터넷(IoT) 기술은 사람의 개입 없이 물리적 디바이스들의 네트워크들이 이벤트들을 모니터링하고 데이터를 교환하는 것을 가능하게 한다. IoT 기술의 개발의 동기는 광범위한 산업에 걸쳐 종래의 모니터링 및 제어 방법들을 대체하기 위한 실시간 데이터 수집 및 자동 제어 메커니즘들에 대한 필요성이다. IoT 시스템들은 대량의 데이터를 생성하고 네트워크 확장성, 강력한 사이버 보안, 신뢰할 수 있는 연결 및 최소 네트워크 레이턴시를 갖춘 시스템들에 의존한다.

[0006] 현재, 중앙화된 아키텍처 모델들은 IoT 네트워크에서 노드들을 인증, 인가 및 연결하는 데 널리 사용된다. 이러한 모델들은 공격에 취약하고 단일 장애점(single point of failure)으로서 작동한다. 중앙화된 시스템이 손상되는 경우, IoT 네트워크에 액세스하기 위한 허가(permission)가 악성 디바이스들에 부여되거나 기존 디바이스들로부터 제거될 수 있다. 예컨대, 악성 디바이스에 IoT 네트워크에 대한 액세스가 부여되는 경우, 그 디바이스는 예컨대, 민감한 데이터를 수집하거나 네트워크를 중단시킬 수 있다.

[0007] P2P(Peer-to-Peer) 아키텍처는 중앙화된 아키텍처에 비해 더 안전하고 효율적인 솔루션을 제공하며, 이에 의해 이웃들은 이들 간의 임의의 중앙화된 노드 또는 에이전트를 사용하지 않고 서로 직접 상호 작용한다. 블록체인 기술은 안전한 P2P 통신의 토대이며 IoT 시스템들의 개발에 혁명을 일으킬 것으로 기대된다. 그러나 IoT 디바이스들을 위한 차세대의 블록체인 기반 시스템들이 현실화되는 경우, IoT에 대한 블록체인 기반 제어 방법들이 개방 시스템들에 내재된 난제들을 극복할 필요가 있다. 이들은 블록체인 자체에 내재하지 않을 수 있는 데이터 프라이버시 및 디바이스 보호/제어 메커니즘들을 포함한다.

[0008] 본원에서 개시된 일 양상에 따르면, 제1 네트워크에 참여하도록 요청자에게 허가를 부여하기 위한 컴퓨터-구현 방법이 제공되며, 제1 네트워크는 브리징(bridging) 노드들의 세트 및 브리징 노드들의 세트 중 하나 이상에 의해 제어 가능한 디바이스들의 세트를 포함하고, 각각의 브리징 노드는 또한 블록체인 네트워크의 개개의 노드이고; 방법은, 등록 기관(registration authority)에 의해 수행되며, 제1 블록체인 트랜잭션을 생성하는 단계 - 제1 블록체인 트랜잭션은 등록 기관의 제1 공개 키에 링크된 서명을 포함하는 입력, 및 제1 인증서를 포함하는 제1 출력을 포함하고, 제1 인증서는 요청자에게 할당된 식별자를 포함함 - ; 및 블록체인에의 포함을 위해 블록체인 네트워크로 제1 블록체인 트랜잭션을 송신하는 단계를 포함한다.

[0009] 제1 네트워크(예컨대, IoT 네트워크)는 하나 이상의 브리징 노드들 및 브리징 노드들 중 하나 이상에 의해 제어될 수 있는 하나 이상의 디바이스들을 포함한다. 브리징 노드들은 또한 블록체인 네트워크의 노드들이다. 즉, 이들이 (예컨대, 다른 네트워크 노드들 및 디바이스들과 통신하기 위해) IoT 네트워크 및 (예컨대, 트랜잭션들을 블록체인에 송신하고 블록체인 상에 기록된 트랜잭션들을 식별하고 판독하기 위해) 블록체인 네트워크 둘 모두에 연결될 수 있다는 점에서, 이들은 IoT 네트워크 및 블록체인 네트워크의 부분이다. 이들 노드들은 제1 네트워크와 블록체인 네트워크 사이의 게이트웨이 또는 브리지로서 작용한다. 또한, 이들은 블록체인 네트워크의 채굴 노드들, 포워딩 노드들 또는 저장 노드들의 역할들을 가질 필요는 없지만, 이 역할들을 갖는 것이 또한 배제되지는 않는다. 일부 예들에서, 제1 네트워크의 디바이스들 중 하나 이상은 또한 블록체인 네트워크의 노드일 수 있다.

[0010] 등록 기관(IoT 네트워크의 브리징 노드일 수 있고 아닐 수도 있음)은 블록체인 네트워크의 노드이다. 즉, 등록 기관은 블록체인에 연결되고 블록체인 네트워크에 트랜잭션들을 송신하도록 구성된다. 등록 기관은 요청자들(요청 엔티티들)에게 인증서들을 부여하는 것을 담당하며, 이 인증서들은 그 후 네트워크에 참여하도록 요청자에 대한 허가를 부여한다. 여기에서 요청자가 네트워크에 참여하면, 요청자는 예컨대, 제1 네트워크에 연결된 디바이스들 및/또는 다른 노드들과 통신하는 것과 같은 작업들을 수행할 수 있다. 등록 기관은 새로운 디바이

스들의 크리덴셜(credential)들을 검증하고 디지털 인증서들을 발행할 수 있는 신뢰 당사자이다. 등록 기관은 진입에 대한 장벽으로서 작용하고 IoT 네트워크 상에서 새로운 디바이스들을 허용하거나 허용하지 않는다.

[0011] 제1 블록체인 트랜잭션은 트랜잭션의 출력에서, 요청자에게 발행된 인증서를 포함한다. 인증서는 요청자에게 고유한 식별자("디바이스 ID")를 포함한다. 디바이스 ID는 바이트(bytes)의 의사 난수 스트링일 수 있다. 등록 기관은 트랜잭션(및 이에 따른 인증서)을 블록체인 네트워크에 브로드캐스트한다. 제1 블록체인 트랜잭션은 제1 입력의 서명을 통해 등록 기관에 링크된다. 서명은 등록 기관의 공개 키에 링크된다. 즉, 서명은 등록 기관의 공개 키에 대응하는 개인 키에 기초하여 생성된다. 등록 기관만이 개인 키에 관한 지식을 갖고, 이에 따라 등록 기관만이 그의 서명으로 트랜잭션에 서명할 수 있다. 즉, 인증서는 위조될 수 없다. 따라서 인증서들은 진성(genuine) 엔티티들에만 부여될 수 있다. 일단 인증서가 발행된 요청자가 제1 네트워크의 다른 노드들 또는 디바이스들과 통신할 때, 그 노드들 및 디바이스들은 요청자에게 인증서가 발행되었는지를 체크하고 이에 따라 요청자가 진성 엔티티인지를 체크할 수 있다.

[0012] 일부 실시예들에서, 요청자는 또한 블록체인 네트워크의 노드이다. 이 경우에, 인증서는 요청 노드에 할당된 공개 키를 포함할 수 있다. 즉, 공개 키는 보증된 공개 키이다. 요청 노드가 그의 보증된 공개 키 또는 그로부터 도출된 키를 사용하여 다른 노드들과 통신하는 경우(예컨대, 상기 공개 키 중 하나로 서명된 블록체인 또는 노드들에 트랜잭션들을 전송함), 다른 노드들은 이러한 트랜잭션들을 전송하는 노드가 진성 노드라고 확신할 수 있다.

[0013] 본원에서 개시된 다른 양상에 따르면, 제1 네트워크에 참여하기 위한 허가를 요청하기 위한 컴퓨터-구현 방법이 제공되며, 제1 네트워크는 브리징 노드들의 세트 및 브리징 노드들의 세트 중 하나 이상에 의해 제어 가능한 디바이스들의 세트를 포함하고, 각각의 브리징 노드는 또한 블록체인 네트워크의 개개의 노드이고; 방법은, 요청자에 의해 수행되며, 제1 네트워크에 참여하기 위한 요청을 등록 기관에 송신하는 단계; 및 제1 인증서를 획득하는 단계를 포함하고, 인증서는 등록 기관에 의해 발행되고 요청자에게 할당된 식별자를 포함한다.

도면의 간단한 설명

[0014] 본 개시의 실시예들의 이해를 보조하기 위해 그리고 그러한 실시예들이 어떻게 실행될 수 있는지를 보여주기 위하여, 단지 예로서 첨부 도면들에 대한 참조가 이루어진다.

도 1은 블록체인을 구현하기 위한 시스템의 개략적인 블록도이다.

도 2는 블록체인에 기록될 수 있는 트랜잭션들의 일부 예들을 개략적으로 예시한다.

도 3은 블록체인을 구현하기 위한 다른 시스템의 개략적인 블록도이다.

도 4는 출력-기반 모델의 노드 프로토콜에 따라 트랜잭션들을 프로세싱하기 위한 노드 소프트웨어 조각의 개략적인 블록도이다.

도 5는 IoT 네트워크와 블록체인 네트워크 간의 오버랩을 개략적으로 예시한다.

도 6은 계층적 네트워크 토폴로지를 개략적으로 예시한다.

도 7a 및 도 7b는 부분 및 완료 커맨드 트랜잭션들을 개략적으로 예시한다.

도 8a 및 도 8b는 대안적인 부분 및 완료 트랜잭션들을 개략적으로 예시한다.

도 9는 커맨드 요청 및 응답 사이클을 개략적으로 예시한다.

도 10a 및 도 10b는 서버 노드로부터 슬레이브 노드로 송신되는 부분 및 완료 커맨드 트랜잭션들을 개략적으로 예시한다.

도 11a 및 도 11b는 커맨드 요청 및 승인 트랜잭션들을 개략적으로 예시한다.

도 12a 및 도 12b는 암호화된 부분 및 완료 커맨드 트랜잭션들을 개략적으로 예시한다.

도 13은 예시적인 커맨드 데이터 포맷을 예시한다.

도 14는 예시적인 피어 투 피어 인쇄 시스템을 개략적으로 예시한다.

도 15a 내지 도 15c는 피어 투 피어 인쇄 시스템에서 사용하기 위한 예시적인 트랜잭션을 개략적으로 예시한다.

도 16a 및 도 16b는 인증서 트랜잭션 및 예시적인 인증서 포맷을 개략적으로 예시한다.

발명을 실시하기 위한 구체적인 내용

[0015] **예시적인 시스템 개요**

[0016] 도 1은 일반적으로 블록체인(150)을 구현하기 위한 예시적인 시스템(100)을 도시한다. 시스템(100)은 패킷-교환 네트워크(101), 통상적으로 인터넷과 같은 광역 인터네트워크를 포함한다. 패킷-교환 네트워크(101)는 패킷-교환 네트워크(101) 내에서 P2P(peer-to-peer) 오버레이 네트워크(106)를 형성하도록 배열된 복수의 노드들(104)을 포함한다. 각각의 노드(104)는 피어들의 컴퓨터 장비를 포함하며, 노드들(104) 중 상이한 노드들은 상이한 피어들에 속한다. 각각의 노드(104)는 하나 이상의 프로세서들, 예컨대, 하나 이상의 CPU(central processing unit)들, 가속기 프로세서들, 애플리케이션 특정 프로세서 및/또는 FPGA(field programmable gate array)들을 포함하는 프로세싱 장치를 포함한다. 각각의 노드는 또한 메모리, 즉 비-일시적 컴퓨터-판독 가능 매체 또는 매체들의 형태의 컴퓨터-판독 가능 저장소를 포함한다. 메모리는 하나 이상의 메모리 매체들, 예컨대, 하드 디스크와 같은 자기 매체; 솔리드 스테이트 드라이브(SSD), 플래시 메모리 또는 EEPROM과 같은 전자 매체; 및/또는 광학 디스크 드라이브와 같은 광학 매체를 사용하는 하나 이상의 메모리 유닛들을 포함할 수 있다.

[0017] 블록체인(150)은 데이터의 블록들의 체인(151)을 포함하며, 블록체인(150)의 개개의 사본이 P2P 네트워크(160)의 복수의 노드들 각각에서 유지된다. 체인의 각각의 블록(151)은 하나 이상의 트랜잭션들(152)을 포함하며, 여기서 이 맥락에서 트랜잭션은 일종의 데이터 구조를 지칭한다. 데이터 구조의 성질은 트랜잭션 모델 또는 체계(scheme)의 일부로서 사용되는 트랜잭션 프로토콜의 유형에 의존할 것이다. 주어진 블록체인은 통상적으로 전반에 걸쳐 하나의 특정 트랜잭션 프로토콜을 사용할 것이다. 하나의 공통 유형의 트랜잭션 프로토콜에서, 각각의 트랜잭션(152)의 데이터 구조는 적어도 하나의 입력 및 적어도 하나의 출력을 포함한다. 각각의 출력은, 출력이 암호학적으로 잠겨 있는 사용자(103)에 속한 디지털 자산의 양을 표현하는 금액을 지정한다(잠금해제되고 그리하여 리딤(redeem) 또는 지출되기 위해 그 사용자의 서명을 요구함). 각각의 입력은 선행 트랜잭션(152)의 출력을 뒤로 가리키고, 그리하여 트랜잭션들을 링크한다.

[0018] 노드들(104) 중 적어도 일부는 트랜잭션들(152)을 포워딩하고 그리하여 전파시키는 포워딩 노드들(104F)의 역할을 맡는다. 노드들(104) 중 적어도 일부는 블록들(151)을 채굴하는 채굴자들(104M)의 역할을 맡는다. 노드들(104) 중 적어도 일부는, 각각이 각자의 메모리에 동일한 블록체인(150)의 개개의 사본을 저장하는 저장 노드들(104S)(때로는 또한 "전체-사본(full-copy)" 노드들이라 칭함)의 역할을 맡는다. 각각의 채굴자 노드(104M)는 또한 블록(151)으로 채굴되기를 기다리는 트랜잭션들(152)의 풀(154)을 유지한다. 주어진 노드(104)는 포워딩 노드(104), 채굴자(104M), 저장 노드(104S) 또는 이들 중 2개 또는 모두의 임의의 조합일 수 있다.

[0019] 주어진 현재 트랜잭션(152j)에서, 그(또는 각각의) 입력은 그러한 출력이 현재 트랜잭션(152j)에서 "지출"되거나 리딤됨을 지정하도록 트랜잭션들의 시퀀스에서 선행 트랜잭션(152i)의 출력을 참조하는 포인터를 포함한다. 일반적으로, 선행 트랜잭션은 풀(154) 또는 임의의 블록(151)의 임의의 트랜잭션일 수 있다. 선행 트랜잭션(152i)은 현재 트랜잭션(152j)이 생성되거나 심지어 네트워크(106)로 전송될 때 반드시 존재할 필요는 없지만, 선행 트랜잭션(152i)은 현재 트랜잭션이 유효하기 위해 존재하고 유효성 검증될 필요가 있을 것이다. 따라서 본원에서 "선행(preceding)"이라 함은 포인터들에 의해 링크된 논리적 시퀀스의 선행자를 지칭하며, 반드시 시간적 시퀀스의 전송 또는 생성 시간은 아니고, 따라서 트랜잭션들(152i, 152j)은 순서와 다르게(out-of-order)(고아 트랜잭션들에 대한 아래 논의 참조) 전송되거나 생성되는 것을 반드시 배제하지 않는다. 선행 트랜잭션(152i)은 앞선(antecedent) 트랜잭션 또는 선행자(predecessor) 트랜잭션으로 동등하게 칭해질 수 있다.

[0020] 현재 트랜잭션(152j)의 입력은 또한 선행 트랜잭션(152i)의 출력이 잠겨 있는 사용자(103a)의 서명을 포함한다. 차례로, 현재 트랜잭션(152j)의 출력은 새로운 사용자(103b)에 대해 암호학적으로 잠길 수 있다. 따라서 현재 트랜잭션(152j)은 선행 트랜잭션(152i)의 입력에서 정의된 금액을 현재 트랜잭션(152j)의 출력에서 정의된 바와 같은 새로운 사용자(103b)에게 전달할 수 있다. 일부 경우들에서, 트랜잭션(152)은 다수의 사용자들(이들 중 하나는 잔액(change)을 주기 위해 오리지널 사용자(103a)일 수 있음) 사이에서 입력 금액을 분할하기 위해 다수의 출력들을 가질 수 있다. 일부 경우들에서, 트랜잭션은 또한 하나 이상의 선행 트랜잭션들의 다수의 출력들로부터 금액들을 수집하고 현재 트랜잭션의 하나 이상의 출력들에 재분배하기 위해 다수의 입력들을 가질 수 있다.

[0021] 위의 내용은 "출력-기반" 트랜잭션 프로토콜로서 지칭될 수 있으며, 때로는 또한 미지출 트랜잭션 출력(unspent

transaction output; UTXO) 유형 프로토콜(여기서 출력들은 UTXO들로서 지칭됨)로서 지칭된다. 사용자의 총 잔고는 블록체인에 저장된 임의의 하나의 숫자에서 정의되지 않으며, 대신 사용자는 블록체인(151)의 다수의 상이한 트랜잭션들(152) 전반에 걸쳐 흩어져 있는 그 사용자의 모든 UTXO들의 값들을 대조하기 위해 특별한 "지갑" 애플리케이션(105)을 필요로 한다.

[0022] 트랜잭션 프로토콜의 대안적인 유형은 계정-기반 트랜잭션 모델의 일부로서 "계정-기반" 프로토콜로서 지칭될 수 있다. 계정-기반의 경우에, 각각의 트랜잭션은 과거 트랜잭션들의 시퀀스에서 선행 트랜잭션의 UTXO를 뒤로 참조하기 보다는, 절대 계정 잔액을 참조함으로써 전달될 금액을 정의한다. 모든 계정들의 현재 상태는 블록체인과 별개로 채굴자들에 의해 저장되며 지속적으로 업데이트된다. 이러한 시스템에서, 트랜잭션들은 (또한 "포지션"이라 불리는) 계정의 실행 중인 트랜잭션 총계를 사용하여 순서화된다. 이 값은 그의 암호화 서명의 일부로 발신인에 의해 서명되고 트랜잭션 참조 계산의 부분으로서 해시된다. 게다가, 선택적 데이터 필드가 또한 트랜잭션에 서명할 수 있다. 이 데이터 필드는 예컨대, 이전 트랜잭션 ID가 데이터 필드에 포함된 경우 이전 트랜잭션을 뒤로 가리킬 수 있다.

[0023] 어느 유형의 트랜잭션 프로토콜이든, 사용자(103)가 새로운 트랜잭션(152j)을 시행(enact)하기를 원할 때, 그/그녀는 자신의 컴퓨터 단말(102)로부터 P2P 네트워크(106)의 노드들(104) 중 하나(이는 최근에는 통상적으로 서버들 또는 데이터 센터이지만, 원칙적으로 다른 사용자 단말들일 수 있음)로 새로운 트랜잭션을 전송한다. 이 노드(104)는 노드들(104) 각각에 적용되는 노드 프로토콜에 따라 트랜잭션이 유효한지를 체크한다. 노드 프로토콜의 세부사항들은 해당 블록체인(150)에서 사용되는 트랜잭션 프로토콜의 유형에 대응하며, 전체 트랜잭션 모델을 함께 형성한다. 노드 프로토콜은 통상적으로 노드(104)가 새로운 트랜잭션(152j)의 암호화 서명이 예상되는 서명과 매칭되는지를 체크하도록 요구하며, 이는 트랜잭션들(152)의 순서화된 시퀀스에서 이전 트랜잭션(152i)에 의존한다. 출력-기반의 경우에, 이는 새로운 트랜잭션(152j)의 입력에 포함된 사용자의 암호화 서명이 새로운 트랜잭션이 지출하는 선행 트랜잭션(152i)의 출력에 정의된 조건과 매칭되는지를 체크하는 것을 포함하며, 여기에서 이 조건은 통상적으로 적어도 새로운 트랜잭션(152j)의 입력의 암호화 서명이 새로운 트랜잭션의 입력이 가리키는 이전 트랜잭션(152i)의 출력을 잠금해제한다는 것을 체크하는 것을 포함한다. 일부 트랜잭션 프로토콜들에서, 조건은 입력 및/또는 출력에 포함된 사용자 정의 스크립트(custom script)에 의해 적어도 부분적으로 정의될 수 있다. 대안적으로 이는 단순히 노드 프로토콜만으로 고정되거나, 이들의 조합으로 인한 것일 수 있다. 어느 쪽이든, 새로운 트랜잭션(152j)이 유효한 경우, 현재 노드는 이를 P2P 네트워크(106)의 노드들(104) 중 하나 이상의 다른 노드들에 포워딩한다. 이러한 노드(104) 중 적어도 일부는 또한 포워딩 노드(104F)로서 작용하여, 동일한 노드 프로토콜에 따른 동일한 테스트를 적용하고, 이에 따라 새로운 트랜잭션(152j)을 하나 이상의 추가 노드들(104)로 포워딩하는 식이다. 이러한 방식으로, 새로운 트랜잭션은 노드들(104)의 네트워크 전반에 걸쳐 전파된다.

[0024] 출력-기반 모델에서, 주어진 출력(예컨대, UTXO)이 지출되는지 여부에 대한 정의는 그것이 노드 프로토콜에 따라 다른 전방 트랜잭션(152j)의 입력에 의해 유효하게 리딤되었는지의 여부이다. 트랜잭션이 유효하기 위한 다른 조건은 지출 또는 리딤을 시도하는 선행 트랜잭션(152i)의 출력이 다른 유효한 트랜잭션에 의해 이미 지출/리딤되지 않은 것이다. 재차, 유효하지 않은 경우, 트랜잭션(152j)은 블록체인에 기록되거나 전파되지 않을 것이다. 이는 지출자가 동일한 트랜잭션의 출력을 한번 초과로 지출하고자 시도하는 이중-지출을 경계한다. 반면, 계정-기반 모델은 계정 잔액을 유지함으로써 이중-지출을 경계한다. 재차, 트랜잭션들의 정의된 순서가 존재하기 때문에, 계정 잔액은 임의의 한 시간에 단일의 정의된 상태를 갖는다.

[0025] 유효성 검증에 추가하여, 노드들(104M) 중 적어도 일부는 또한 채굴로서 알려진 프로세스에서 트랜잭션들의 블록들을 최초로 생성하기 위해 경쟁하며, 이는 "작업 증명"에 의해 뒷받침된다. 채굴 노드(104M)에서, 아직 블록에 나타나지 않은 유효한 트랜잭션들의 풀에 새로운 트랜잭션들이 추가된다. 그 후, 채굴자들은 암호화 퍼즐을 해결하도록 시도함으로써 트랜잭션들의 풀(154)로부터 트랜잭션들(152)의 새로운 유효한 블록(151)을 조립하기 위해 경쟁한다. 통상적으로 이는 "논스(nonce)"가 트랜잭션들(154)의 풀과 연결되고(concatenated) 해시될 때, 해시의 출력이 미리 결정된 조건을 충족시키도록 논스 값을 검색하는 것을 포함한다. 예컨대, 미리 결정된 조건은 해시의 출력이 미리 정의된 특정 수의 선행 0들을 갖는 것일 수 있다. 해시 함수의 특성은 해시 함수가 그의 입력에 대해 예측 불가능한 출력을 갖는다는 것이다. 따라서 이 검색은 무차별 대입(brute force)에 의해서만 수행될 수 있고, 이에 따라 퍼즐을 해결하고자 하는 각각의 노드(104M)에서 상당한 양의 프로세싱 자원을 소비한다.

[0026] 퍼즐을 해결하고자 하는 제1 채굴자 노드(104M)는 이를 네트워크(106)에 발표하고, 그 해(solution)를 증명으로서 제공하며, 이는 그 후 네트워크의 다른 노드(104)들에 의해 쉽게 체크될 수 있다(해시에 대한 해가

주어지면, 그 해가 해시의 출력으로 하여금 조건을 충족시키게 한다는 것을 체크하는 것은 간단함). 승자가 퍼즐을 해결한 트랜잭션들의 풀(154)은 각각의 그러한 노드에서 승자의 발표된 해를 체크한 것에 기초하여, 저장 노드들(104S)로서 작용하는 노드들(104) 중 적어도 일부에 의해 블록체인(150)에 새로운 블록(151)으로서 기록된다. 블록 포인터(155)가 또한 체인에서 이전에 생성된 블록(151n-1)을 뒤로 가리키는 새로운 블록(151n)에 할당된다. 작업 증명은 새로운 블록(151)을 생성하는 데 대량의 노력이 필요하므로 이중 지출의 위험을 감소시키는데 도움이 되고, 이중 지출을 포함하는 임의의 블록은 다른 노드들(104)에 의해 거부될 가능성이 높기 때문에, 채굴 노드들(104M)은 이중 지출들이 그의 블록들에 포함되는 것을 허용하지 않도록 장려된다. 일단 생성되면, 블록(151)은 수정될 수 없는데, 그 이유는 그것이 동일한 프로토콜에 따라 P2P 네트워크(106)의 저장 노드들(104S) 각각에서 인식 및 유지되기 때문이다. 블록 포인터(155)는 또한 블록들(151)에 순차적인 순서를 부과한다. 트랜잭션들(152)은 P2P 네트워크(106)의 각각의 저장 노드(104S)에서 순서화된 블록들에 기록되기 때문에, 이는 이에 따라, 트랜잭션들의 변경 불가능한 공개 원장을 제공한다.

[0027] 임의의 주어진 시간에 퍼즐을 해결하기 위해 경쟁하는 상이한 채굴자들(104M)은, 해의 검색을 시작한 시기에 의존하여, 임의의 주어진 시간에 채굴되지 않은 트랜잭션 풀(154)의 상이한 스냅샷들에 기초하여 퍼즐을 해결다는 것에 주의한다. 각자의 퍼즐을 먼저 해결하는 사람은 어느 트랜잭션들(152)이 다음의 새로운 블록(151n)에 포함되는지를 정의하고, 채굴되지 않은 트랜잭션들의 현재 풀(154)은 업데이트된다. 그 후 채굴자들(104M)은 새롭게 정의된 미해결 풀(154)로부터 블록을 생성하기 위해 계속 경쟁하며, 이와 같이 계속된다. 발생할 수 있는 임의의 "포크(fork)" – 이는 2개의 채굴자들(104M)이 서로 매우 짧은 시간 내에 그의 퍼즐을 해결하여서, 블록체인에 대한 상충되는 뷰(view)가 전파되는 경우임 – 를 해결하기 위한 프로토콜이 또한 존재한다. 요컨대, 가장 길게 성장하는 포크의 갈래가 확정적인 블록체인(150)이 된다.

[0028] 대부분의 블록체인들에서, (한 사용자로부터 다른 사용자에게 디지털 자산의 금액을 전달하는 일반 트랜잭션들과 대조적으로) 승리한 채굴자(104M)는 새로운 수량의 디지털 자산을 어디에선지 모르게 생성하는 특별한 종류의 새로운 트랜잭션으로 자동으로 보상된다. 따라서 승리한 노드는 일정 수량의 디지털 자산을 "채굴"한 것으로 여겨진다. 이 특별한 유형의 트랜잭션은 때로는 "생성(generation)" 트랜잭션으로서 지칭된다. 이는 새로운 블록(151n)의 부분을 자동으로 형성한다. 이 보상은 채굴자들(104M)이 작업 증명 경쟁에 참여하도록 하는 인센티브를 제공한다. 종종 일반(비-생성) 트랜잭션(152)이 또한 그 트랜잭션이 포함된 블록(151n)을 생성한 승리한 채굴자(104M)를 추가로 보상하기 위해, 그의 출력들 중 하나에 부가적인 트랜잭션 수수료를 지정할 것이다.

[0029] 채굴에 수반되는 컴퓨테이셔널 자원으로 인해, 통상적으로 적어도 채굴자 노드들(104M) 각각은 하나 이상의 물리적 서버 유닛들, 또는 심지어 전체 데이터 센터를 포함하는 서버의 형태를 취한다. 각각의 포워딩 노드(104M) 및/또는 저장 노드(104S)는 또한 서버 또는 데이터 센터의 형태를 취할 수 있다. 그러나 원칙적으로 임의의 주어진 노드(104)는 사용자 단말 또는 함께 네트워킹된 사용자 단말들의 그룹의 형태를 취할 수 있다.

[0030] 각각의 노드(104)의 메모리는 노드 프로토콜에 따라 각자의 역할 또는 역할들을 수행하고 트랜잭션들(152)을 처리하기 위해 노드(104)의 프로세싱 장치 상에서 실행되도록 구성된 소프트웨어를 저장한다. 본원에서 노드(104)에 기인한 임의의 동작은 각자의 컴퓨터 장비의 프로세싱 장치 상에서 실행되는 소프트웨어에 의해 수행될 수 있다는 것이 이해될 것이다. 또한, 본원에서 사용하는 바와 같은 "블록체인"이라는 용어는 일반적으로 기술의 종류를 지칭하는 일반 용어이며, 임의의 특정 사유 블록체인, 프로토콜 또는 서비스로 제한하지 않는다.

[0031] 또한 네트워크(101)에는 소비 사용자들의 역할을 하는 복수의 당사자들(103) 각각의 컴퓨터 장비(102)가 연결되어 있다. 이들은 트랜잭션들에서 지급인들 및 수취인들로서 작용하지만, 다른 당사자들을 대신하여 트랜잭션들을 채굴하거나 전파시키는 데 반드시 참여할 필요는 없다. 이들은 반드시 채굴 프로토콜을 실행할 필요는 없다. 제1 당사자(103a) 및 그/그녀의 개개의 컴퓨터 장비(102a) 및 제2 당사자(103b) 및 그/그녀의 개개의 컴퓨터 장비(102b)인 두 당사자들(103) 및 이들의 개개의 장비(102)가 예시 목적으로 도시된다. 훨씬 더 많은 이러한 당사자들(103) 및 이들의 개개의 컴퓨터 장비(102)가 존재하고 시스템에 참여할 수 있지만, 편의상 그것들은 예시되지 않는다는 것이 이해될 것이다. 각각의 당사자(103)는 개인 또는 조직일 수 있다. 순전히 예시로서, 제1 당사자(103a)는 본원에서 앨리스(Alice)로서 지칭되고 제2 당사자(103b)는 밥(Bob)으로서 지칭되지만, 이것이 제한적이지 않고 본원에서 앨리스 또는 밥에 대한 임의의 참조는 각각 "제1 당사자" 및 "제2 당사자"로 대체될 수 있다는 것이 인지될 것이다.

[0032] 각각의 당사자(103)의 컴퓨터 장비(102)는 하나 이상의 프로세서들, 예컨대, 하나 이상의 CPU들, GPU들, 다른 가속기 프로세서들, 애플리케이션 특정 프로세서들 및/또는 FPGA들을 포함하는 개개의 프로세싱 장치를 포함한

다. 각각의 당사자(103)의 컴퓨터 장비(102)는 메모리, 즉 비-일시적 컴퓨터-판독 가능 매체 또는 매체들의 형태의 컴퓨터-판독 가능 저장소를 더 포함한다. 이 메모리는 하나 이상의 메모리 매체들, 예컨대, 하드 디스크와 같은 자기 매체; 솔리드 스테이트 SSD, 플래시 메모리 또는 EEPROM과 같은 전자 매체; 및/또는 광학 디스크 드라이브와 같은 광학 매체를 사용하는 하나 이상의 메모리 유닛들을 포함할 수 있다. 각각의 당사자(103)의 컴퓨터 장비(102) 상의 메모리는 프로세싱 장치 상에서 실행되도록 배열된 적어도 하나의 클라이언트 애플리케이션(105)의 개개의 인스턴스를 포함하는 소프트웨어를 저장한다. 본원에서 주어진 당사자(103)에 기인한 임의의 동작은 개개의 컴퓨터 장비(102)의 프로세싱 장치 상에서 실행되는 소프트웨어를 사용하여 수행될 수 있다는 것이 이해될 것이다. 각각의 당사자(103)의 컴퓨터 장비(102)는 적어도 하나 사용자 단말, 예컨대, 데스크 톱 또는 랩톱 컴퓨터, 태블릿, 스마트폰, 또는 스마트워치와 같은 웨어러블 디바이스를 포함한다. 주어진 당사자(103)의 컴퓨터 장비(102)는 또한 사용자 단말을 통해 액세스되는 클라우드 컴퓨팅 자원들과 같은 하나 이상의 다른 네트워킹된 자원들을 포함할 수 있다.

[0033] 예컨대, 서버로부터 다운로드되거나, 또는 이동식 저장 디바이스 이를테면, 이동식 SSD, 플래시 메모리 키, 이동식 EEPROM, 이동식 자기 디스크 드라이브, 자기 플로피 디스크 또는 테이프, 광학 디스크 이를테면, CD 또는 DVD ROM 또는 이동식 광학 드라이브 등 상에서 제공되는 클라이언트 애플리케이션 또는 소프트웨어(105)는 적절한 컴퓨터-판독 가능 저장 매체 상에서 임의의 주어진 당사자(103)의 컴퓨터 장비(102)에 초기에 제공될 수 있다.

[0034] 클라이언트 애플리케이션(105)은 적어도 "지갑" 기능을 포함한다. 이는 2개의 메인 기능성들을 갖는다. 이들 중 하나는 개개의 사용자 당사자(103)가 노드들(104)의 네트워크 전반에 전파되고 그리하여 블록체인(150)에 포함될 트랜잭션들(152)을 생성, 서명 및 전송하는 것을 가능하게 하는 것이다. 남은 하나는 개개의 당사자에게 자신이 현재 소유하고 있는 디지털 자산의 금액을 다시 보고하는 것이다. 출력-기반 시스템에서, 이 제2 기능성은 블록체인(150) 전반에 걸쳐 흩어져 있는 해당 당사자에 속하는 다양한 트랜잭션들(152)의 출력들에서 정의된 금액들을 대조하는 것을 포함한다.

[0035] 각각의 컴퓨터 장비(102) 상의 클라이언트 애플리케이션(105)의 인스턴스는 P2P 네트워크(106)의 포워딩 노드들(104F) 중 적어도 하나에 동작 가능하게 커플링된다. 이는 클라이언트(105)의 지갑 기능이 트랜잭션들(152)을 네트워크(106)로 전송하는 것을 가능하게 한다. 클라이언트(105)는 또한 개개의 당사자(103)가 수령인인 임의의 트랜잭션들에 대해 블록체인(150)에 질의하기 위해(또는 실시예들에서, 블록체인(150)은 그의 공개 가시성을 통해 부분적으로 트랜잭션들의 신뢰를 제공하는 공공 시설(public facility)이므로, 실제로 블록체인(150)에서 다른 당사자들의 트랜잭션을 검사하기 위해) 저장 노드들(104) 중 하나, 일부 또는 전부에 접촉할 수 있다. 각각의 컴퓨터 장비(102) 상의 지갑 기능은 트랜잭션 프로토콜에 따라 트랜잭션들(152)을 공식화(formulate) 하고 전송하도록 구성된다. 각각의 노드(104)는 노드 프로토콜에 따라 트랜잭션들(152)을 유효성 검증하도록, 그리고 포워딩 노드(104F)의 경우에는 네트워크(106) 전반에 걸쳐 트랜잭션들(152)을 포워딩하도록 구성된 소프트웨어를 실행한다. 트랜잭션 프로토콜 및 노드 프로토콜은 서로 대응하며, 주어진 트랜잭션 프로토콜은 주어진 트랜잭션 모델을 함께 구현하도록 주어진 노드 프로토콜을 따른다. 블록체인(150)의 모든 트랜잭션들(152)에 대해 동일한 트랜잭션 프로토콜이 사용된다(그러나 트랜잭션 프로토콜은 그 내부에서 상이한 하위유형들의 트랜잭션을 허용할 수 있음). 동일한 노드 프로토콜이 네트워크(106)의 모든 노드들(104)에 의해 사용된다(그러나 이는 그 하위유형에 대해 정의된 규칙들에 따라 상이한 하위유형들을 트랜잭션을 상이하게 처리할 수 있으며, 또한 상이한 노드들은 상이한 역할들을 맡고 이에 따라 프로토콜의 상이한 대응하는 양상들을 구현할 수 있음).

[0036] 언급된 바와 같이, 블록체인(150)은 블록들의 체인(151)을 포함하며, 여기서 각각의 블록(151)은 이전에 논의된 바와 같이 작업 증명 프로세스에 의해 생성된 하나 이상의 트랜잭션들(152)의 세트를 포함한다. 각각의 블록(151)은 또한 블록들(151)에 대한 순차적인 순서를 정의하기 위해 체인에서 이전에 생성된 블록(151)을 뒤로 가리키는 블록 포인터(155)를 포함한다. 블록체인(150)은 또한 작업 증명 프로세스에 의해 새로운 블록에 포함되기를 기다리는 유효 트랜잭션들의 풀(154)을 포함한다. (생성 트랜잭션 외의) 각각의 트랜잭션(152)은 트랜잭션들의 시퀀스들에 대한 순서를 정의하기 위해 이전 트랜잭션에 대한 역 포인터를 포함한다(트랜잭션들(152)의 시퀀스들은 분기가 허용됨을 주의함). 블록들의 체인(151)은 체인의 제1 블록이었던 제네시스(genesis) 블록(Gb)(153)까지 완전히 거슬러 올라간다. 체인(150) 초반의 하나 이상의 오리지널 트랜잭션들(152)은 선행 트랜잭션이 아닌 제네시스 블록(153)을 가리켰다.

[0037] 주어진 당사자(103), 이를테면 앨리스가 블록체인(150)에 포함될 새로운 트랜잭션(152j)을 전송하기를 원할 때, 그녀는 (자신의 클라이언트 애플리케이션(105)의 지갑 기능을 사용하여) 관련 트랜잭션 프로토콜에 따라 새로운

트랜잭션을 공식화한다. 그 후, 그녀는 클라이언트 애플리케이션(105)으로부터 그녀가 연결되는 하나 이상의 포워딩 노드들(104F) 중 하나로 트랜잭션(152)을 전송한다. 예컨대, 이는 앨리스의 컴퓨터(102)에 가장 가깝거나 가장 잘 연결된 포워딩 노드(104F)일 수 있다. 임의의 주어진 노드(104)가 새로운 트랜잭션(152j)을 수신할 때, 주어진 노드는 노드 프로토콜 및 각자의 역할에 따라 이를 처리한다. 이는 새로 수신된 트랜잭션(152j)이 "유효"하기 위한 특정 조건을 충족시키는지를 먼저 체크하는 것을 포함하며, 그의 예들은 곧 보다 자세히 논의될 것이다. 일부 트랜잭션 프로토콜들에서, 유효성 검증을 위한 조건은 트랜잭션들(152)에 포함된 스크립트들에 의해 트랜잭션 단위로 구성 가능할 수 있다. 대안적으로, 조건은 단순히 노드 프로토콜의 내장 피처이거나, 스크립트 및 노드 프로토콜의 조합으로 정의될 수 있다.

[0038] 새로 수신된 트랜잭션(152j)이 유효한 것으로 간주되기 때문에 테스트를 통과한다는 것을 조건으로(즉, 그것이 "유효성 검증"된다는 조건으로), 트랜잭션(152j)을 수신하는 임의의 저장 노드(104S)는 새로운 유효성 검증된 트랜잭션(152)을 그 노드(104S)에서 유지되는 블록체인(150)의 사본의 풀(154)에 추가할 것이다. 또한, 트랜잭션(152j)을 수신하는 임의의 포워딩 노드(104F)는 유효성 검증된 트랜잭션(152)을 P2P 네트워크(106)의 하나 이상의 다른 노드들(104)로 전방으로 전파시킬 것이다. 각각의 포워딩 노드(104F)가 동일한 프로토콜을 적용하기 때문에, 트랜잭션(152j)이 유효하다고 가정하면, 이는 그것이 곧 전체 P2P 네트워크(106)에 걸쳐 전파될 것임을 의미한다.

[0039] 하나 이상의 저장 노드들(104)에서 유지되는 블록체인(150)의 사본의 풀(154)에 허여되면, 채굴자 노드들(104M)은 새로운 트랜잭션(152)을 포함하는 풀(154)의 최신 버전에서 작업 증명 퍼즐을 해결하기 위해 경쟁하기 시작할 것이다(다른 채굴자들(104M)은 풀(154)의 구 뷰(old view)에 기초하여 퍼즐을 해결하고자 여전히 시도할 수 있지만, 누구든 먼저 도달한 사람이 다음 새로운 블록(151)이 끝나고 새로운 풀(154)이 시작되는 곳을 정의할 것이며, 결국 누군가가 앨리스의 트랜잭션(152j)을 포함하는 풀(154)의 부분에 대한 퍼즐을 해결할 것임). 새로운 트랜잭션(152j)을 포함하는 풀(154)에 대한 작업 증명이 완료되면, 이는 변경 불가능하게 블록체인(150)의 블록들(151) 중 하나의 부분이 된다. 각각의 트랜잭션(152)은 이전 트랜잭션에 대한 역 포인터를 포함하여서, 트랜잭션들의 순서가 또한 변경 불가능하게 기록된다.

[0040] **UTXO-기반 모델**

[0041] 도 2는 예시적인 트랜잭션 프로토콜을 예시한다. 이는 UTXO-기반 프로토콜의 예이다. 트랜잭션(152)(약칭 "Tx")은 블록체인(150)의 기본 데이터 구조이다(각각의 블록(151)은 하나 이상의 트랜잭션들(152)을 포함함). 다음은 출력-기반 또는 "UTXO" 기반 프로토콜을 참조하여 설명될 것이다. 그러나 이것은 모든 가능한 실시예들로 제한되지 않는다.

[0042] UTXO-기반 모델에서, 각각의 트랜잭션("Tx")(152)은 하나 이상의 입력들(202) 및 하나 이상의 출력들(203)을 포함하는 데이터 구조를 포함한다. 각각의 출력(203)은 (UTXO가 아직 리딤되지 않은 경우) 다른 새로운 트랜잭션의 입력(202)에 대한 소스로서 사용될 수 있는 미지출 트랜잭션 출력(UTXO)을 포함할 수 있다. UTXO는 디지털 자산(가치의 저장소)의 금액을 지정한다. 이는 또한 다른 정보 중에서, 그것이 발생한 트랜잭션의 트랜잭션 ID를 포함할 수 있다. 트랜잭션 데이터 구조는 또한 입력 필드(들)(202) 및 출력 필드(들)(203)의 크기의 표시자를 포함할 수 있는 헤더(201)를 포함할 수 있다. 헤더(201)는 또한 트랜잭션의 ID를 포함할 수 있다. 실시예들에서, 트랜잭션 ID는 (트랜잭션 ID 자체는 제외한) 트랜잭션 데이터의 해시이고 채굴자들(104M)에게 제출된 원시 트랜잭션(152)의 헤더(201)에 저장된다.

[0043] 도 2의 각각의 출력이 UTXO로서 도시되지만, 트랜잭션은 부가적으로 또는 대안적으로 하나 이상의 지출 불가능한 트랜잭션 출력들을 포함할 수 있다.

[0044] 앨리스(103a)가 해당 디지털 자산의 금액을 밥(103b)에게 전달하는 트랜잭션(152j)을 생성하기를 원한다고 하자. 도 2에서 앨리스의 새로운 트랜잭션(152j)은 "Tx_1"로서 라벨이 지정된다. 이는 시퀀스의 선행 트랜잭션(152i)의 출력(203)에서 앨리스에게 잠긴 디지털 자산의 금액을 취하고, 이 중 적어도 일부를 밥에게 전달한다. 선행 트랜잭션(152i)은 도 2에서 "Tx_0"로 라벨이 지정된다. Tx_0 및 Tx_1은 임의의 라벨일 뿐이다. 이들은, Tx_0이 블록체인(151)의 최초 트랜잭션이거나, Tx_1이 풀(154)에서 바로 다음 트랜잭션이라는 것을 반드시 의미하지는 않는다. Tx_1은 앨리스에게 잠긴 미지출 출력(203)을 여전히 갖는 임의의 선행(즉, 앞선) 트랜잭션을 뒤로 가리킬 수 있다.

[0045] 선행 트랜잭션 Tx_0은 앨리스가 자신의 새로운 트랜잭션 Tx_1을 생성할 때, 또는 적어도 그녀가 그것을 네트워크

(106)에 전송할 때까지 이미 유효성 검증되고 블록체인(150)에 포함되었을 수 있다. 이는 그 시간에 이미 블록들(151) 중 하나에 포함되었거나, 풀(154)에서 여전히 대기 중일 수 있으며, 이 경우에 곧 새로운 블록(151)에 포함될 것이다. 대안적으로 Tx_0 및 Tx_1이 생성되고 네트워크(102)에 함께 전송될 수 있거나, 또는 노드 프로토콜이 "고아(orphan)" 트랜잭션들을 버퍼링하도록 허용하는 경우 Tx_0는 Tx_1 이후에도 전송될 수 있다. 트랜잭션들의 시퀀스의 맥락에서 본원에서 사용된 바와 같은 "선행" 및 "후속"이라는 용어들은 (트랜잭션이 다른 트랜잭션을 뒤로 가리키고, 이와 같이 계속되는) 트랜잭션들에서 지정된 트랜잭션 포인터들에 의해 정의된 바와 같은 시퀀스에서의 트랜잭션들의 순서를 지칭한다. 이들은 "선행자(predecessor)" 및 "후행자(successor)", 또는 "앞선(antecedent)"과 "후위의(descendant)", "부모" 및 "자식" 등으로 동등하게 대체될 수 있다. 이는 그것들이 생성되고, 네트워크(106)로 전송되거나, 임의의 주어진 노드(104)에 도달하는 순서를 반드시 의미하지는 않는다. 그럼에도 불구하고, 선행 트랜잭션(앞선 트랜잭션 또는 "부모")을 가리키는 후속 트랜잭션(후위의 트랜잭션 또는 "자식")은 부모 트랜잭션이 유효성 검증될 때까지 그리고 유효성 검증되지 않는 한 유효성 검증되지 않을 것이다. 그의 부모 이전에 노드(104)에 도달하는 자식은 고아로 간주된다. 이는 노드 프로토콜 및/또는 채굴자 거동에 의존하여 부모를 기다리기 위해 특정 시간 동안 버퍼링되거나 폐기될 수 있다.

[0046] 선행 트랜잭션 Tx_0의 하나 이상의 출력들(203) 중 하나는, 본원에서 $UTXO_0$으로서 라벨이 지정되는 특정 UTXO를 포함한다. 각각의 UTXO는 UTXO에 의해 표현되는 디지털 자산의 금액을 지정하는 값 및 후속 트랜잭션이 유효성 검증되고 따라서 UTXO가 성공적으로 리딤되기 위하여 후속 트랜잭션의 입력(202)에서 잠금해제 스크립트에 의해 만족되어야 하는 조건을 정의하는 잠금 스크립트를 포함한다. 통상적으로, 잠금 스크립트는 특정 당사자(그것이 포함된 트랜잭션의 수혜자)에게로 금액을 잠근다. 즉, 잠금 스크립트는, 통상적으로 후속 트랜잭션의 입력의 잠금해제 스크립트가 선행 트랜잭션이 잠겨 있는 당사자의 암호화 서명을 포함하는 조건을 포함하는 잠금해제 조건을 정의한다.

[0047] 잠금 스크립트(일명 scriptPubKey)는 노드 프로토콜에 의해 인식되는 도메인 특정 언어로 작성된 코드 조각이다. 이러한 언어의 특정 예는 "스크립트(Script)"(대문자 S)라 불린다. 잠금 스크립트는 트랜잭션 출력(203)을 지출하는 데 어떤 정보가 필요한지, 예컨대, 앨리스의 서명 요건을 지정한다. 잠금해제 스크립트들은 트랜잭션들의 출력에서 나타난다. 잠금해제 스크립트(일명 scriptSig)는 잠금 스크립트 기준들을 충족시키는 데 필요한 정보를 제공하는 도메인 특정 언어로 작성된 코드 조각이다. 예컨대, 이는 밥의 서명을 포함할 수 있다. 잠금해제 스크립트들은 트랜잭션들의 입력(202)에 나타난다.

[0048] 따라서 예시된 예에서, Tx_0의 출력(203)의 $UTXO_0$은 $UTXO_0$가 리딤되기 위해(엄밀히, $UTXO_0$을 리딤하고자 시도하는 후속 트랜잭션이 유효하기 위해) 앨리스의 서명 Sig P_A를 요구하는 잠금 스크립트 [Checksig P_A]를 포함한다. [Checksig P_A]는 앨리스의 공개-개인 키 쌍으로부터의 공개 키 P_A를 포함한다. Tx_1의 입력(202)은 (예컨대, 실시예에서, 전체 트랜잭션 Tx_0의 해시인 그의 트랜잭션 ID인 $TxID_0$에 의해) Tx_1을 뒤로 가리키는 포인터를 포함한다. Tx_1의 입력(202)은 Tx_0의 임의의 다른 가능한 출력들 사이에서 그것을 식별하기 위해 Tx_0 내에서 $UTXO_0$을 식별하는 인덱스를 포함한다. Tx_1의 입력(202)은 앨리스의 암호화 서명을 포함하는 잠금해제 스크립트 <Sig P_A>를 더 포함하며, 이는 앨리스가 키 쌍으로부터 자신의 개인 키를 데이터의 미리 정의된 부분(때로는 암호화에서 "메시지"라 불림)에 적용함으로써 생성된다. 유효한 서명을 제공하기 위해 앨리스에 의해 서명될 필요가 있는 데이터(또는 "메시지")는 잠금 스크립트, 노드 프로토콜 또는 이들의 조합에 의해 정의될 수 있다.

[0049] 새로운 트랜잭션 Tx_1이 노드(104)에 도달할 때, 노드는 노드 프로토콜을 적용한다. 이는 잠금해제 스크립트가 잠금 스크립트에 정의된 조건(이 조건은 하나 이상의 기준들을 포함할 수 있음)을 충족시키는지를 체크하기 위해 잠금 스크립트 및 잠금해제 스크립트를 함께 실행하는 것을 포함한다. 실시예들에서, 이는 2개의 스크립트들을 연결시키는 것을 수반한다.

[0050] $$<\text{Sig } P_A><P_A> \,||\, [\text{Checksig } P_A]$$

[0051] 여기에서 "||"는 연결을 표현하고 "<...>"는 스택 상에 데이터를 배치하는 것을 의미하고, "[…]"는 잠금해제 스크립트(이 예에서, 스택-기반 언어)에 의해 구성된 함수이다. 동등하게, 스크립트들을 연결시키는 대신, 스크립트들은 공통 스택을 사용하여 차례로 실행될 수 있다. 어느 쪽이든, 함께 실행될 때, 스크립트들은 Tx_0의 출력의 잠금 스크립트에 포함된 바와 같은 앨리스의 공개 키 P_A를 사용하여, Tx_1의 입력의 잠금 스크립트가 데이

터의 예상되는 부분에 서명하는 앨리스의 서명을 포함한다는 것을 인증한다. 이 인증을 수행하기 위하여 데이터의 예상되는 부분 자체("메시지")가 또한 Tx_0에 포함될 필요가 있다. 실시예들에서, 서명된 데이터는 Tx_0 전체를 포함한다(이에 따라, 평문으로 데이터의 서명된 부분을 지정하는 별개의 요소가 포함될 필요가 있는데, 그 이유는 이것이 이미 본질적으로 존재하기 때문임).

[0052] 공개-개인 암호화에 의한 인증의 세부사항들은 당업자에게 친숙할 것이다. 기본적으로, 앨리스가 자신의 개인 키로 메시지를 암호화함으로써 그 메시지에 서명한 경우, 앨리스의 공개 키 및 평문의 일반 메시지(암호화되지 않은 메시지)를 감안하여, 노드(104)와 같은 다른 엔티티는 암호화된 버전의 메시지가 앨리스에 의해 서명된 것임이 틀림없다는 것을 인증할 수 있다. 서명은 통상적으로 메시지를 해시하는 것, 해시에 서명하는 것, 그리고 이를 서명으로서 메시지의 평문 버전에 태깅하고, 이에 따라 공개 키의 임의의 보유자(holder)가 서명을 인증하는 것을 가능하게 하는 것을 포함한다.

[0053] Tx_1의 잠금해제 스크립트가 Tx_0의 잠금 스크립트에 지정된 하나 이상의 조건들을 충족시키는 경우(이에 따라, 보여진 예에서, 앨리스의 서명이 Tx_1에서 제공되고 인증된 경우), 노드(104)는 Tx_1이 유효한 것으로 간주한다. 그것이 채굴 노드(104M)인 경우, 이는 그것이 작업 증명을 기다리는 트랜잭션들의 풀(154)에 추가될 것임을 의미한다. 그것이 포워딩 노드(104F)인 경우, 트랜잭션 Tx_1을 네트워크(106)의 하나 이상의 다른 노드들(104)로 전달하여서, 그 트랜잭션이 네트워크에 걸쳐 전파될 것이다. Tx_1이 유효성 검증되고 블록체인(150)에 포함되면, 이는 지출된 것으로 Tx_0으로부터 $UTXO_0$를 정의한다. Tx_1은 그것이 미지출 트랜잭션 출력(203)을 지출하는 경우에만 유효할 수 있다는 것에 주의한다. 다른 트랜잭션(152)에 의해 이미 지출된 출력을 지출하려고 시도하는 경우, 다른 모든 조건들이 충족되는 경우 조차도 Tx_1은 유효하지 않을 것이다. 따라서 노드(104)는 또한 선행 트랜잭션 Tx_0에서 참조된 UTXO가 이미 지출되었는지(다른 유효한 트랜잭션에 대한 유효한 입력을 이미 형성했는지)를 체크할 필요가 있다. 이는 트랜잭션들(152) 상에 정의된 순서를 부과하는 것이 블록체인(150)에 대해 중요한 하나의 이유이다. 실제로 주어진 노드(104)는 트랜잭션들(152)이 지출된 UTXO들(203)을 마킹하는 별개의 데이터베이스를 유지할 수 있지만, 궁극적으로 UTXO가 지출되었는지를 정의하는 것은 블록체인(150)의 다른 유효한 트랜잭션에 대한 유효한 입력이 이미 형성되었는지의 여부이다.

[0054] UTXO-기반 트랜잭션 모델에서, 주어진 UTXO는 전체로서 지출될 필요가 있다는 것에 주의한다. 다른 프랙션(fraction)이 지출되면서, 지출된 것으로 UTXO에서 정의된 금액의 프랙션이 "남겨둘" 수는 없다. 그러나 UTXO로부터의 금액은 다음 트랜잭션의 다수의 출력들 사이에서 분할될 수 있다. 예컨대, Tx_0의 $UTXO_0$에 정의된 금액은 Tx_1의 다수의 UTXO들 사이에서 분할될 수 있다. 따라서 앨리스가 $UTXO_0$에 정의된 모든 금액을 밥에게 주기를 원하지 않는 경우, 앨리스는 Tx_1의 제2 출력에서 자신에게 잔돈을 주거나, 다른 당사자에게 지불하는데 나머지를 사용할 수 있다.

[0055] 실제로 앨리스는 또한 일반적으로 승리한 채굴자에 대한 수수료를 포함할 필요가 있을 것인데, 그 이유는 최근에는 생성 트랜잭션의 보상만으로는 일반적으로 채굴에 동기를 부여하는데 충분하지 않기 때문이다. 앨리스가 채굴자에 대한 수수료를 포함하지 않는 경우, Tx_0은 채굴자 노드들(104M)에 의해 거부될 가능성이 높을 것이고, 이에 따라 기술적으로 유효하더라도, 그것은 여전히 전파되어 블록체인(150)에 포함되지 않을 것이다(채굴자 프로토콜은 채굴자들(104M)이 원하지 않는 경우 이들에게 트랜잭션들(152)을 수락하도록 강요하지 않음). 일부 프로토콜들에서, 채굴 수수료는 자체의 별개의 출력(203)을 요구하지 않는다(즉, 별개의 UTXO가 필요하지 않음). 대신 주어진 트랜잭션(152)의 입력(들)(202)에 의해 가리켜지는 총 금액과 출력(들)(203)에 지정된 총 금액 사이의 임의의 차이가 승리한 채굴자(104)에게 자동으로 주어진다. 예컨대, $UTXO_0$에 대한 포인터가 Tx_1에 대한 유일한 입력이고 Tx_1는 단 하나의 출력 $UTXO_1$만을 갖는다고 하자. $UTXO_0$에 지정된 디지털 자산의 금액이 $UTXO_1$에 지정된 금액보다 큰 경우, 차이는 승리한 채굴자(104M)에게 자동으로 넘어간다. 그러나 대안적으로 또는 부가적으로, 채굴자 수수료가 트랜잭션(152)의 UTXO들(203) 중 자체 UTXO에서 명시적으로 지정될 수 있다는 것이 반드시 배제되는 것은 아니다.

[0056] 또한, 주어진 트랜잭션(152)의 모든 출력들(203)에서 지정된 총 금액이 모든 그의 입력들(202)에 의해 가리켜지는 총 금액보다 큰 경우, 이는 대부분의 트랜잭션 모델들에서 무효에 대한 다른 근거란 것에 주의한다. 따라서, 이러한 트랜잭션들은 블록들(151) 내로 채굴되거나 전파되지 않을 것이다.

[0057] 앨리스 및 밥의 디지털 자산들은 블록체인(150)의 임의의 위치의 임의의 트랜잭션들(152)에서 그들에게 잠겨 있는 미지출 UTXO로 구성된다. 따라서 통상적으로, 주어진 당사자(103)의 자산들은 블록체인(150) 전반에 걸친 다양한 트랜잭션들(152)의 UTXO들에 걸쳐 흩어져 있다. 블록체인(150)의 어떤 위치에도 주어진 당사자(103)의 총 잔고를 정의하는 숫자는 전혀 없다. 클라이언트 애플리케이션(105)에서 지갑 기능의 역할은, 개개의 당사자에게 잠겨 있으며 다른 전방 트랜잭션에서 아직 지출되지 않은 모든 다양한 UTXO들의 값들을 함께 대조하는 것이다. 이는 저장 노드들(104S) 중 임의의 것, 예컨대, 개개의 당사자의 컴퓨터 장비(102)에 가장 가깝거나 가장 잘 연결된 저장 노드(104S)에 저장된 바와 같은 블록체인(150)의 사본을 질의함으로써 가능할 수 있다.

[0058] 스크립트 코드는 종종 도식적으로(즉, 정확한 언어가 아님) 표현된다는 것에 주의한다. 예컨대, [Checksig P_A] = OP_DUP OP_HASH160 <H(Pa)> OP_EQUALVERIFY OP_CHECKSIG를 의미하도록 [Checksig P_A]가 작성될 수 있다. "OP_..."는 스크립트 언어의 특정 작업코드를 지칭한다. OP_CHECKSIG(또한 "Checksig"라 불림)는 2개의 입력들(서명 및 공개 키)을 취하고 ECDSA(Elliptic Curve Digital Signature Algorithm)을 사용하여 서명의 유효함을 검증하는 스크립트 작업코드이다. 런타임 시에, 서명('sig')의 임의의 발생은 스크립트로부터 제거되지만, 해시 퍼즐과 같은 부가적인 요건들은 'sig' 입력으로 검증된 트랜잭션에서 유지된다. 다른 예로서, OP_RETURN은 트랜잭션 내에 메타데이터를 저장하고 그리하여 메타데이터를 블록체인(150)에 변경 불가능하게 기록할 수 있는 트랜잭션의 지출 불가능한 출력을 생성하기 위한 스크립트 언어의 작업코드이다. 예컨대, 메타데이터는 블록체인에 저장하고자 하는 문서를 포함할 수 있다.

[0059] 서명 P_A는 디지털 서명이다. 실시예들에서, 이는 타원 곡선 secp256k1을 사용하는 ECDSA에 기초한다. 디지털 서명은 특정 데이터 조각에 서명한다. 실시예들에서, 주어진 트랜잭션에 대해, 서명은 트랜잭션 입력의 일부, 및 트랜잭션 출력의 전부 또는 일부에 서명할 것이다. 서명되는 출력들의 특정 부분들은 SIGHASH 플래그에 의존한다. SIGHASH 플래그는 어느 출력들이 서명되는지를 선택하기 위해 서명의 끝에 포함된 4-바이트 코드이다(이에 따라, 서명 시에 고정됨).

[0060] 잠금 스크립트는 때로는, 그것이 개개의 트랜잭션이 잠겨 있는 당사자의 공개 키를 포함한다는 사실을 지칭하는 "scriptPubKey"라 칭해진다. 잠금해제 스크립트는 때로는 그것이 대응하는 서명을 제공한다는 사실을 지칭하는 "scriptSig"라 칭해진다. 그러나, 보다 일반적으로, UTXO가 리딤되기 위한 조건이 서명을 인증하는 것을 포함하는 것이 블록체인(150)의 모든 애플리케이션들에서 필수적인 것은 아니다. 보다 일반적으로 스크립팅 언어는 임의의 하나 이상의 조건들을 정의하는 데 사용될 수 있다. 따라서 보다 일반적인 용어들 "잠금 스크립트" 및 "잠금해제 스크립트"가 선호될 수 있다.

[0061] **선택적 사이드 채널**

[0062] 도 3은 블록체인(150)을 구현하기 위한 추가 시스템(100)을 도시한다. 시스템(100)은 부가적인 통신 기능성이 수반된다는 점을 제외하고는 도 1과 관련하여 설명된 것과 실질적으로 동일하다. 앨리스 및 밥의 컴퓨터 장비(102a, 120b) 각각 상의 클라이언트 애플리케이션은 각각 부가적인 통신 기능성을 포함한다. 즉, 이는 (어느 한 당사자 또는 제3자의 주도로) 앨리스(103a)가 밥(103b)과 별개의 사이드 채널(301)을 설정하는 것을 가능하게 한다. 사이드 채널(301)은 P2P 네트워크와 별개로 데이터 교환을 가능하게 한다. 이러한 통신을 때로는 "오프-체인(off-chain)"으로서 지칭된다. 예컨대, 이는 당사자들 중 하나가 네트워크(106)로 브로드캐스팅하기로 선택할 때까지, (아직) 트랜잭션이 네트워크 P2P(106) 상에 공개되거나 체인(150)으로 진행됨 없이 앨리스와 밥 사이에서 트랜잭션(152)을 교환하는 데 사용될 수 있다. 대안적으로 또는 부가적으로, 사이드 채널(301)은 키들, 협상된 금액들 또는 조건들(terms), 데이터 콘텐츠 등과 같은 임의의 다른 트랜잭션 관련 데이터를 교환하는 데 사용될 수 있다.

[0063] 사이드 채널(301)은 P2P 오버레이 네트워크(106)와 동일한 패킷 교환 네트워크(101)를 통해 설정될 수 있다. 대안적으로 또는 부가적으로, 사이드 채널(301)은 상이한 네트워크 이를테면, 모바일 셀룰러 네트워크, 또는 로컬 영역 네트워크 이를테면, 로컬 무선 네트워크, 또는 심지어, 앨리스 및 밥의 디바이스들(1021, 102b) 사이의 직접 유선 또는 무선 링크를 통해 설정될 수 있다. 일반적으로, 본원의 임의의 위치에서 지칭되는 바와 같은 사이드 채널(301)은 "오프-체인", 즉 P2P 오버레이 네트워크(106)와 별개로 데이터를 교환하기 위한 하나 이상의 네트워킹 기술들 또는 통신 매체들을 통한 임의의 하나 이상의 링크들을 포함할 수 있다. 하나 초과의 링크가 사용되는 경우, 오프-체인 링크들의 번들(bundle) 또는 모음은 전체적으로 사이드 채널(301)로서 지칭될 수 있다. 따라서 앨리스 및 밥이 사이드 채널(301)을 통해 특정 정보 조각들 또는 데이터 등을 교환한다고 하면, 이는 이러한 모든 데이터 조각들이 정확히 동일한 링크 또는 심지어 동일한 유형의 네트워크를 통해 전송되어야

한다는 것을 반드시 의미하는 것은 아니란 것에 주의한다.

[0064] **노드 소프트웨어**

[0065] 도 4는 UTXO-기반 또는 출력-기반 모델의 예에서 P2P 네트워크(106)의 각각의 노드(104) 상에서 실행되는 노드 소프트웨어(400)의 예를 예시한다. 노드 소프트웨어(400)는 프로토콜 엔진(401), 스크립트 엔진(402), 스택(403), 애플리케이션-레벨 판단 엔진(404), 및 일 세트의 하나 이상의 블록체인-관련 기능 모듈들(405)을 포함한다. 임의의 주어진 노드(104)에서, 이들은 채굴 모듈(405M), 포워딩 모듈(405F) 및 저장 모듈(405S) 중 (노드의 역할 또는 역할들에 의존하여) 임의의 하나, 둘 또는 3개 전부를 포함할 수 있다. 프로토콜 엔진(401)은 트랜잭션(152)의 상이한 필드들을 인식하고 이들을 노드 프로토콜에 따라 프로세싱하도록 구성된다. 다른 선행 트랜잭션(152m-1)(Tx_{m-1})의 출력(예컨대, UTXO)을 가리키는 입력을 갖는 트랜잭션(152m)(Tx_m)이 수신될 때, 프로토콜 엔진(401)은 Tx_m의 잠금해제 스크립트를 식별하고 이를 스크립트 엔진(402)에 전달한다. 프로토콜 엔진(401)은 또한 Tx_m의 입력의 포인터에 기초하여 Tx_{m-1}를 식별 및 리트리브(retrieve)한다. 이는 Tx_{m-1}가 아직 블록체인(150)에 있지 않은 경우 보류 중인 트랜잭션의 개개의 노드의 자체 풀(154)로부터, 또는 Tx_{m-1}이 이미 블록체인(150) 상에 있는 경우 개개의 노드 또는 다른 노드(104)에 저장된 블록체인(150)의 블록(151)의 사본으로부터 Tx_{m-1}를 리트리브할 수 있다. 어느 쪽이든, 스크립트 엔진(401)은 Tx_{m-1}의 가리켜진 출력에서 잠금 스크립트를 식별하고 이를 스크립트 엔진(402)으로 전달한다.

[0066] 따라서, 스크립트 엔진(402)은 Tx_m의 대응하는 입력으로부터의 잠금해제 스크립트 및 Tx_{m-1}의 잠금 스크립트를 갖는다. 예컨대, Tx_1 및 Tx_2가 도 4에 예시되지만, Tx_0 및 Tx_1 등과 같은 트랜잭션들의 임의의 쌍에 동일하게 적용될 수 있다. 스크립트 엔진(402)은 이전에 논의된 바와 같이 2개의 스크립트들을 함께 실행하며, 이는 사용되고 있는 스택-기반 스크립팅 언어(예컨대, 스크립트(Script))에 따라 스택(403) 상에 데이터를 배치하고 스택(403)으로부터 데이터를 리트리브하는 것을 포함할 것이다.

[0067] 스크립트들을 함께 실행함으로써, 스크립트 엔진(402)은 잠금해제 스크립트가 잠금 스크립트에 정의된 하나 이상의 기준들을 충족시키는지 — 즉, 잠금 스크립트가 포함되는 출력을 "잠금해제"하는가? 를 결정한다. 스크립트 엔진(402)은 이 결정의 결과를 프로토콜 엔진(401)에 반환한다. 잠금해제 스크립트가 대응하는 잠금 스크립트에 지정된 하나 이상의 기준들을 충족시키는 것으로 스크립트 엔진(402)이 결정하는 경우, 결과 "참"이 반환된다. 그렇지 않으면, 결과 "거짓"이 반환된다.

[0068] 출력-기반 모델에서, 스크립트 엔진(402)으로부터의 결과 "참"은 트랜잭션의 유효함을 위한 조건들 중 하나이다. 통상적으로 또한 충족되어야 하는, 프로토콜 엔진(401)에 의해 평가되는 하나 이상의 추가의 프로토콜-레벨 조건들; 이를테면, Tx_m의 출력(들)에 지정된 디지털 자산의 총 금액이 입력(들)에 의해 가리켜지는 총 금액을 초과하지 않는 것, 그리고 Tx_{m-1}의 가리켜진 출력이 다른 유효한 트랜잭션에 의해 이미 지출되지 않았을 것이 존재한다. 프로토콜 엔진(401)은 하나 이상의 프로토콜-레벨 조건들과 함께 스크립트 엔진(402)으로부터의 결과를 평가하고, 이들이 모두 참인 경우에만, 트랜잭션 Tx_m을 유효성 검증한다. 프로토콜 엔진(401)은 트랜잭션이 유효한지에 관한 표시를 애플리케이션-레벨 판단 엔진(404)에 출력한다. Tx_m이 실제로 유효성 검증된다는 조건에서만, 판단 엔진(404)은 Tx_m에 대해 각자의 블록체인-관련 기능을 수행하도록 채굴 모듈(405M) 및 포워딩 모듈(405F) 중 하나 또는 둘 모두를 제어하기로 선택할 수 있다. 이는 채굴 모듈(405M)이 블록(151)으로의 채굴을 위해 노드의 개개의 풀(154)에 Tx_m을 추가하는 것 및/또는 포워딩 모듈(405F)이 P2P 네트워크(106)의 다른 노드(104)에 Tx_m을 포워딩하는 것을 포함할 수 있다. 그러나 실시예들에서, 판단 엔진(404)이 무효 트랜잭션을 포워딩하거나 채굴하기로 선택하지 않을 것임에도 불구하고, 이는, 역으로 단순히 트랜잭션이 유효하기 때문에 이 유효한 트랜잭션의 채굴 또는 포워딩이 어쩔 수 없이 트리거(trigger)하게 되어야 한다는 것을 반드시 의미하지는 않는다는 것에 주의한다. 선택적으로, 실시예들에서, 판단 엔진(404)은 기능들 중 어느 하나 또는 둘 모두를 트리거하기 전에 하나 이상의 부가적인 조건들을 적용할 수 있다. 예컨대, 노드가 채굴 노드(104M)인 경우, 판단 엔진은 트랜잭션이 유효하고 채굴 수수료가 충분히 남아 있다는 것을 조건으로만 트랜잭션을 채굴하기로 선택할 수 있다.

[0069] 또한, 본원에서 "참" 및 "거짓"이라는 용어들은 단지 단일 이진 숫자(비트)의 형태로 표현되는 결과를 반환하는

것으로 반드시 제한되지는 않지만, 이는 확실히 하나의 가능한 구현이라는 것에 주의한다. 보다 일반적으로, "참"은 성공 또는 긍정적인 결과를 표시하는 임의의 상태를 지칭할 수 있고 "거짓"은 실패 또는 비-긍정적인 결과를 표시하는 임의의 상태를 지칭할 수 있다. 예컨대, 계정-기반 모델(도 4에 예시되지 않음)에서, "참"의 결과는 노드(104)에 의한 서명의 암시적인 프로토콜 레벨의 유효성 검증 및 스마트 계약의 부가적인 긍정적인 출력의 조합에 의해 표시될 수 있다(개별 결과들 둘 모두가 참인 경우, 전체 결과가 참을 시그널링하는 것으로 간주됨).

[0070] **사물 인터넷(INTERNET OF THINGS)**

[0071] IoT는 일상의 물리적 디바이스들 및 오브젝트들로의 인터넷의 확장이다. 컴퓨테이셔널 프로세싱 파워 및 인터넷 연결이 매립되면, 디바이스들은 서로 통신하고 상호작용할 수 있으며 원격으로 모니터링 및 제어될 수 있다. 무선 센서 네트워크들 및/또는 제어 시스템들을 지원할 수 있는 디바이스들의 시스템들이 IoT를 가능하게 할 가능성이 높다는 것이 일반적으로 받아들여지고 있지만, 시간이 지남에 따라 IoT의 정의는 기계 학습, 실시간 분석 및 다수의 기술들의 융합으로 인해 진화했다.

[0072] IoT 시스템들은 여러 난제에 직면해 있다. 예컨대, 그러한 시스템들의 확장성 및 비용은 IoT 시스템들이 그의 잠재력을 최대한 발휘하는 것을 방해할 수 있다. 중앙화된 방식으로 연결 및 제어될 때 IoT 디바이스들은 데이터를 송신하고 제어 커맨드들을 수신하기 위한 백-엔드 인프라스트럭처(back-end infrastructure)들을 요구한다. 이러한 백-엔드 인프라스트럭처들은 제3자 클라우드 서비스들 또는 온-프레미스 서버 팜(on-premise server farm)들 상에 호스팅된다. 그 후, IoT 솔루션들의 확장성은 백-엔드 서버들 및 데이터 센터들의 확장성에 의해 결정되며, IoT 서비스 제공자들의 엄청나게 높은 운영 비용들이 들 수 있다. 결과적으로 다수의 제안된 IoT 솔루션들은 비용 효율적이지 않고 일상적인 시나리오들에서 사용하기에 적합하지 않다. 네트워크 레이턴시와 같은 성능 측정들이 또한 IoT 채택률을 결정하는 중요한 요인이 될 것이다.

[0073] IoT 시스템들이 직면한 다른 난제는 자동화와 제어 간의 절충이다. IoT 솔루션들은 일상적인 전자 디바이스들에 대한 원격 액세스 및 제어를 가능하게 하도록 설계된다. 대부분의 IoT 솔루션들은 디바이스들과 다른 IoT 솔루션 구성요소들 간의 완전한 사용자 제어와 자동화된 통신 사이의 균형을 유지한다. 디바이스 또는 IoT 시스템이 오작동하는 경우에, 오버라이드 메커니즘과 같은 안전 조치들 설치될 필요가 있다.

[0074] 다른 난제는 사이버 공격들로부터의 위협이다. 인터넷을 통해 디바이스들의 자동화된 제어를 가능하게 함으로써, 사용자들은 2개의 형태로 나타나는 잠재적인 보안 위험들에 이들을 노출시키는데, 하나는 인터넷을 통해 IoT 디바이스 메타데이터를 송신함으로써 초래되는 프라이버시 위험이다. 예컨대, 도청자들이 가전 제품들과 같은 디바이스들로부터의 데이터에 대한 액세스를 획득하는 경우, 집에 사람이 있을 때를 예측하기 위해 디바이스 사용 패턴이 범죄자들 예컨대, 도둑들에 의해 사용될 수 있다. 제2 위험은 공격자들 또는 다른 제3자들이 IoT 디바이스들의 제어를 획득할 가능성이다. 이를테면, 중장비 또는 위험물들을 동작시키는 데 사용되는 성능 중요 제어 소프트웨어에 대해, 공격은 치명적인 결과들을 가질 수 있다.

[0075] IoT 시스템들은 중앙화 또는 탈중앙화 및/또는 하이브리드로 설계될 수 있다. 중앙화된 솔루션들은 병목 현상들에 시달리지만, IoT 시스템의 특권 구성요소에 의해 더 빠르고 더 신뢰할 수 있는 제어를 가능하게 한다. 상태 업데이트들의 탈중앙화된 보고는 IoT 솔루션들이 보다 확장 가능하게 되는 것을 가능하게 한다. 에지 컴퓨팅(edge computing)은 중요한 애플리케이션들의 네트워크 레이턴시를 감소시키고 클라우드에 대한 IoT 시스템들의 의존도를 낮추며 대량의 IoT 데이터의 더 양호한 관리를 제공하는데 도움이 될 수 있다. 탈중앙화된 프로세싱의 부상은 중앙화된 및 분산된 아키텍처들의 이점들을 더 잘 활용하는 시스템 아키텍처의 기회들을 강조한다. 계층적 제어 구조들 내에서 중앙화된 및 분산된 시스템을 결합하는 하이브리드 시스템은 사용자 안전 및 유연성 목적들을 향상시킬 수 있다.

[0076] 블록체인 기술은, 블록체인은 하나의 네트워크로의 지불 및 제어의 통합을 가능하게 하고; 기존 인프라스트럭처가 디바이스-상태 변화들에 관한 메시지들을 피기백(piggyback)하는데 사용될 수 있고; 그리고 네트워크 상의 데이터의 탈중앙화된 제어가 더 빠른 사용자-디바이스 상호작용을 가능하게 한다는 이유로, IoT의 미래에서 선도적인 역할을 할 잠재력을 갖는다. 블록체인 기술과 결합하여, 물리적 세계에서 역할들을 수행하는 기존 IoT 디바이스들은 동시에 값을 메시징 및 교환할 수 있을 것이다. 공개 블록체인들은 자신의 프로토콜 내에 강력한 암호화 보안이 내장된 – 이는 IoT와 연관된 여러 위험들을 자동으로 해결함 – 범용 상품 원장뿐만 아니라 글로벌 지불 네트워크로서 역할을 한다.

[0077] 도 5는 본 개시의 실시예들을 구현하기 위한 예시적인 시스템(500)을 예시한다. 예시적인 시스템(500)은 하나

이상의 종단 디바이스들(즉, 컴퓨팅 디바이스들)(502)의 제1 네트워크(501) 및 하나 이상의 브리징 노드들(503)(즉, 블록체인 클라이언트 애플리케이션을 실행하고 이에 따라 블록체인 네트워크(106)와 제1 네트워크(501) 사이에서 브리지로서 작용하는 컴퓨팅 디바이스)를 포함한다. 명료함을 위해, 제1 네트워크(501)는 IoT 네트워크, 즉 인터넷에 의해 상호 연결된 컴퓨팅 디바이스들의 네트워크로서 지칭될 것이다. 통상적으로 종단 디바이스들(502) 및 브리징 노드들(503)은 일상적인 디바이스들에 임베딩된다. 종단 디바이스(502)는 다양한 형태들 예컨대, 사용자 디바이스들(예컨대, 스마트 TV들, 스마트 스피커들, 장난감들, 웨어러블들 등), 스마트 기기들(예컨대, 냉장고들, 세탁기들, 오븐들 등), 계량기들 또는 센서들(예컨대, 스마트 온도 조절기들, 스마트 조명, 보안 센서들 등) 중 하나의 형태를 취할 수 있다. 유사하게, 브리징 노드(503)는 또한 종단 디바이스가 취할 수 있는 것과 동일한 형태들을 포함(그러나 이에 제한되지 않음)할 수 있는 다양한 형태들을 취할 수 있다. 노드(503)는 또한 전용 서버 장비, 기지국, 액세스 포인트, 라우터 등의 형태를 취할 수 있다. 일부 예들에서, 각각의 디바이스는 고정 네트워크(예컨대, IP) 주소를 가질 수 있다. 예컨대, 종단 디바이스들 중 하나, 일부 또는 전부는 모바일 디바이스와 대조적으로, 정적 디바이스(예컨대, 스마트 조명 또는 스마트 중앙 난방 제어기 등)일 수 있다.

[0078] IoT 네트워크는 패킷-교환 네트워크(101), 통상적으로 인터넷과 같은 광역 인터네트워크를 포함한다. 패킷-교환 네트워크(101)의 노드들(503) 및 디바이스들(502)은 패킷-교환 네트워크(101) 내에서 P2P(peer-to-peer) 오버레이 네트워크(501)를 형성하도록 배열된 복수의 노드들(104)을 포함한다. 각각의 노드(503)는 각각이 하나 이상의 프로세서들, 예컨대, 하나 이상의 CPU(central processing unit)들, 가속기 프로세서들, 애플리케이션 특정 프로세서 및/또는 FPGA(field programmable gate array)들을 포함하는 개개의 프로세싱 장치를 포함하는 개개의 컴퓨터 장비를 포함한다. 각각의 노드(503)는 또한 메모리, 즉 비-일시적 컴퓨터-판독 가능 매체 또는 매체들의 형태의 컴퓨터-판독 가능 저장소를 포함한다. 메모리는 하나 이상의 메모리 매체들, 예컨대, 하드 디스크와 같은 자기 매체; 솔리드 스테이트 드라이브(SSD), 플래시 메모리 또는 EEPROM과 같은 전자 매체; 및/또는 광학 디스크 드라이브와 같은 광학 매체를 사용하는 하나 이상의 메모리 유닛들을 포함할 수 있다.

[0079] IoT 네트워크의 각각의 노드(503)는 또한 블록체인 노드(104)이다. 이러한 노드들(503)은 제1 네트워크(501)와 블록체인 네트워크(106) 사이의 브리지(게이트웨이)로서 작용하는 브리징 노드들(게이트웨이 노드들)로서 배열된다. 블록체인 노드(104)는 "리스닝 노드"일 수 있다. 리스닝 노드는 블록체인의 전체 사본을 유지하고 새로운 트랜잭션 및 블록을 유효성 검증 및 전파하지만, 능동적으로 새로운 블록들을 채굴하거나 생성하지 않는 클라이언트 애플리케이션을 실행한다. 대안적으로, 노드는 "SPV(simplified payment verification) 노드"일 수 있다. SPV 노드는 비트코인 트랜잭션들을 생성 및 브로드캐스트하고 주소들을 간접적으로 모니터링할 수 있지만, 블록체인의 전체 사본을 유지하지 않는 경량 클라이언트를 실행한다.

[0080] IoT 네트워크의 각각의 노드(503)는 직접 또는 간접적으로 종단 디바이스(502)를 제어하도록 구성된다. 종단 디바이스(502)에 직접 연결되는 노드(503)는 그 디바이스를 직접 제어할 수 있다. 종단 디바이스(502)에 직접 연결되지 않은 노드(503)는 예컨대, 하나 이상의 중개 노드들을 통해 종단 노드에 제어 메시지를 포워딩함으로써 그 디바이스를 간접적으로만 제어할 수 있다. 각각의 노드(503)는 하나 이상의 채굴 노드들(104M)에 연결된다.

[0081] 도 5는 또한 블록체인 네트워크(106)의 서브세트인 채굴 노드들(104M)의 네트워크(504)를 예시한다. 채굴 노드들은 도 1 내지 도 3을 참조하여 위에서 논의되었다. 채굴 노드들(104M)은 유효한 트랜잭션(예컨대, IoT 노드들로부터 송신된 트랜잭션들)을 블록체인(150)으로 채굴하도록 구성된다.

[0082] 도 5에 도시된 바와 같이, 노드들(503)은 P2P 네트워크(501) 및 블록체인 P2P 네트워크(106) 둘 모두의 부분을 형성하는 반면, 채굴 노드들(104M)은 블록체인 P2P 네트워크(106)만의 부분을 형성한다. 종단 디바이스들(502)이 P2P IoT 네트워크(501)만의 부분을 형성하는 것으로 도 5에 도시되지만, 종단 디바이스들(502)이 또한 블록체인 노드들(104)일 수 있다는 것은 배제되지 않는다.

[0083] 도 6은 예시적인 IoT 네트워크(501) 토폴로지를 예시한다. IoT 네트워크(501)는 마스터 노드(503a), 하나 이상의 중개 노드들(503b, 503c)의 하나 이상의 세트들(601), 및 종단 디바이스들(502)의 세트를 제어할 수 있다. 마스터 노드(502a)는 하나 이상의 중개 노드들(503b, 503c)을 제어하도록 구성된다. IoT 네트워크(501)가 중개 노드들의 다수의 세트들(예컨대, 층들(601a, 601b)을 포함하는 경우, 마스터 노드(503a)는 중개 노드들의 제1 세트(층)(601a)("서버 노드들"(503b))를 직접 제어하고 중개 노드들의 하나 이상의 추가 세트들(층들)(601b)(예컨대, "슬레이브 노드들"(503c)의 층)을 간접적으로 제어하도록 구성된다. 마스터 노드(503a)는 서버 및 슬레이브 노드들을 무효화 및 제어하는 능력을 갖는 제어 노드이다. 각각의 서버 노드(503b)는 슬레이브 노드들

(503c)을 제어하는 능력을 갖는 노드이다. 각각의 슬레이브 노드(503c)는 서버 노드들(503b) 및 마스터 노드(503a)의 제어 하에 있는 노드이다. 예로서, 종단 디바이스(502a)에 지시하기 위해, 마스터 노드(503a)는 서번트(servant) 노드(503b)를 통해 슬레이브 노드(503c)에 커맨드를 발행할 것이다.

[0084] 도 6의 예시적인 IoT 네트워크는 단 2개의 층들의 중개 노드들(서버 노드들 및 슬레이브 노드)만을 도시하지만, 다른 예들은 다른 예들은 예컨대, 마스터 노드(503a)와 서버 노드들(503b) 사이, 그리고/또는 서버 노드들(503b)과 슬레이브 노드(503c) 사이에 중개 노드들의 하나 이상의 추가 세트들을 포함할 수 있다. 도시된 바와 같이, 각각의 노드는 개개의 연결(602)을 통해 하나 이상의 다른 노드들에 연결되고, 각각의 종단 디바이스(502)는 개개의 연결(602)을 통해 하나 이상의 슬레이브 노드들에 연결된다. 하나 이상의 노드들(예컨대, 마스터 노드)은 아래에서 제어 노드들로서 지칭된다. 각각의 제어 노드는 커맨드들의 발행을 통해 동작을 수행하도록 다른 노드들에 지시할 수 있는 노드(503)이다.

[0085] IoT 네트워크 노드들(503)은 기능성의 범위, 명령들/특권들의 우월성, 및/또는 액세스들의 스팬(span)에서의 계층들에 대응할 수 있다. 일부 구현들에서, SPV 노드들의 계층적 세트는 도 5 및 도 6의 마스터(503a), 서버(503b) 및 슬레이브 노드들(503c)에 대응하는 3개 레벨의 계층을 갖는 "IoT 제어기"를 구현한다. 마스터 노드(503a)는 하나 이상의 서버 노드들(503b)에 지시하고, 각각의 서버 노드는 하나 이상의 슬레이브 노드들(503c)에 지시한다. 각각의 슬레이브 노드(503c)는 하나 이상의 서버 노드들(503b)로부터 명령들을 수신한다. 모든 슬레이브 노드(503c)는 하나 이상의 IoT 종단 디바이스들(502)과 통신하고, 이들은 IoT 제어기(503)와 IoT 종단 디바이스들(502) 사이의 직접 통신 채널들이다. IoT 제어기(503)의 실행의 상태들은 블록체인 트랜잭션들 Tx에 기록된다. 각각의 IoT 노드 – 마스터, 서버 또는 슬레이브 – 는 대응하는 트랜잭션들 Tx를 생성하여 블록체인 네트워크(106)에 브로드캐스트할 수 있다. 각각의 슬레이브 노드는 종단 디바이스들(502)로부터의 트리거 및/또는 확인 신호들을 모니터링하고, 모든 각각의 IoT 노드(503)는 IoT 제어기의 전체 로직을 실행할 목적으로 임의의 다른 IoT 노드와 상호작용하는 능력을 갖는다.

[0086] 마스터 노드, 서버 노드(들) 및 슬레이브 노드(들)는 각각 독립적으로 블록체인 네트워크(106) 상의 노드들(104)에 연결되고, (예컨대, 블록체인 주소들을 주시하기 위해) 블록체인 지갑을 동작시키고, 어쩌면, 전체 노드를 실행할 수 있다(그러나 이것이 요구되진 않음). 마스터 노드(503a)는 그의 제어 하에 다른 IoT 노드들의 활동을 직접 및 간접적으로 모니터링하고 블록체인 트랜잭션들 Tx의 형태로 이러한 노드들에 커맨드들을 발행하고 경고들에 응답하도록 구성된다. 서버 노드(503b)는 서버 노드(503b)에 의해 직접 제어되지 않는 주소들을 포함하는 다수의 주소들을 주시하도록 구성된다. 서버 노드들(503b)은 마스터 노드(503a)에 의해 동작들을 수행하도록 명령될 수 있다. 슬레이브 노드(503c)는 그의 제어 하에 있는 종단 디바이스(502)의 활동들을 직접적으로 모니터링하도록 구성된다. 슬레이브 노드들(503c)은 서버 노드들(503b)의 직접적인 커맨드 하에 있고 또한 마스터 노드(503a)에 의해 동작들을 수행하도록 명령될 수 있다. 슬레이브 노드들(503c)은 종단 디바이스들(502)에 대한 게이트웨이 노드들(즉, 종단 디바이스와 블록체인 네트워크(106) 사이의 게이트웨이)로서 작용한다. 종단 디바이스(502)는 근처의 슬레이브 디바이스들에 연결하도록 구성된다. 이들은 오프-체인 메시징 프로토콜을 사용하여 종단 디바이스 상태를 보고한다.

[0087] 종단 디바이스(502)가 IoT 노드들(503)에 의해 제어되지만 IoT 노드들(503)을 그들 자신이 제어하지 않는다는 점에서 IoT 노드(503)와 종단 디바이스(502) 사이에 구별이 이루어지지만, 종단 디바이스(502)는 또한 블록체인 네트워크(106)의 노드(104)일 수 있다는 것에 주의한다. 즉, 일부 예들에서, 종단 디바이스(502)는 블록체인 프로토콜 클라이언트 또는 지갑 애플리케이션을 동작시킬 수 있다.

[0088] IoT 네트워크(501)는 블록체인 네트워크 인프라스트럭처를 사용하여 커맨드 및 제어 계층을 결합함으로써 중앙화와 탈중앙화 사이의 균형을 유지한다. 네트워크(501)의 사용자들은 클라이언트-서버뿐만 아니라 디바이스들 간의 피어-투-피어 관계들을 포함하는 자체 다중레벨 제어 계층을 생성할 수 있다. 네트워크 아키텍처는 3개의 층들: IoT 네트워크(501), 블록체인 P2P 네트워크(104)(즉, 전체 및 경량 블록체인 클라이언트들, 예컨대, 마스터, 서번트 및 슬레이브 노드들은 SPV 지갑들을 운영하는 경량 클라이언트들임), 및 블록체인 채굴 네트워크(504)(IoT 노드들에 의해 전파되는 트랜잭션들을 유효성 검증, 전파 및 저장하는 블록체인 P2P 네트워크의 서브세트)를 포함한다. 블록체인 네트워크(106)는 백-엔드 인프라스트럭처로서 작용하며 IoT 네트워크(501)와 블록체인 P2P 네트워크(106) 사이에 오버랩이 있다.

[0089] **퍼미셔닝 프로토콜**

[0090] 본 개시의 실시예들은 네트워크(501)에 대한 액세스를 요청하는 노드(503) 또는 디바이스(502)에 네트워크(501)에 참여하기 위한 허가를 부여하기 위한 프로토콜을 제공한다. IoT의 맥락에서, 새로운 노드들(503)은 등록

기관(예컨대, 네트워크 내의 신뢰 엔티티)에 의해 제공되는 온-체인 위조 방지 디지털 인증서를 사용하여 IoT 네트워크(501) 상에서 허가된다. 프로토콜은 진성 노드(genuine node)들만이 네트워크에 액세스하고 그리고/또는 네트워크 내의 다른 노드들 또는 디바이스들을 제어할 수 있도록 보장함으로써 사이버-공격들과 연관된 문제들을 해결한다.

[0091] IoT 네트워크(501)에 참여하기 위한 허가는 등록 기관(등록 기관은 "허가 부여 기관" 또는 "보증 기관"으로서 또한 지칭될 수 있음)에 의해 부여된다. 등록 기관은 요청 엔티티들(예컨대, 요청 노드 또는 요청 디바이스)에 디지털 인증서들을 발행하는 것을 담당한다. 유효한 인증서를 갖는 엔티티는 IoT 네트워크(501)에 대한 액세스를 갖는다. 등록 기관은 각각이 하나 이상의 프로세서들, 예컨대, 하나 이상의 CPU(central processing unit)들, 가속기 프로세서들, 애플리케이션 특정 프로세서 및/또는 FPGA(field programmable gate array)들을 포함하는 개개의 프로세싱 장치를 포함하는 개개의 컴퓨터 장비를 포함한다. 등록 기관의 컴퓨팅 장비는 또한 메모리, 즉 비-일시적 컴퓨터-판독 가능 매체 또는 매체들의 형태의 컴퓨터-판독 가능 저장소를 더 포함한다. 메모리는 하나 이상의 메모리 매체들, 예컨대, 하드 디스크와 같은 자기 매체; 솔리드 스테이트 드라이브(SSD), 플래시 메모리 또는 EEPROM과 같은 전자 매체; 및/또는 광학 디스크 드라이브와 같은 광학 매체를 사용하는 하나 이상의 메모리 유닛들을 포함할 수 있다.

[0092] 요청 엔티티가 네트워크(501)에 참여하도록 허가를 부여하기 위해, 등록 기관은 "인증서 트랜잭션"으로서 아래에서 지칭되는 블록체인 트랜잭션 Tx를 생성한다. 예시적인 인증서 트랜잭션이 도 16a에서 예시된다. 인증서 트랜잭션 Tx는 하나 이상의 입력들 및 하나 이상의 출력들을 포함한다. 적어도 하나의 입력(1501a)은 등록 기관의 디지털 서명을 포함한다. 즉, 등록 기관은 디지털 서명이 생성될 수 있는 제1 개인 키(예컨대, 제1 개인-공개 키 쌍)를 갖고, 등록 기관은 그 디지털 서명을 사용하여 트랜잭션에 서명한다. 인증서 포맷들의 예는 도 16b에서 예시된다. 인증서 트랜잭션에 서명함으로써, 등록 기관은 트랜잭션의 출력(들)에 포함된 데이터를 증명한다. 디지털 서명은 제1 개인 키에 관한 지식을 갖는 등록 기관에 의해서만 생성될 수 있다. 트랜잭션은 또한 등록 기관에 의해 요청자에게 발행된 디지털 인증서를 포함하는 제1 출력(1502a)(예컨대, 지출 불가능한 출력)을 갖는다. 디지털 인증서는 요청자에게 할당된 식별자를 포함한다. 식별자는 IoT 네트워크(501) 내에서 요청자에게 고유하다. 요청자에게는, 한 번 발행되면 고정된 채로 유지되어야 하고 디바이스가 발행받은 임의의 인증서에서 나타나는 식별자가 할당된다. 바람직하게는 디바이스 식별자는 인증서가 생성될 때 할당된다. 그러나 요청자가 이미 디바이스 식별자를 갖고 있는 경우가 제외되지 않으며, 이 디바이스 식별자는 그 후 인증서에의 포함을 통해 보증된다.

[0093] 일단 생성되면, 등록 기관은 블록체인(150)에 기록될 블록체인 네트워크(106)의 하나 이상의 노드들(104)에 인증서 트랜잭션을 송신한다. 일단 블록체인(150)에 기록되면, 요청자는 인증서를 사용하여, 요청자가 네트워크(501)에 참여하기 위한 허가를 부여받았다는 것을 네트워크(501)의 다른 노드들 또는 디바이스들에 입증할 수 있다. 예컨대, 네트워크(501)의 다른 노드들(503)과 통신할 때, 요청자는 인증서 트랜잭션 및 이에 따라 인증서를 식별하는 정보를 포함할 수 있다.

[0094] 도 1 내지 도 3을 참조하면, 이 예들에서, 제1 노드는 앨리스(103a)의 컴퓨터 장비(102a)일 수 있고 제2 노드는 밥(103b)의 컴퓨터 장비(102b)일 수 있다.

[0095] 요청자가 네트워크(501)의 노드(503)인 경우(또는 노드로서 네트워크(501)에 참여하기 위한 허가를 요청하는 경우), 인증서는 그 노드에 할당된 고유 공개 키를 포함할 수 있다. 공개 키는 요청 노드(503)가 일단 네트워크(501)에 참여하면, 블록체인 트랜잭션들을 송신 및 수신할 수 있게 한다.

[0096] 개인 키는 개인 키의 소유자에게만 알려지는 비밀 숫자 값이다. 예컨대, 개인 키는 256-비트 스트링일 수 있다. 공개 키는 개인 키로부터 도출되고 공유될 수 있는 연관된 공개 값이다. 예컨대, 공개 키는 개인 키와 secp256k1 타원 곡선 생성 포인트의 타원 곡선 곱셈에 의해 계산될 수 있다. 서명은 예컨대, ECDSA(elliptic curve digital signature algorithm)를 사용하여 생성된 것과 같은 암호화 서명일 수 있다. 라빈(Rabin) 서명과 같은 대안적인 서명 체계들이 사용될 수 있다.

[0097] 인증서 트랜잭션은 등록 기관의 제2 공개 키에 잠긴 제2 출력(1502b)을 포함할 수 있다. 제2 공개 키는 인증서 트랜잭션에 서명하는 서명을 생성하는 데 사용된 공개 키와 동일할 수 있거나 상이한 공개 키일 수 있다. 제2 출력(1502b)은, 제2 공개 키에 관한 지식이 출력을 잠금해제하는 데 요구된다는 의미에서 제2 공개 키에 잠긴다. 예컨대, 제2 출력은 제2 공개 키의 해시를 포함할 수 있고, 추후 트랜잭션의 입력에 의해 잠금해제되기 위해, 그 입력은 제2 공개 키를 포함해야 한다. 제2 출력(1502b)이 제2 트랜잭션의 입력과 함께 실행될 때, 입력에서 제공된 제2 공개 키는 해싱되고 제2 출력(1502b)에 포함된 해시와 비교된다. 2개의 해시들이 매칭되

는 경우, 제2 출력(1502b)이 잠금해제될 수 있다(임의의 부가적인 제약들이 충족된 경우).

[0098] 출력은 P2PKH(pay-to-public-key-hash)를 통해 공개 키에 잠길 수 있다. P2PKH는 출력을 공개 키 해시에 잠그는 스크립트 패턴이다. 수신자가 공개 키 해시와 매칭되는 공개 키에 대해 유효한 서명을 제공하는 경우 P2PKH 출력들이 지출될 수 있다. 즉, P2PKH 출력은 지출자에게 2개의 아이템들: 공개 키의 해시가 P2PKH 출력의 주소와 매칭되도록 하는 공개 키, 및 공개 키 및 트랜잭션 메시지에 대해 유효한 서명(반드시 그 순서일 필요는 없음)을 제공하도록 요구한다.

[0099] 제2 출력(1502b)이 등록 기관의 공개 키에 잠기기 때문에, 등록 기관만이 인증서를 취소(revoke)할 수 있다. 이는, 악의적인 당사자로부터 인증서가 취소되는 것을 방지할 수 있다.

[0100] 제2 출력(1502b)은 제2 공개 키에 시간-잠금될 수 있다. 시간-잠금된 출력은 미리 결정된 시간 기간 후까지 잠금해제될 수 없다. 예컨대, 등록 기관은 인증서 트랜잭션에 잠금 시간을 포함시킬 수 있다. 잠금 시간은 인증서 트랜잭션의 제2 출력(1502b)이 특정 시간(예컨대, Unix 시간 또는 블록 높이에 의해 지정될 수 있음) 후까지 추후 트랜잭션에 의해 성공적으로 지출되는 것을 방지한다. 잠금 시간은 트랜잭션의 "nLocktime" 필드를 사용하여 구현될 수 있다. nLocktime은 출력이 지출될 수 있는 최소 시간을 요구하는 트랜잭션의 파라미터이다. nLocktime과 함께, 작업코드(예컨대, OP_CHECKLOCKTIMEVERIFY(CLTV))는 제1 트랜잭션 상의 nLockTime이 작업코드에 제공된 시간 파라미터 이상이 아니라면, 추후 트랜잭션이 (스크립트 실행을 못하게 함으로써) 제2 출력을 지출하는 것을 방지할 것이다. 추후 트랜잭션은 그의 nLockTime이 과거인 경우에만 유효한 블록에 포함될 수 있으므로, 이는 추후 트랜잭션이 유효한 블록에 포함될 수 있기 전에 CLTV-기반 시간블록이 만료되는 것을 보장한다.

[0101] 부가적으로 또는 대안적으로, 제2 출력(1502b)은 "다중-서명" 출력일 수 있다. 다중-서명 출력은 다수의 공개 키들, 즉 등록 기관의 제2 공개 키와 하나 이상의 다른 공개 키에 잠긴다. 다른 공개 키는 네트워크(501)의 다른 노드, 또는 IoT 네트워크 외부에 있지만 블록체인 네트워크(106) 내에 있는 제3자 노드의 공개 키일 수 있다. 제2 출력(1502b)을 지출하려고 시도하는 추후 트랜잭션의 입력은 제2 출력이 잠기는 각각의 공개 키마다 하나씩, 다수의 서명들을 포함해야 한다.

[0102] 시간-잠금은 등록 기관이 합의된 시간 이전에 또는 다른 노드(예컨대, 마스터 노드)의 허가 없이 인증서를 취소하는 것을 방지한다. 다중-서명 출력은 상이한 노드(예컨대, 마스터 노드)의 허가 없이 인증서가 취소되는 것을 방지한다. 양 기술들은 최소 인증서 지속기간 길이를 시행한다.

[0103] 각각의 트랜잭션은 블록체인(150)에 기록될 때, 고유한 트랜잭션 식별자 TxID로 식별될 수 있다. 직렬화된 트랜잭션 바이트들의 (이중) SHA256 해시를 컴퓨팅함으로써 트랜잭션 식별자가 생성될 수 있다. 다른 해시 함수들이 SHA256 대신 사용될 수 있다. 등록 기관은 인증서 트랜잭션의 트랜잭션 식별자를 요청자에게 송신할 수 있다. 이는 요청자는 인증서 트랜잭션을 식별할 수 있게 하고 이에 따라 인증서 트랜잭션 내의 인증서를 획득할 수 있게 한다. 대안적으로, 요청자는 등록 기관의 주소로부터 블록체인(150)으로 송신되는 트랜잭션을 리스닝(listen)할 수 있다.

[0104] 요청자가 노드(예컨대, 서번트 노드)로서 네트워크(501)에 참여하는 경우, 요청 노드는 등록 기관의 제1 공개 키를 획득하고 그 제1 공개 키로부터 전송된 하나 이상의 추가 트랜잭션들(즉, 추가 인증서 트랜잭션들)을 식별하기 위해 트랜잭션 식별자를 사용할 수 있다. 추가 트랜잭션들은 네트워크(501)의 하나 이상의 추가 노드들 또는 디바이스들의 개개의 인증서를 각각 포함할 수 있다. 그 후 요청자는 이 인증서들을 획득(예컨대, 다운로드 및 저장)할 수 있다. 인증서들 내의 정보(예컨대, 디바이스 식별자 및/또는 공개 키)는 네트워크(501)의 다른 노드들(503) 및/또는 디바이스들(502)과 통신하는 데 사용될 수 있다. 예컨대, 요청자는 예컨대, 보증된 공개 키에 잠긴 트랜잭션에 출력(예컨대, P2PKH 출력)을 포함시킴으로써 그 노드의 보증된 공개 키를 사용하여 블록체인 트랜잭션을 다른 노드(503)에 송신할 수 있다. 커맨드를 수신할 때, 요청자는 커맨드가 허가된 노드(503) 또는 디바이스(502)로부터 발행되었는지를 체크하기 위해 인증서들을 사용할 수 있다.

[0105] 요청자가 블록체인에 액세스할 수 없는 종단 디바이스(502)로서 네트워크(501)에 참여하는 경우, 등록 기관은 예컨대, 유선 연결 또는 Bluetooth, Wi-Fi 등과 같은 예컨대, 유선 연결 또는 무선 연결을 통해 인증서를 종단 디바이스로 송신할 수 있다. 등록 기관은 또한 하나 이상의 제2 인증서들의 세트를 요청 종단 디바이스(requesting end device)(502)에 송신할 수 있다. 네트워크(501)의 개개의 노드 또는 종단 디바이스에 각각 발행된 이러한 제2 인증서들은 요청 종단 디바이스의 통신이 허가된 노드들(503) 및 디바이스들(502)을 향하거나 이로부터 오는 것임을 보장하는 데 사용될 수 있다.

[0106] 각각의 인증서(제1 및 제2)는 인증서가 발행되는 노드(503) 또는 종단 디바이스(502)의 네트워크 주소(예컨대, IP 주소)를 포함할 수 있다. 요청자는 허가된(즉, 보증된) 노드의 네트워크 주소를 사용하여 노드와 통신 예컨대, 센서 판독 또는 커맨드 확인응답(acknowledgement)을 전송할 수 있다.

[0107] 등록 기관은 요청자에게 발행된 인증서를 네트워크(501)의 하나 이상의 노드들 및/또는 종단 디바이스들에 송신할 수 있다. 이러한 종단 디바이스들은 인증서를 사용하여 요청자와 통신하고 요청자가 네트워크(501)에 참여하기 위한 허가를 부여받았는지를 검증할 수 있다.

[0108] 제3자들이 인증서의 콘텐츠들(민감한 정보를 포함할 수 있음)을 보는 것을 방지하기 위해, 등록 기관은 인증서를 암호화할 수 있다. 예컨대, 인증서는 등록 기관의 공개 키(등록 기관의 제1 및/또는 제2 공개 키들과 동일할 수 있거나 동일하지 않을 수도 있음)에 기초한 암호화 키를 사용하여 암호화될 수 있다. 대안적으로, 암호화 키는 등록 기관에 의해 생성된 난수일 수 있다.

[0109] 등록 기관의 일부 예들에서, 등록 기관은 명시적 요청을 수신하지 않고 단순히 요청자들에게 인증서들을 발행할 수 있다. 다른 예들에서, 요청자는 먼저 등록 기관에 요청을 송신할 수 있다. 요청은 요청자의 하나 이상의 크리덴셜들을 포함할 수 있다. 예컨대, 크리덴셜들은 다음 즉, 디바이스 유형(예컨대, 랩톱, 전화, 오븐, 냉장고 등), 노드 유형(예컨대, 마스터 노드, 서번트 노드, 슬레이브 노드, 종단 디바이스), 네트워크 주소(예컨대, IPv6 주소일 수 있는 IP 주소) 등 중 하나 이상을 포함할 수 있다. 등록 기관 또는 상이한 노드(예컨대, 마스터 노드)가 요청을 유효성 검증할 수 있다. 유효성 검증되는 경우, 등록 기관은 제1 트랜잭션을 생성하고 이를 블록체인 네트워크(106)로 송신할 수 있다. 요청이 허여되지 않는 경우, 등록 기관은 트랜잭션을 생성하지 않을 수 있다.

[0110] 일부 경우들에서, 요청자에게 발행된 인증서가 취소되어야 할 필요가 있을 수 있다. 예컨대, 요청자는 손상되었을 수 있거나 결함이 발생했을 수 있다. 인증서를 취소하기 위해, 등록 기관은 제2 블록체인 트랜잭션("취소 트랜잭션")을 생성한다. 취소 트랜잭션은 인증서 트랜잭션의 제2 출력(즉, 등록 기관의 제2 공개 키에 잠긴 출력)을 참조하는 입력을 갖는다. 입력은 제2 공개 키에 링크된 서명을 포함한다. 인증서 트랜잭션의 제2 출력이 P2PKH 출력인 경우, 취소 트랜잭션의 입력은 공개 키의 해시(예컨대, OP_HASH160)가 P2PKH 출력의 공개 키 해시와 매칭되도록 하는 공개 키를 포함해야 한다. P2PKH 출력은 지출자에게 2개의 아이템들: 공개 키의 해시가 P2PKH 출력의 주소와 매칭되도록 하는 공개 키, 및 공개 키 및 트랜잭션 메시지에 대해 유효한 서명(반드시 그 순서일 필요는 없음)을 제공하도록 요구한다.

[0111] 취소 트랜잭션은 하나 이상의 출력, 예컨대, 등록 기관의 제3 공개 키에 잠긴 출력(이는 등록 기관의 제1 및/또는 제2 공개 키와 동일할 수 있거나 동일하지 않을 수도 있음)을 포함할 수 있다. 그 후 등록 기관은 블록체인(150)에 기록되도록 블록체인 네트워크(106)에 취소 트랜잭션을 송신한다. 취소 트랜잭션이 블록체인(150)에 기록되면, 인증서 트랜잭션은 UTXO(unspent transaction output) 세트로부터 제거될 것이다. UTXO는 다른 블록체인 트랜잭션에 의해 지출되지 않은 블록체인 트랜잭션의 출력이다. 네트워크(501) 상의 상이한 노드가 요청 노드에 발행된 인증서를 식별하려고 시도할 때, 그 노드는 인증서를 포함하는 인증서 트랜잭션이 지출되었음을 발견하고 이를 인증서가 취소되는 것으로서 해석할 것이다. 네트워크(501)의 노드들은 발행 주소(즉, 제2 공개 키)로의 그리고 그로부터 생성된 트랜잭션을 주시함으로써 자신의 피어 목록(즉, 허가/보증된 노드들의 목록)을 동적으로 업데이트할 수 있다.

[0112] 노드/디바이스 인증서의 유효성은 3개의 기준들: 발행 키는 인식된 발행 키(마스터 키에 의해 서명된 인증서에 포함됨)이고, 인증서는 미리 결정된 프로토콜에 따라 올바르게 포맷팅되고, 그리고 인증서 트랜잭션의 지출 가능한 출력이 미지출이라는 것에 의존할 수 있다. 인증서들은 취소된 후 업데이트될 수 있다. 등록 기관은 구 인증서의 UTXO를 지출하고 그 후 업데이트된 정보를 갖는 새로운 인증서 Tx를 생성한다. 그 후 등록 기관은 IoT 네트워크(501) 상의 디바이스들에 새로운 인증서 아웃포인트 로케이션] 인덱스를 브로드캐스트할 수 있다. 이는 등록 기관의 자체(자체-서명된) 인증서에도 적용된다.

[0113] 위에서 논의한 바와 같이, 인증서는 요청자의 고유 식별자를 포함한다. 또한, 인증서는 요청자의 고유 공개 키를 포함할 수 있다. 일반적으로, 인증서는 다음 필드들 중 하나 이상을 포함할 수 있다:

표 1

[0114]

필드 크기(바이트)	필드 이름	데이터 유형	설명
4	IOT 프로토콜 식별자	uint32_t	프로토콜을 표시하는 프리픽스

1	페이로드 유형	uint16_t	메시지가 정규 메시지인지 아니면 인증서인지를 표시하는 1바이트 식별자
32	새로운 디바이스 ID	char[32]	새로운 디바이스에 대한 고유 디바이스 ID
33	새로운 디바이스 공개 키	char[32]	노드와 통신하는 데 사용되는 secp256k1 (압축) 공개 키(블록체인 P2P 네트워크 상에 있는 경우)
4	디바이스 유형	uint16_t	디바이스 유형(예컨대, 랩톱, 전화, 오븐 냉장고, 가로등 기둥 등)
4	IOT 노드 유형	uint16_t	노드 유형 예컨대, 마스터, 서번트, 슬레이브, 종단 디바이스
16 + 2	IPv6 주소 + 포트	Char[16]	IPv6 네트워크 주소. 네트워크 바이트 순서
4	UNIX 시간 생성 날짜	uint16_t	디바이스 생성 날짜
4	UNIX 시간 인증서 만료 날짜	uint16_t	인증서가 만료되고 새로운 인증서가 발행되어야 하는 UNIX 시간
0-80	부가적인 디바이스 정보	char[]	부가적인 디바이스 정보(제조자 정보를 포함함).

[0115] 이 포맷의 인증서는 예컨대, 블록체인 트랜잭션의 지출 불가능한(OP_RETURN) 출력으로 인코딩된 104 내지 184 바이트의 데이터를 요구한다.

[0116] 위에서 언급된 바와 같이, 네트워크(501)는 마스터 노드(503)를 포함할 수 있다. 일부 예들에서, 등록 기관은 마스터 노드(503a)를 포함할 수 있다. 요청자는 노드(503) 또는 종단 디바이스(502)일 수 있다. 따라서, 마스터 노드(503a)는 자체적으로 노드들(503) 또는 디바이스들(502)에 인증서들을 발행할 수 있다. 부가적인 또는 대안적인 예들에서, 요청자는 마스터 노드(503a)일 수 있다. 마스터 노드(503a)가 등록 기관 및 요청자인 경우, 마스터 노드(503a)는 자체 인증서에 자체-서명한다.

[0117] IoT 네트워크(501)에 대한 액세스는 허가되고 퍼미셔닝(permissioning)(또는 부트스트래핑(bootstrapping)) 알고리즘을 사용하여 새로운 엔티티들(노드들 또는 디바이스들)을 인가한다. 2개의 예시적인 알고리즘들이 아래에서 제공된다. 마스터 노드(503a)는 네트워크(501) 상에서 허용되도록 허가를 요청하는 임의의 새로운 디바이스의 크리덴셜들을 (직접적으로 또는 간접적으로) 검증할 수 있는 등록 기관에 의해 제어될 수 있다. 마스터 노드(503a)는 그 후 네트워크(501) 상의 다른 모든 노드들(503)에 브로드캐스트될 온-체인 인증서를 발행할 수 있다. 블록체인 지갑들을 동작시키지 않는 디바이스들의 부트스트래핑 알고리즘은 블록체인 트랜잭션들을 통하는 대신 IP 주소들 간의 통신에 의존하기 때문에 상이하다.

[0118] 예시적인 퍼미셔닝 알고리즘(마스터/서번트/슬레이브)

[0119] 단계 1: 노드를 실행하는 컴퓨팅 디바이스는 모든 관련 제조 정보를 포함하여 그의 크리덴셜들을 등록 기관에 등록한다. 등록 기관은 마스터 노드(503a)를 동작시키는 동일한 엔티티일 수 있다. 그러나 정규 커맨드들 또는 메시지들에 서명하는 데 사용되는 공개 키는 바람직하게는 인증서 트랜잭션에 서명하는 데 사용되는 공개 키와 상이해야 한다.

[0120] 단계 2: 키 PK_{Issue} 에 대한 제어를 갖는 등록 기관은 크리덴셜들의 진위를 유효성 검증하고 디바이스가 IoT 네트워크(501)에 진입하도록 인가되어야 하는지를 결정한다.

[0121] 단계 3: 디바이스가 인가되는 경우, 등록 기관은 노드에 대한 고유한 인증서 트랜잭션을 생성한다. 이 트랜잭션은 PK_{Issue} 로부터 PK_{Issue} 주소로 브로드캐스트된다. 등록 기관은 새로운 노드에 TxID를 전송한다.

[0122] 단계 4: 새로운 노드는 트랜잭션을 찾고 등록 기관으로 PK_{Issue} 를 식별한다. 새로운 노드는 PK_{Issue}에 의해 발행되고 현재 활성인 모든 인증서들을 다운로드 및 검증한다.

[0123] 단계 5: 네트워크(501) 상에 이미 있는 서번트 및 슬레이브 노드들은 PK_{Issue} 로 그리고 이로부터 브로드캐스트되는 트랜잭션들을 리스닝하도록 구성된다. 이들이 새로운 트랜잭션을 보고 새로운 디바이스 인증서를 다운로드하고 평가하면, 이들은 새로운 노드와 통신하도록 그의 지갑들을 구성할 수 있다.

[0124] 예시적인 퍼미셔닝 알고리즘(종단 디바이스)

[0125] 단계 1: 종단 디바이스(502)는 모든 관련 제조 정보 및 IP 주소를 포함하는 자신의 크리덴셜들을 등록 기관에

등록한다. 재차, 등록 기관은 마스터 노드(503a)를 동작시키는 동일한 엔티티일 수 있다. 그러나 정규 커맨드들 또는 메시지들에 서명하는 데 사용되는 공개 키는 바람직하게는 인증서 트랜잭션에 서명하는 데 사용되는 공개 키와 상이해야 한다.

[0126] 단계 2: 키 PK_{Issue}에 대한 제어를 갖는 등록 기관은 크리덴셜들의 진위를 유효성 검증하고 디바이스(502)가 IoT 네트워크(501)에 진입하도록 인가되어야 하는지를 결정한다.

[0127] 단계 3: 디바이스(502)가 인가되는 경우, 등록 기관은 디바이스에 대한 고유한 디바이스 인증서 트랜잭션을 생성한다. 이 트랜잭션은 PK_{Issue}에 의해 제어되는 주소로 그리고 이로부터 브로드캐스트된다. 인증서 데이터는 암호화되어서, IOT 노드들만이 세부사항들을 볼 수 있다.

[0128] 단계 4: 등록 기관은 종단 디바이스(502)가 피어-목록을 생성하는 것을 가능하게 하는 인증서들의 목록을 전송한다. 이 목록은 종단 디바이스(502)가 네트워크(501) 상의 다른 노드들(503c)에 연결되게 하고 공개 키 계층을 해석할 수 있게 할 것이다.

[0129] 단계 5: 네트워크(501) 상의 노드들(503)은 디바이스 인증서를 볼 수 있을 것이고 종단 디바이스와 IP 대 IP로 통신할 수 있다(TLS).

[0130] **요청 및 응답 프로토콜**

[0131] 본 개시는 또한 커맨드 요청들을 발행하고, 이러한 커맨드 요청들에 기초하여 디바이스들에 지시하고, 커맨드 확인응답들을 발행하기 위해 블록체인 트랜잭션들 Tx를 사용하도록 네트워크(예컨대, IoT 네트워크)(501)의 노드들을 위한 프로토콜을 제공한다. IoT 네트워크(501)와 관련하여 실시예들이 설명될 것이지만, 일반적으로, 본 발명의 교시는 블록체인 프로토콜 클라이언트 애플리케이션(105)을 동작시키는 노드들, 및 이러한 노드들의 적어도 서브세트에 의해 제어 가능한 종단 디바이스들을 포함하는 임의의 네트워크에 적용될 수 있다.

[0132] 네트워크(501)의 제1 브리징 노드(503)(예컨대, 마스터 노드(503a) 또는 서버 노드(503b))는 제1 노드에 의해 서명된 입력 및 커맨드 데이터를 포함하는 출력을 포함하는 제1 트랜잭션 Tx_1을 생성한다. 커맨드 데이터는 제어될 종단 디바이스(502)의 식별자 및 종단 디바이스(502)를 제어하기 위한 커맨드 메시지를 포함한다. 제1 노드는 커맨드의 발신자일 수 있다. 즉, 제1 노드는 커맨드 데이터를 생성할 수 있다.

[0133] 제1 노드는 종단 디바이스(502)를 제어하는 제1 네트워크(501)(예컨대, 슬레이브 노드(503c))의 제2 브리징 노드(503)에 제1 트랜잭션 Tx_1을 송신할 수 있다. 제1 트랜잭션 Tx_1은 오프체인(off-chain)으로, 즉 블록체인으로 송신되지 않고 송신될 수 있다. 예컨대, 제1 트랜잭션 Tx_1은 예컨대, 인터넷을 통해 제1 노드로부터 제2 노드로 직접 전송될 수 있다. 예컨대, 제1 노드는 서버 노드(503b)일 수 있고 제2 노드는 슬레이브 노드(503c)일 수 있다. 대안적으로, 제1 트랜잭션 Tx_1은 예컨대, 하나 이상의 중개 노드들을 통해 간접적으로 전송될 수 있다. 예컨대, 제1 트랜잭션 Tx_1은 서버 노드(503b)를 통해 마스터 노드(503a)로부터 슬레이브 노드(503c)로 전송될 수 있다. 제2 노드는 유선 또는 무선 연결을 통해, 예컨대, 이더넷 또는 Wi-Fi 연결을 통해 종단 디바이스(502)에 연결될 수 있다.

[0134] 부가적으로 또는 대안적으로, 제1 노드는 블록체인(150)에 기록되도록 블록체인 네트워크(106)에 제1 트랜잭션 Tx_1을 송신한다. 이는 제1 트랜잭션 Tx_1이 유효한 트랜잭션이라는 것에 의존한다. 아래에서 논의되는 바와 같이, 일부 경우들에서, 제1 트랜잭션 Tx_1을 블록체인으로 송신하지 않는 것이 바람직하다.

[0135] 도 1 내지 도 3을 참조하면, 이 예들에서, 제1 노드는 앨리스(103a)의 컴퓨터 장비(102a)에 의해 구성될 수 있고 제2 노드는 밥(103b)의 컴퓨터 장비(102b)에 의해 구성될 수 있다. 이전에 설명된 바와 같이 앨리스 및 밥은 당사자들 중 하나가 네트워크(106)에 브로드캐스트하기로 선택할 때까지, (여전히) 트랜잭션이 블록체인 네트워크(106) 상에 공개되거나 체인(150)으로 진행되게 함 없이 트랜잭션을 교환하기 위해 사이드 채널(예컨대, 사이드 채널(301))을 사용할 수 있다.

[0136] 제2 노드는 제1 노드로부터 직접적으로 또는 간접적으로 제1 트랜잭션 Tx_1을 획득할 수 있는데, 예컨대, 제1 트랜잭션 Tx_1은 하나 이상의 중개 노드들을 통해 제2 노드에 포워딩될 수 있다. 제2 노드는 커맨드 데이터를 사용하여 커맨드 데이터에서 디바이스 식별자("디바이스 ID")에 의해 식별된 종단 디바이스(502)에 제어 명령을 송신한다. 커맨드 데이터의 제어 메시지는 종단 디바이스(502)의 원하는 동작을 정의할 수 있다. 제어 메시지

는 제2 노드가 여러 가능한 명령들 중 특정 명령을 종단 디바이스(502)에 송신하게 하도록 구성될 수 있다. 대안적으로, 제2 노드는 단일 명령을 종단 디바이스(502)에 전송하도록 구성될 수 있는데, 즉 제2 노드는 항상 동일한 명령만을 종단 디바이스에 전송한다. 이는 예컨대, 종단 디바이스(502)가 센서와 같은 단순한 디바이스이고 명령이 센서 판독에 대한 요청인 경우에 해당할 수 있다.

[0137] 커맨드(즉, 종단 디바이스에 대한 명령)는 예컨대, Wi-Fi를 사용하여 유선 또는 무선 연결을 통해 디바이스에 오프체인으로 송신될 수 있다. 대안적으로, 디바이스가 또한 네트워크의 노드인 경우, 커맨드는 블록체인 트랜잭션 Tx를 통해 송신될 수 있다.

[0138] 일부 실시예들에서, 디바이스 및 제어기 통신을 위한 요청 및 응답 사이클은 제1 및 제2 노드들에 의해 구현될 수 있다. 요청(커맨드)은 커맨드 데이터(예컨대, OP_RETURN 페이로드)를 포함하는 출력을 포함하는 부분적으로 완료된 트랜잭션으로서 발행된다. 응답(커맨드의 확인응답)은 요청자와 응답자 노드들 둘 모두의 서명을 포함하는 완성된 트랜잭션의 브로드캐스팅이다. 트랜잭션 유연성(malleability)은 메시지 수신자가 커맨드 데이터(예컨대, OP_RETURN 페이로드)를 변경할 수 없는 동안 입력들 및 출력들을 추가할 수 있기 때문에 이러한 통신 방법을 가능하게 한다.

[0139] 작업코드는 데이터에 대한 스택-기반 연산 및 암호화-연산들을 수행하도록 채굴자들(104M)에 지시하기 위해 스크립트 엔진(402) 내에서 사용되는 명령 음절 또는 파슬(parcel)이다. 여기서, 스크립트 엔진(402)은 블록체인 트랜잭션 Tx 내에서 스크립트를 유효성 검증하는 데 사용되는 실행 환경이고 스택(403)은 2개의 주요 동작들: 모음에 요소를 추가하는 '푸시(push)' 및 가장 최근에 추가된 요소를 제거하는 '팝(pop)'을 갖는 데이터 구조(요소들의 모음)이다. 작업코드는 스택 요소에 대한 동작들을 수행하도록 설계된다. 트랜잭션 Tx를 유효성 검증할 때, 스크립트 엔진(402)은 OP_RETURN 작업코드 이후에 출력 스크립트(ScriptPubkey)의 임의의 데이터를 실행하지 않을 것이다. 실제로 이것은 나머지 스크립트 데이터가 임의적일 수 있고 출력 자체가 지출 불가능함을 의미한다(하나의 블록체인 프로토콜에서, 출력의 비-지출성을 보장하기 위해 OP_RETURN에는 OP_FALSE 작업코드가 선행될 필요가 있음).

[0140] 제1 노드로부터 제2 노드로 송신되는 제1 트랜잭션 Tx_1은 제2 출력 없이 송신될 수 있다. 즉, 트랜잭션은 단일 출력(커맨드 데이터를 포함하는 출력)을 포함한다. 부분적 트랜잭션을 완료하기 위해, 제2 트랜잭션은 제1 트랜잭션에 입력 및 출력을 추가함으로써 트랜잭션을 업데이트할 수 있다. 입력은 제2 노드의 서명, 즉, 제2 노드의 개인 키를 사용하여 생성된 서명을 포함한다. 출력은 제2 노드의 공개 키에 잠긴 출력 예컨대, P2PKH 출력이다. P2PKH 출력을 지출하기 위해, 지출 트랜잭션의 입력은, 공개 키의 해시(예컨대, OP_HASH160)가 P2PKH 출력의 공개 키 해시와 매칭되도록 하는 공개 키를 포함해야 한다. P2PKH 출력은 지출자에게 2개의 아이템들: 공개 키의 해시가 P2PKH 출력의 주소와 매칭되도록 하는 공개 키, 및 공개 키 및 트랜잭션 메시지에 대해 유효한 서명(반드시 그 순서일 필요는 없음)을 제공하도록 요구한다. 공개 키는 서명을 생성하는 데 사용되는 개인 키에 대응할 수 있다. 대안적으로, 서명은 제1 공개 키에 링크될 수 있고 출력이 상이한 공개 키에 잠길 수 있다. 제2 노드는 그 후 완료된 트랜잭션을 블록체인 네트워크(106)에 송신할 수 있다. 완료된 트랜잭션(이러한 실시예들에서, 커맨드 트랜잭션으로서 지칭됨)은 다른 노드들, 예컨대, 제1 노드가 보도록 블록체인(150)에서 이용 가능하고, 디바이스에 의해 수행된 커맨드의 레코드로서 작용한다. 즉, 트랜잭션이 브로드캐스트되면, 독립적인 관찰자가 어느 공개 키가 커맨드/메시지를 발행했는지 그리고 어느 공개 키가 그에 응답했는지를 볼 수 있다.

[0141] 도 7a 및 도 7b는 예시적인 부분적 제1 트랜잭션 Tx_1(부분) 및 예시적인 업데이트된 제1 트랜잭션 Tx_1(완료)을 예시한다. 부분적 제1 트랜잭션은 단일 입력(701a) 및 단일 출력(702a)을 포함한다. 업데이트된 제1 트랜잭션은 제2 노드에 의해 추가된 입력(701b) 및 출력(702b)을 포함한다. SIGHASH_SINGLE 서명 유형은 원하는 레벨의 트랜잭션 유연성을 획득하는데 사용될 수 있다. 예컨대, 공개 키 PK_0를 갖는 노드는 공개 키 PK_1를 갖는 노드에 명령을 전송한다. 명령은 SIGHASH_SINGLE 서명 유형을 사용하여 서명된 트랜잭션의 지출 불가능한 출력(예컨대, OP_RETURN 출력)에서 인코딩된다(도 10a). 부분적 완료 트랜잭션은 유효하다. 명령의 완료 시에, PK_1을 갖는 제2 노드는 그의 주소에 잠긴 출력을 추가한다. 그 후, PK_1을 가진 제2 노드는 SIGHASH_ALL 서명 유형을 사용하여 전체 트랜잭션에 서명함으로써 트랜잭션을 완성시킨다(도 10b 참조).

[0142] 대안적인 실시예들에서, 제1 노드로부터 제2 노드로 송신되는 제1 트랜잭션 Tx_1은 제2 출력과 함께 송신될 수 있다. 제2 출력은 제2 노드의 공개 키에 잠긴다. 예컨대, 제2 출력은 제2 노드의 공개 키에 대한 P2PKH일 수

있다.

[0143] 제1 트랜잭션 Tx_1을 완료하기 위해, 제2 노드는 제1 트랜잭션에 입력을 추가함으로써 제1 트랜잭션을 업데이트한다. 제1 트랜잭션 Tx_1은 이제 2개의 입력들 및 2개의 출력들을 포함한다. 제2 입력은 제2 노드의 공개 키를 포함한다. 제2 입력의 공개 키는 제2 출력이 잠기는 공개 키와 동일할 수 있거나 동일하지 않을 수도 있다. 완료되면, 업데이트된 제1 트랜잭션(이 실시예들에서 커맨드 트랜잭션으로서 지칭됨)은 블록체인(150)에의 포함을 위해 블록체인 네트워크(106)로 전송된다. 커맨드 트랜잭션이 브로드캐스트되면, 임의의 독립적인 관찰자는 어느 공개 키가 커맨드/메시지를 발행했는지 그리고 어느 공개 키가 그에 응답했는지를 볼 수 있다.

[0144] 제2 노드의 공개 키에 잠긴 제2 출력은 제1 트랜잭션의 제1 입력에 의해 참조되는 디지털 자산의 양보다 많은 디지털 자산의 양을 전달할 수 있다. 이 경우에, 제1 트랜잭션 Tx_1은 부분적 완료 트랜잭션이며, 이는 블록체인 네트워크(106)의 다른 노드들에 의해 유효한 것으로 간주되지 않을 것이다. 즉, 제1 트랜잭션 Tx_1은 블록체인 노드가 따르는 합의 규칙들을 충족하지 않을 것이고 이에 따라 블록체인(150)의 블록(152) 내로 채굴되지 않을 것이다. 제1 트랜잭션 Tx_1을 업데이트할 때, 제2 노드는 제1 및 제2 입력들에 의해 참조되는 디지털 자산의 결합된 양이 제2 출력에 잠긴 디지털 자산의 양보다 크다는 것을 보장해야 할 것이다.

[0145] 도 8a 및 도 8b는 예시적인 부분적 제1 트랜잭션 Tx_1(부분) 및 예시적인 업데이트된 제1 트랜잭션 Tx_1(완료)을 예시한다. 제1 트랜잭션은 제1 출력(802a) 및 제2 노드의 공개 키에 잠긴 제2 출력(802b)의 커맨드 데이터를 포함한다. 업데이트된 제1 트랜잭션은 제2 노드에 의해 추가된 부가적인 입력(801b)을 포함한다. PK_0을 갖는 제1 노드가 PK_1을 갖는 제2 노드에, PK_1을 갖는 제2 노드에 의해서만 수행되기를 원한다는 명령을 전송하는 경우, 이들은 양 출력들(802a, 802b)을 잠그지만 수수료를 지불하지 않는(이에 따라 채굴되거나 전파되지 않을 것임) 부분적 완료 트랜잭션을 전송할 수 있다. PK_1에 잠긴 디지털 자산을 리딤하기 위해, PK_1을 갖는 제2 노드는 수수료를 지불하는 입력(801b)을 제공할 필요가 있을 것이다. 부분적 완료 트랜잭션을 사용하여 커맨드를 발행하기 위해, $< \mathrm{Sig}_{\mathrm{PK}_n} >$에 대한 SIGHASH 플래그가 SIGHASH_ANYONECANPAY로 설정되고 커맨드 데이터와 함께 OP_RETURN 출력을 포함한다. 이는 제1 출력(802a)에 포함된 커맨드 데이터는 고정되지만, 누구나 부가적인 입력을 추가할 수 있다는 것을 의미한다. 커맨드를 수신한 공개 키는 입력(801a)의 자금들을 리딤하기 위해 부가적인 입력(801b)을 추가할 수 있다. 새로운 입력(801b)을 보호하고 추가 트랜잭션 유연성을 방지하기 위해, 자금의 수신자는 최소 값(dust) 입력을 추가하고 SIGHASH_ALL을 사용하여 트랜잭션 출력에 서명한다.

[0146] SIGHASH 플래그는 서명이 트랜잭션의 어느 부분을 서명했는지를 표시하기 위해 트랜잭션 입력들의 서명들에 추가된 플래그라는 것에 주의한다. 디폴트는 SIGHASH_ALL이다(ScriptSig 이외의 트랜잭션의 모든 부분들이 서명됨). 트랜잭션의 서명되지 않은 부분들은 수정될 수 있다.

[0147] 예시적인 요청 및 응답 알고리즘이 도 9, 도 10a 및 도 10b를 참조하여 아래에 제공된다. 제어 디바이스(503b)는 네트워크(501) 상의 다른 노드들과 통신하도록 구성되고 네트워크 상의 임의의 다른 노드로의 최단 통신 루트를 계산할 수 있다. 예컨대, PK_{serv}는 PK_{slave}가 device_ID를 갖는 디바이스에 가장 가까운 제어기임을 식별한다.

[0148] 단계 1: 공개 키 PK_{serv}를 갖는 제어 디바이스(503b)는 부분 커맨드 Tx_1(도 10a 참조)를 공개 키 PK_{slave}를 갖는 제2 제어 디바이스(503c)로 전송한다. 트랜잭션에 포함된 IoT 메시지는 커맨드 및 device_ID를 갖는 타겟 디바이스를 지정한다.

[0149] 단계 2: 제2 제어 디바이스(PK_{slave})는 트랜잭션에 대한 서명이 유효하고, IoT 메시지 페이로드 내에 포함된 메시지가 네트워크(501)의 규칙들에 따라 유효하다는 것을 체크한다.

[0150] 단계 3. 제2 제어 디바이스(PK_{slave})는 오프 체인 통신(예컨대, 유선 연결, Bluetooth, IP-to-IP)을 통해 디바이스(device_ID)에 커맨드 메시지("Msg")를 전송한다.

[0151] 단계 4: 커맨드 요청 동작의 완료 시에, 디바이스(device_ID)는 커맨드 완료 또는 확인응답 메시지("ack")를 제2 제어 디바이스(PK_{slave})로 다시 전송한다.

[0152] 단계 5: 제2 제어기(PK_{slave})는 제2 입력 및 서명을 추가하고 트랜잭션을 완성시킨다(도 10b 참조). 이는 제2 제어기가 커맨드의 완료를 확인한다는 것을 시그널링할 것이다.

[0153] 단계 6: 제2 제어기(PK_{slave})는 완성된 트랜잭션을 블록체인(채굴) 네트워크(504)에 브로드캐스트한다.

[0154] 일부 실시예들에서, 제1 트랜잭션 Tx_1은 트랜잭션이 아니라 커맨드 요청 트랜잭션 Tx_1(요청)일 수 있다. 즉, 제1 노드는 종단 디바이스(502)를 제어하기 위한 승인을 위해 네트워크의 둘 이상의 노드들에 요청을 송신할 수 있다. 예컨대, 제1 노드는 마스터 노드(503a)로부터 승인을 요청하는 서버 노드(503b) 및 상이한 브리징 노드(503)(예컨대, 다른 서버 노드(503b))일 수 있다. 제1 트랜잭션 Tx_1은 종단 디바이스(502)를 제어하기 위한 커맨드 메시지를 포함하는 커맨드 데이터를 포함한다. 그러나, 종단 디바이스(502)를 제어하는 노드로 제1 트랜잭션 Tx_1을 송신하는 대신, 제1 노드는 커맨드 요청들을 승인할 수 있는 둘 이상의 노드들에 제1 트랜잭션 Tx_1을 송신한다. 즉, 제1 노드는 네트워크(501)의 둘 이상의 노드들에 잠긴 출력을 포함한다. 출력을 잠금해제하기 위해, 각각의 노드로부터의 서명이 추후 트랜잭션 Tx_2의 입력에서 제공되어야 한다. 제1 트랜잭션 Tx_1은 블록체인(150)에의 포함을 위해 블록체인 네트워크(160)로 브로드캐스트된다. 둘 이상의 노드들이 (그의 공개 키들 또는 공개 키 주소들로 지불되는 트랜잭션들을 리스닝함으로써) 트랜잭션 Tx_2를 볼 때, 둘 이상의 노드들은, 이들이 커맨드 요청을 승인하기를 원하는 경우, 커맨드 승인 트랜잭션 Tx_2(승인)의 입력에 각각 서명하고 그 트랜잭션 Tx_2를 블록체인 네트워크(106)에 브로드캐스트한다. 커맨드 승인 트랜잭션 Tx_2은 커맨드 요청 트랜잭션 Tx_1과 동일한 커맨드 데이터를 포함하고, 부가적인 출력은 종단 디바이스(502)를 제어하는 제2 노드의 공개 키에 잠긴다.

[0155] 도 11a 및 도 11b는 예시적인 제1 및 제2 트랜잭션을 예시하며, 제1 트랜잭션 Tx_1은 커맨드 요청 트랜잭션이고 제2 트랜잭션 Tx_2는 커맨드 승인 트랜잭션이다. 제1 트랜잭션 Tx_1은 제1 노드에 의해 서명된 입력(1101a), 커맨드 데이터를 포함하는 제1 출력(1102a) 및 2개의 상이한 공개 키들: 제3 노드의 하나 및 제4 노드의 하나(또한 IoT 네트워크(501)의 브리징 노드(503))에 잠긴 제2 출력(1102b)을 포함한다. 제2 트랜잭션 Tx_2는 제3 및 제4 노드들에 의해 서명된 입력(1101b), 제1 출력(1102a) 및 제2 노드의 공개 키에 잠긴 제2 출력(1102c)을 포함한다. 다중 서명 스크립트들은 커맨드들에 대한 다중-요인 승인을 가능하게 한다. 일부 경우들에서, 커맨드들의 해석 또는 인증서들의 유효성 검사는 둘 이상의 노드들(예컨대, IoT 네트워크의 노드들)로부터의 서명들을 요구할 수 있다. 그것은 또한 지출 가능한 출력이 UTXO 세트에 있는지를 검사하는 것을 요구할 수 있다. 예컨대, 공개 키 PK_0을 가진 제1 노드는 동작을 수행하도록 공개 키 PK_3을 가진 제2 노드에게 지시하기를 원할 수 있다. 보안 요건으로서, 이 커맨드는 공개 키들 PK_1 및 PK_2을 갖는 제3 및 제4 노드들이 각각 승인할 것을 요구한다. 공개 키 PK_0를 가진 제1 노드는 다중 서명 주소에 자금을 전송하는 제1 트랜잭션을 생성한다. 제1 트랜잭션은 또한 승인이 필요한 커맨드 데이터(PK_3에 대한 커맨드를 인코딩함)를 또한 포함한다. 응답으로, 공개 키들 PK_1 및 PK_2을 갖는 노드들(503)은 둘 모두가 서명을 제공하는 경우 다중 서명 주소로부터 자금들을 지출하는 옵션을 갖는다. 승인은 커맨드 요청 트랜잭션의 출력의 지출로서 해석된다. 이 도면들은 2/2(two-of-two) P2MS(pay to multi-sig) 출력을 보여주지만, 일반적으로 P2MS 출력은 n/n 출력일 수 있고 – 여기서 n은 임의의 정수임 – P2MS는 지불인이 출력을 다수의 주소에 잠그게 할 수 있는 일 유형의 스크립트 패턴이다. 출력은 지출되려면, 공개 키들의 지정된 세트로부터의 하나 이상의 서명들을 요구할 수 있다.

[0156] 일부 실시예들에서, 위에서 논의된 바와 같이, 제1 노드는 커맨드 요청 트랜잭션 Tx_1을 생성할 수 있다. 부가적으로 또는 대안적으로, 제1 노드는 IoT 네트워크(501)의 상이한 노드에 의해 생성된 커맨드 요청 트랜잭션에 대한 응답으로 커맨드 승인 트랜잭션 Tx_2를 생성할 수 있다. 예컨대, 제1 노드는 요청들에 대한 승인을 부여할 수 있는 마스터 노드(503a)일 수 있다. 그 경우에, 블록체인(150)은 IoT 네트워크(501)의 하나 이상의 추가 노드들(503)의 하나 이상의 개개의 공개 키들 및 제1 노드의 공개 키에 잠긴 출력을 포함하는 커맨드 요청 트랜잭션을 포함한다. 예컨대, 제1 노드는 제2 노드에 의해 제어되는 종단 디바이스(502)에 발행된 커맨드들을 승인하는 노드일 수 있다. 제1 노드가 커맨드를 승인하는 경우, 제1 노드는 커맨드 요청 트랜잭션의 출력을 참조하는 커맨드 승인 트랜잭션에 서명한다. 제1 노드가 커맨드 승인 트랜잭션에 서명한 마지막 노드인 경우 제1 노드는 트랜잭션을 블록체인 네트워크(106)로 송신한다. 브로드캐스트 트랜잭션 Tx_2는 커맨드 요청의 승인으로서

해석된다.

[0157] 제2 노드(예컨대, 슬레이브 노드)가 커맨드 트랜잭션 또는 커맨드-승인 트랜잭션을 획득할 때, 제2 노드는 커맨드 데이터에서 디바이스 식별자(Device_ID)에 의해 식별되는 디바이스에 커맨드(Msg)를 송신한다. 일부 예들에서, 디바이스(502)는 자신이 커맨드를 수신하고 및/또는 커맨드에 대처했음을 표시하기 위해 확인응답 메시지(Ack)를 제2 노드로 전송할 수 있다. 이들 예들에서, 제2 노드는 디바이스로부터 확인응답을 수신했다는 것을 조건으로, 제1 트랜잭션을 업데이트(그리고 업데이트된 트랜잭션을 브로드캐스트)할 수 있다. 이는 종단 디바이스가 커맨드를 실행했음을 뒷받침하는 증거를 추가 제공한다.

[0158] 대부분의 일상적인 소형 전자 디바이스들이 갖는 자원 제약들로 인해, 이들은 블록체인(150)을 쉽게 모니터링하고 그리고/또는 심지어 그의 인접한 로케이션 외부의 IoT 네트워크 구성요소와 통신할 수 없을 수 있고, 이에 따라 종단 디바이스들(502)의 제어는 로컬로(디바이스에 대한 제2 노드) 그리고 오프-체인으로 수행된다. 종단 디바이스들로의 그리고 이들로부터의 메시지들은 부가적인 트랜잭션 메타데이터 없이 원시 커맨드 데이터(예컨대, OP_RETURN 페이로드)의 형태를 취할 수 있다. 이는 메시지들을 포함하는 데이터 패킷들이 작게 유지되고 계산 집약적인 동작들(이를테면, 타원 곡선 수학)이 요구되지 않는다는 것을 보장한다.

[0159] 일부 실시예들에서, 트랜잭션에 포함된 커맨드 데이터는 암호화된다. 커맨드 데이터는 난수에 기초한 암호화 키로 암호화될 수 있지만, 바람직하게는 암호화 키는 제1 노드의 공개 키 및 제2 노드의 공개 키에 기초하여 생성된다. 제1 및/또는 제2 노드의 공개 키는 보증된 공개 키들일 수 있다(보증된 공개 키들을 아래에서 논의됨). 보증된 공개 키들은 제1 및/또는 제2 노드들에 발행된 개개의 인증서에 포함된다. 이러한 공개 키들은 제1 트랜잭션을 생성하고 업데이트하기 위해 제1 및 제2 노드에 의해 사용되는 공개 키들과 상이할 수 있다. 즉, 서명들을 생성하거나 트랜잭션의 출력들을 잠그는 데 사용되는 공개 키들("트랜잭션 공개 키들")은 암호화 키를 생성하는 데 사용되는 공개 키들과 동일하지 않을 수 있다. 다른 예들에서, 트랜잭션 공개 키들은 암호화 키를 생성하는 데 사용될 수 있다.

[0160] 커맨드 데이터를 암호화하는 데 사용되는 암호화 키는 제1 및 제2 노드들에 의해 독립적으로 생성될 수 있다. 예컨대, 제1 노드는 제1 노드에게 알려진 개인 키 및 제2 노드의 공개 키에 기초하여 암호화 키를 생성할 수 있다. 제2 노드는 제1 노드의 개인 키에 대응하는 공개 키 및 제2 노드의 공개 키에 대응하는 개인 키에 기초하여 암호화 키를 생성할 수 있다. 이는, 예컨대, 제2 노드가 디바이스에 지시하기 위해 커맨드 메시지에 액세스할 수 있도록 제1 및 제2 노드들 둘 모두가 커맨드 데이터를 복호화할 수 있다는 것을 보장한다.

[0161] 선택적으로, 커맨드 데이터를 포함하는 제1 트랜잭션의 출력은 커맨드 데이터를 암호화하는 데 사용되는 암호화 키를 포함할 수 있다. 암호화된 키는 제2의 상이한 암호화 키로 암호화될 수 있다. 제2 암호화 키를 알고 있는 당사자는 암호화된 암호화 키를 복호화하고 그 후 암호화된 커맨드 데이터를 복호화할 수 있다. 예컨대, 마스터 노드와 같은 엔티티가 네트워크의 노드들로 그리고 이들로부터 송신되는 모든 커맨드 메시지들을 볼 수 있는 것이 유리할 수 있다. 제2 암호화 키는 제1 노드의 공개 키 및 상이한 노드, 예컨대, 마스터 노드의 공개 키에 기초하여 생성될 수 있다. 제1 노드의 공개 키는 제1 노드의 보증된 키 또는 트랜잭션 공개 키일 수 있다. 유사하게, 마스터 노드의 공개 키는 마스터 노드의 보증된 키 또는 트랜잭션 공개 키일 수 있다.

[0162] 암호화는 대칭적일 수 있다. 대칭 암호화는 인터넷을 통한 통신을 위해 HTTPS가 제공하는 것과 동일한 레벨의 프라이버시를 온체인 데이터에 대해 제공한다. 이를 위해, 마스터 노드는 로컬 IoT 네트워크의 노드들 간의 모든 정규 메시지들을 암호화하는 데 사용되는 비밀 암호화 키를 생성한다. 각각의 IoT 트랜잭션 OP_RETURN 페이로드(즉, 커맨드 데이터를 포함하는 출력)는 2개의 데이터 청크들을 가질 수 있다. 첫째로, 암호화 키가 요청 및 응답 디바이스들(종단 간)의 (보증된) 공개 키들을 사용하여 도출되는 BIE1 ECIES 암호화된 IoT 메시지. 둘째로, IoT 메시지에 대한 BIE1 ECIES 암호화된 암호화/복호화 키. ECIES(Elliptic Curve Integrated Encryption Scheme)는 디피-헬먼(Diffie-Hellman) 교환에 기초하는 암호화 체계이다. (이 데이터 푸시를 암호화하는 데 사용되는) 제2 암호화 키는 마스터 및 요청자 공개 키들을 사용하여 도출된다. 요청 노드와 마스터 간에 BIE1을 사용하여 자체 암호화된 IoT 메시지에 대한 복호화 키를 포함하는 암호화된 페이로드의 말미에 부가적인 바이트 푸시가 추가된다. 이는, 마스터 노드가 네트워크 상의 디바이스들 간에 전송된 복호화된 데이터를 볼 수 있다는 것을 보장한다. AES(American Encryption Standard) 암호화와 같은 다른 암호화 기술들이 사용될 수 있다.

[0163] 도 12a 및 도 12b는 암호화된 페이로드 데이터를 갖는 설명된 커맨드 및 응답 트랜잭션을 예시한다.

[0164] 위에서 언급된 바와 같이, 제1 및 제2 노드는 트랜잭션들을 생성 및 업데이트할 때 트랜잭션 공개 키들을 사용

할 수 있다. 공개 키들은 IoT 네트워크 상에서 노드들을 식별하는 데 사용되지만, 바람직하게는 이러한 공개 키들은 트랜잭션들에 서명하는데 사용되지 않아야 한다. 예컨대, 각각의 노드는 등록 기관에 의해 발행된 인증서에 포함된 공개 키를 가질 수 있다. 보증된 키들(예컨대, 마스터 노드에 의해 서명된 대응하는 인증서를 갖는 공개 키들)의 소유자들은 제3자들에게 자신의 아이덴티티들을 마스킹하고 트랜잭션들에 서명하는 데 사용될 수 있는 공유 비밀들을 도출할 수 있다.

[0165] 위에서 논의된 바와 같이, IoT 네트워크(501)는 마스터 노드(503a)를 포함할 수 있다. 마스터 노드(503a)는 시드 키를 획득(예컨대, 생성)하고, 그 후 시드 키에 각각 기초한 일 세트의(예컨대, 복수의) 개인 키들을 생성할 수 있다. 마스터 노드(503a)는 그 후 네트워크의 노드들(예컨대, 서버 및 슬레이브 노드들)에 개인 키들의 세트(아래에서 공동 개인 키들로서 지칭됨)를 송신할 수 있다. 각각의 노드는 동일한 세트의 공동 개인 키들을 수신하지만 시드 개인 키는 수신하지 않는다.

[0166] 각각의 노드는 개개의 메인 개인 키, 예컨대, 그 노드의 보증된 공개 키에 대응하는 개인 키를 갖는다. 마스터 노드(503a)를 포함하는 각각의 노드는 공동 개인 키들의 세트를 사용하여 대응하는 2차(또는 트랜잭션) 개인 키들의 세트를 생성한다. 트랜잭션 개인 키들은 각각의 공동 키에 개개의 노드의 메인 개인 키를 추가함으로써 생성된다. 각각의 노드에 대해, 대응하는 트랜잭션 공개 키들의 세트가 트랜잭션 개인 키들의 세트로부터 생성될 수 있다.

[0167] 제1 노드는 마스터 노드(503a), 즉 공동 개인 키들의 세트를 생성하는 노드일 수 있다. 대안적으로, 제1 노드는 마스터 노드로부터 공동 개인 키들의 세트를 수신하는 중개 노드(예컨대, 서버 또는 슬레이브 노드)일 수 있다. 일부 예들에서, 각각의 노드는 트랜잭션 공개 키를 한 번만 사용할 수 있다.

[0168] 보증된 키로부터 트랜잭션 키들을 생성하기 위한 예시적인 키-마스킹 알고리즘이 아래에 제공된다. 로컬 IoT 네트워크(501) 상의 노드들(503)은 모두 공개 키를 등록하는 인증서들을 발행받는다. 구체적으로, 서버 노드는 키 PK_{serv} 를 보증하였고 마스터 노드는 개인 키 sk_{Master} 로 인증 키 PK_{Master} 를 보증하였다.

[0169] 단계 1: 마스터 노드(503a)는 시드로부터 마스터 확장 개인 키 m^{joint} 를 생성한다. m은 IoT 네트워크(501)에 걸쳐 공유되고 각각의 노드 주소를 마스킹하는 데 사용되는 키들의 지갑을 생성하는 데 사용될 것이다. 공동 지갑에는 인덱싱된 키들을 갖는다:

[0170] $$sk_{i,j}^{joint}, \quad PK_{i,j}^{joint} = sk_{i,j}^{joint} \cdot G$$

[0171] 단계 2: 마스터 노드(503a)는 오프-체인 종단간 암호화 체계(예컨대, BIE1 ECIES)를 사용하여 오프-체인 메시지에서 IoT 네트워크 상의 다른 노드들인 노드들과 m을 공유한다.

[0172] 단계 3: 서버 노드(503b)가 m을 획득하면, 그것은 지갑에서 계층적 결정론적 키 쌍들의 세트를 도출할 수 있다.

[0173] 단계 4: 네트워크(501) 상의 각각의 노드(503)는 보증된 키 쌍으로부터의 개인 키를 공동 지갑으로부터 생성된 개인 키들에 추가함으로써 그의 지갑 개인 키들을 생성한다. 예컨대, 마스터 노드는 트랜잭션 서명 키들 $s_{i,j}^{M}$을 생성하며, 여기서

[0174] $sk_{i,j}^{M} = sk_{i,j}^{joint} + sk_{Master}$ 이다.

[0175] 단계 5: 각각의 노드(503)는 공동 지갑으로부터의 공개 키에 다른 노드 공개 키들을 추가함으로써 IoT 네트워크(501) 상의 다른 노드들(503)에 대한 모든 지불 엔드포인트들(주소들)을 식별할 수 있다. 예컨대, 서버 노드(503b)는 마스터 노드의 지불 주소 공개 키들 $PK_{i,j}^{M}$을 도출할 수 있으,며, 여기서

[0176] $PK_{i,j}^{M} = PK_{i,j}^{joint} + PK_{Master}$. 이다.

[0177] IoT 네트워크(501) 상의 각각의 노드(503)는 자체 지갑을 도출하고 이들이 관련 IoT 인증서들의 로케이션 및 m을 알고 있는 경우 다른 디바이스들의 주소들을 모니터링할 수 있다.

[0178] 암호화 및 키 마스킹 둘 모두는 로컬 IoT 네트워크(501) 상의 모든 노드들(503)에 대한 디바이스 가시성을 여전히 보장하면서, 디바이스 활동 데이터가 제3자들에게 누출되는 것을 방지한다.

[0179] IoT 네트워크(501)의 노드들(503)에 의해 송신된 각각의 트랜잭션은 커맨드 데이터를 포함하는 출력을

포함한다. 출력 및/또는 커맨드 데이터는 출력이 커맨드 데이터를 포함한다는 것을 표시하기 위한 프로토콜 플래그를 포함할 수 있다. 이는, IoT 디바이스들 및 독립적인 제3자들이 온-체인 커맨드, 동작 또는 상태 업데이트가 발생했을 때를 식별하는 것을 가능하게 한다.

[0180] 도 13은 제1 트랜잭션의 예시적인 커맨드 데이터 출력을 예시한다. 제1 트랜잭션은 제1 노드의 서명을 포함하는 입력(도시되지 않음) 및 커맨드 데이터를 포함하는 출력(1301)을 포함한다. 제1 트랜잭션은 또한 아래에서 논의될 제2 출력(도시되지 않음)을 포함할 수 있다. 이 예에서 프로토콜 식별자(4바이트) 다음에는 IoT 통신 정보를 포함하는 93바이트 페이로드가 이어진다. 통신 정보는 커맨드 명령의 의도된 수신자의 32바이트 디바이스 ID, 디바이스 인증서의 로케이션, 커맨드들 및 디바이스 상태를 포함한다. 일부 예들에서, 새로운 커맨드 또는 상태 업데이트를 발행하는 모든 각각의 트랜잭션은 이 포맷을 따라야 하거나 또는 그것은 무효 커맨드로 간주된다. 임의의 온-체인 메시지에 대해 필드가 필요하지 않은 경우, 그의 바이트들은 0x00000000으로 설정될 수 있다. 바람직하게는, 아래에서 논의되는 바와 같이, 페이로드 데이터 자체는 암호화될 것이다. 페이로드 데이터는 그 후 복호화 키를 보유한 당사자들에 의해서만 액세스될 수 있다. 아래 표는 예시적인 IOT 메시지 페이로드의 필드들을 설명한다.

표 2

[0181]

필드 크기(바이트)	설명		데이터 유형	코멘트들
4	IOT 프로토콜 식별자		uint32_t	IOT 프로토콜을 표시하는 프리픽스
1	페이로드 유형		uint16_t	메시지가 정규 IOT 메시지인지 아니면 인증서인지를 표시하는 1바이트 식별자
4	소프트웨어 버전 번호		uint32_t	IOT 버전 번호(프로토콜 업데이트들/업그레이드들을 위해 필요함)
32	디바이스 ID		char[32]	커맨드/메시지의 대상인 디바이스에 대한 고유 디바이스 ID
40 (32 + 4 + 4)	디바이스 인증서 로케이션	TXID	char[32]	디바이스 인증서를 포함하는 트랜잭션의 트랜잭션 ID
		VOUT	uint16_t	인증서 TX 내의 취소 UTXO 로케이션
		VOUT	uint16_t	인증서 페이로드의 출력 번호
4	커맨드/메시지		uint32_t	디바이스 ID를 갖는 디바이스에 지향되는 커맨드 또는 메시지를 인코딩하는 스트링
4	상태		uint16_t	현재 디바이스 상태
4	이전 상태		uint16_t	디바이스의 가장 최근 이전 상태

[0182] 디바이스 상태 복제본들은 디바이스의 보고된 상태 또는 원하는 상태의 논리적 표현이다. IoT 메시지 내에, 디바이스 상태 정보에는 디바이스 ID, 상태 및 이전 상태가 인코딩된다. 디바이스 ID와 관련된 최신 트랜잭션은 현재 디바이스 상태를 표현한다. 디바이스들의 상태와 관련된 커맨드들, 응답들 및 데이터를 포함하는 메시지들은 공개 키 암호화 및 작업 증명을 사용하여 보호되는 블록체인 상의 타임스탬핑된 블록들에 포함된다. 요약하면, IoT 네트워크(501) 상의 노드들(503)은 트랜잭션을 브로드캐스트하기 위해 블록체인 네트워크(106)에 연결된으로써 그리고 IoT 커맨드 데이터를 포함하는 트랜잭션들을 사용하여 직접 통신한다. 블록체인(150)은 IoT 네트워크 구성요소들로부터의 커맨드들 및 상태 업데이트들을 기록할 뿐만 아니라 IoT 디바이스들(502)과 관련된 보고서들 및 경고들을 발행하기 위한 영구 데이터 저장소로서 사용된다. 프로토콜은 다음 특징들 중 하나 이상을 사용할 수 있다.

[0183] 요청 및 응답 메시징 시스템 – 커맨드들을 수신하고 확인응답하기 위한 요청 및 응답 시스템이 사용된다. 요청들은 종단 디바이스에 의해 해석될 수 있는 IoT 로직을 인코딩하는 오프라인(피어 투 피어) 트랜잭션들이다. 응답들 또는 확인응답들은 블록체인 네트워크(106) 상의 트랜잭션 가시성으로부터 해석된다.

[0184] 오프라인 트랜잭션 전파 – 트랜잭션들 내에 인코딩된 명령들이 직접 전송된다(피어 투 피어). 마스터, 서버 및 슬레이브 노드들은 트랜잭션 서명을 검증함으로써 트랜잭션의 소스를 독립적으로 검증할 수 있다. 이는 또한 제어기들에 대한 지불 방법으로서 역할을 한다.

[0185] 노드들과 종단 디바이스들 간의 직접 통신 – 트랜잭션 커맨드 페이로드 내에 인코딩된 명령이 종단 디바이스를 위해 의도된 경우, 서버 또는 슬레이브 노드는 트랜잭션으로부터 명령을 추출하여 이를 종단 디바이스에 직접 통신할 수 있다.

[0186] 동작의 확인응답으로서 트랜잭션들의 브로드캐스팅 – 블록체인 네트워크에 트랜잭션을 브로드캐스트하는 것은

커맨드에 인코딩된 동작이 디바이스 ID를 가진 디바이스에 의해 수행되었음을 표시한다.

[0187] 디바이스 상태 및 이력을 인코딩하는 채굴된 트랜잭션들 – 블록체인은 완전한 디바이스 상태 및 이력을 저장하는 (논리적으로) 중앙화되고 물리적으로 분산된 데이터베이스로서 작용한다.

[0188] 실시예들은 다음의 유리한 특징들 중 하나 이상을 제공한다.

[0189] 근본적인 블록체인 인프라스트럭처의 보안 – 가치 전달을 인코딩하고 IoT 상호작용들을 로깅하는 모든 트랜잭션들은 공개 키 암호화 및 작업 증명을 사용하여 보호된다. secp256k1 파라미터들에 기초한 ECC(elliptic curve cryptography)는 IoT 노드들을 식별하는데 사용되는 공개/개인 키들을 보호하고 작업 증명은 IoT 네트워크 상태 및 이력을 레코드하는 블록체인을 보호한다.

[0190] 안전한 키 관리 및 난독화 – 키 난독화 기술들은 민감한 공개 키들이 그의 대응하는 개인 키들의 남용을 통해 취약해지지 않도록 보장하는데 사용된다. 키 난독화는 또한, IoT 솔루션 구성요소들이 그의 공개 주소들을 마스킹함으로써 프라이버시를 증가시키는 것을 가능하게 한다.

[0191] 암호화 – 포함된 디바이스 특정 데이터는 종단 간 암호화(예컨대, BIE1 또는 AES)되어서, 복호화 키를 갖는 IoT 노드들만이 액세스를 획득할 수 있다.

[0192] **예시적인 사용 사례**

[0193] 예시적인 사용 사례는 공공 도서관들을 위한 프린터 서비스이다. 대부분의 정부 지원 또는 대학 도서관들에서, 인쇄 비용은 시트당 최소 금액(예컨대, 6-10p)을 청구함으로써 지불된다. 현재(중앙화된) 모델 내에서, 사용자들은 도서관 관리에 의해 관리되는 계정들을 연다. 계정들은 선불로 입금될 필요가 있고 트랜잭션들은 도서관 운영 소프트웨어에 의해 관리되어야 해서, 도서관들에 큰 관리 부담을 초래한다. 본 개시는 퍼미셔닝 및 피어-투-피어 제어 프로토콜들의 결합된 사용으로 이 문제를 해결한다. 도 14는 P2P 인쇄를 위한 예시적인 IoT 네트워크(501)를 예시한다.

[0194] 1) 도서관 관리자는 마스터 노드(503a)(도서관 관리자에 의해 제어됨)를 설정하고 프린터들(종단 디바이스들)(502)을 직접 제어하는 슬레이브 노드들(503c)을 구성한다.

[0195] 2) 관리자는 슬레이브 노드들(503c) 및 종단 디바이스들(502)이 메시지들을 해석하는 데 사용할 규칙 엔진을 구성한다. 규칙 엔진은 하나 이상의 규칙들을 실행하는 시스템이다.

[0196] 3) 도서관 관리자는 슬레이브 노드(503c)를 구성한다. 이는 물리적 동작들을 수행하도록 프린터에 직접 지시할 수 있는 지불 수신기이다.

[0197] 4) 새로운 도서관 사용자들은 표준 등록/로그-인 방법들(예컨대, 하나 이상의 크리덴셜들, 예컨대, 사용자 이름 및 패스워드)을 사용하여 마스터 노드(503a)에 의해 퍼미셔닝된다. IoT 퍼미셔닝 알고리즘은 백-엔드로 수행된다. 도 15a는 새로운 도서관 사용자에게 인증서를 발행하기 위해 도서관 관리자에 의해 사용되는 예시하는 트랜잭션을 예시한다. 이 경우에, 도서관 관리자는 마스터 노드(503a) 및 인증서 취소 파워를 갖는 등록 기관이다.

[0198] 5) 사용자가 아이템을 인쇄하고자 할 때, 인쇄된 문서는 도서관 인트라넷 시스템 내에서 전송될 수 있다. 문서에는 커맨드 트랜잭션이 덧붙여질 것이다. 이 트랜잭션은 프린터, 프린터 디바이스 ID 및 SIGHASH_SINGLE 트랜잭션 서명을 조정하는 슬레이브 노드에 대한 지불을 포함할 것이다. 도 15b는 랩톱(서버)에 의해 (슬레이브) 제어기로 전송된 예시하는 커맨드 트랜잭션을 예시한다.

[0199] 6) 제어기(슬레이브 노드)는 트랜잭션이 유효한 블록체인 트랜잭션이고, 트랜잭션 소스(공개 키)가 시스템 상에서 퍼미셔닝되었고, 트랜잭션 값은 커맨드에 포함된 명령에 대한 비용을 지불하기에 충분하다는 것을 인증할 것이다.

[0200] 7) 모든 체크들이 통과되는 경우, 슬레이브 노드는 사용자에 의해 요청된 동작을 수행하도록 프린터에 지시한다.

[0201] 8) 슬레이브 노드는 지불을 자신의 주소에 잠그는 출력을 추가하고 SIGHASH_ALL를 갖는 서명을 추가하고, 그 후 트랜잭션을 블록체인 네트워크(106)에 브로드캐스트할 것이다. 도 15c는 제어기에 의해 블록체인 네트워크(106)로 브로드캐스트되는 예시적인 커맨드-확인응답 트랜잭션을 예시한다.

[0202] **결론**

[0203] 위의 실시예들은 단지 예로서만 설명되었다는 것이 인지될 것이다. 보다 일반적으로, 다음 스테이트먼트들 중 임의의 하나 이상에 따른 방법, 장치 또는 프로그램이 제공될 수 있다.

[0204] 스테이트먼트 1. 제1 네트워크에 참여하도록 요청자에게 허가를 부여하기 위한 컴퓨터-구현 방법으로서,

[0205] 제1 네트워크는 브리징 노드들의 세트 및 브리징 노드들의 세트 중 하나 이상에 의해 제어 가능한 디바이스들의 세트를 포함하고, 각각의 브리징 노드는 또한 블록체인 네트워크의 개개의 노드이고;

[0206] 방법은, 등록 기관에 의해 수행되고,

[0207] 제1 블록체인 트랜잭션을 생성하는 단계 - 제1 블록체인 트랜잭션은 등록 기관의 제1 공개 키에 링크된 서명을 포함하는 입력, 및 제1 인증서를 포함하는 제1 출력을 포함하고, 제1 인증서는 요청자에게 할당된 식별자를 포함함 - ; 및

[0208] 블록체인에의 포함을 위해 블록체인 네트워크로 제1 블록체인 트랜잭션을 송신하는 단계를 포함하는,

[0209] 컴퓨터-구현 방법.

[0210] 스테이트먼트 2. 스테이트먼트 1의 방법에 있어서,

[0211] 제1 트랜잭션은 등록 기관의 제2 공개 키에 잠긴 제2 출력을 포함하는,

[0212] 컴퓨터-구현 방법.

[0213] 스테이트먼트 3. 스테이트먼트 2의 방법에 있어서,

[0214] 제1 출력은 등록 기관의 제2 공개 키에 대해 시간-잠금되고, 시간 잠금은 미리 결정된 시간 기간 후까지 제1 출력이 잠금해제되는 것을 방지하는,

[0215] 컴퓨터-구현 방법.

[0216] 스테이트먼트 4. 스테이트먼트 2 또는 스테이트먼트 3의 방법에 있어서,

[0217] 제1 출력은 적어도 등록 기관의 제2 공개 키 및 상이한 공개 키에 잠기는,

[0218] 컴퓨터-구현 방법.

[0219] 상이한 공개 키는 네트워크의 노드의 공개 키이거나 네트워크의 부분이 아닌 제3자일 수 있다. 이는 인증서 취소는 다수의 서명들을 요구한다는 것을 의미한다.

[0220] 스테이트먼트 5. 스테이트먼트 1 내지 스테이트먼트 4 중 어느 하나의 방법에 있어서,

[0221] 제1 블록체인 트랜잭션의 트랜잭션 식별자를 허가 요청자에게 송신하는 단계를 포함하는,

[0222] 컴퓨터-구현 방법.

[0223] 스테이트먼트 6. 스테이트먼트 1 내지 스테이트먼트 5 중 어느 하나의 방법에 있어서,

[0224] 인증서는 암호화 키로 암호화되며, 암호화 키는 등록 기관에 의해 생성되는,

[0225] 컴퓨터-구현 방법.

[0226] 예컨대, 암호화 키는 등록 기관에 의해 생성된 난수일 수 있다.

[0227] 스테이트먼트 7. 스테이트먼트 1 내지 스테이트먼트 6 중 어느 하나의 방법에 있어서,

[0228] 네트워크에 참여하기 위한 요청자로부터의 요청을 수신하는 단계 - 요청은 하나 이상의 크리덴셜들을 포함함 - ; 및

[0229] 하나 이상의 크리덴셜들에 기초하여 요청을 유효성 검증하는 단계를 포함하고, 제1 블록체인 트랜잭션의 상기 생성은 요청이 유효한 것을 조건으로 하는,

[0230] 컴퓨터-구현 방법.

[0231] 하나 이상의 크리덴셜들은 예컨대, 제조 정보 및/또는 요청자의 IP 주소를 포함할 수 있다.

[0232] 스테이트먼트 8. 스테이트먼트 1 내지 스테이트먼트 7 중 어느 하나의 방법에 있어서,

[0233] 브리징 노드들의 세트는 마스터 노드 및 마스터 노드에 의해 제어 가능한 중개 노드들의 세트를 포함하고, 등록 기관은 마스터 노드인,

[0234] 컴퓨터-구현 방법.

[0235] 스테이트먼트 9. 스테이트먼트 1 내지 스테이트먼트 8 중 어느 하나의 방법에 있어서,

[0236] 브리징 노드들의 세트는 마스터 노드 및 마스터 노드에 의해 제어 가능한 중개 노드들의 세트를 포함하고, 요청자는 마스터 노드인,

[0237] 컴퓨터-구현 방법.

[0238] 스테이트먼트 10. 스테이트먼트 1 내지 스테이트먼트 9 중 어느 하나의 방법에 있어서,

[0239] 요청자는 블록체인 네트워크의 개개의 노드이고, 인증서는 허가 요청자에게 할당된 공개 키를 포함하는,

[0240] 컴퓨터-구현 방법.

[0241] 스테이트먼트 11. 스테이트먼트 1 내지 스테이트먼트 8 중 어느 하나의 방법에 있어서,

[0242] 요청자는 제1 네트워크의 하나 이상의 브리징 노드들에 의해 제어 가능한 디바이스이고, 방법은,

[0243] 인증서들의 세트를 요청자에게 송신하는 단계를 포함하고, 세트의 각각의 인증서는 노드들의 세트 중 개개의 하나에 송신된,

[0244] 컴퓨터-구현 방법.

[0245] 스테이트먼트 12. 스테이트먼트 1 내지 스테이트먼트 11 중 어느 하나의 방법에 있어서,

[0246] 브리징 노드들의 세트 중 하나 이상에 제1 인증서를 송신하는 단계를 포함하는,

[0247] 컴퓨터-구현 방법.

[0248] 스테이트먼트 13. 스테이트먼트 2 또는 이에 의존하는 임의의 스테이트먼트의 방법에 있어서,

[0249] 제2 블록체인 트랜잭션을 생성하는 단계 – 제2 블록체인 트랜잭션은 제1 트랜잭션의 제2 출력을 참조하는 입력을 포함하고 등록 기관의 제2 공개 키에 링크된 서명을 포함함 – ; 및

[0250] 블록체인에의 포함을 위해 블록체인 네트워크로 제2 블록체인 트랜잭션을 송신하는 단계를 포함하는,

[0251] 컴퓨터-구현 방법.

[0252] 스테이트먼트 14. 제1 네트워크에 참여하기 위한 허가를 요청하기 위한 컴퓨터-구현 방법으로서,

[0253] 제1 네트워크는 브리징 노드들의 세트 및 브리징 노드들의 세트 중 하나 이상에 의해 제어 가능한 디바이스들의 세트를 포함하고, 각각의 브리징 노드는 또한 블록체인 네트워크의 개개의 노드이고;

[0254] 방법은, 요청자에 의해 수행되고,

[0255] 제1 네트워크에 참여하기 위한 요청을 등록 기관에 송신하는 단계; 및

[0256] 제1 인증서를 획득하는 단계를 포함하고, 인증서는 등록 기관에 의해 발행되고 요청자에게 할당된 식별자를 포함하는,

[0257] 컴퓨터-구현 방법.

[0258] 스테이트먼트 15. 스테이트먼트 14의 방법에 있어서,

[0259] 상기 획득하는 단계는,

[0260] 제1 인증서를 포함하는 제1 블록체인 트랜잭션의 트랜잭션 식별자를 수신하는 단계; 및

[0261] 트랜잭션 식별자를 사용하여 블록체인으로부터 제1 블록체인 트랜잭션을 획득하는 단계를 포함하는,

[0262] 컴퓨터-구현 방법.

[0263] 스테이트먼트 16. 스테이트먼트 15의 방법에 있어서,

[0264] 제1 블록체인 트랜잭션은 인증서를 포함하는 제1 입력, 및 등록 기관의 공개 키에 링크된 제2 출력을 포함하고,

방법은,

[0265] 등록 기관의 공개 키를 식별하는 단계; 및

[0266] 등록 기관의 공개 키로부터 블록체인으로 송신된 하나 이상의 개개의 트랜잭션들에 포함된 하나 이상의 제2 인증서들을 식별하는 단계를 포함하고, 각각의 제2 인증서는 개개의 브리징 노드 또는 디바이스 또는 네트워크에 발행되는,

[0267] 컴퓨터-구현 방법.

[0268] 스테이트먼트 17. 스테이트먼트 16의 방법에 있어서,

[0269] 제1 인증서는 요청자의 공개 키를 포함하고, 제1 네트워크의 브리징 노드들의 세트의 개개의 하나에 발행된 각각의 제2 인증서는 상기 노드의 개개의 공개 키를 포함하고, 방법은,

[0270] 상기 브리징 노드들의 세트 중 적어도 하나에 제3 블록체인 트랜잭션을 송신하는 단계를 포함하고, 제3 블록체인 트랜잭션은 적어도 하나의 브리징 노드의 개개의 공개 키에 잠긴 출력을 포함하는,

[0271] 컴퓨터-구현 방법.

[0272] 스테이트먼트 18. 스테이트먼트 14 내지 스테이트먼트 17 중 어느 하나의 방법에 있어서,

[0273] 상기 제1 인증서를 획득하는 단계는 등록 기관으로부터 제1 인증서를 수신하는 단계를 포함하는,

[0274] 컴퓨터-구현 방법.

[0275] 스테이트먼트 19. 스테이트먼트 14 내지 스테이트먼트 18 중 어느 하나의 방법에 있어서,

[0276] 등록 기관으로부터 하나 이상의 제2 인증서들을 수신하는 단계를 포함하고, 각각의 제2 인증서는 제1 네트워크의 브리징 노드들 또는 디바이스들의 세트의 개개의 하나에 발행되는,

[0277] 컴퓨터-구현 방법.

[0278] 스테이트먼트 20. 스테이트먼트 19의 방법에 있어서,

[0279] 제1 인증서는 요청자의 네트워크 주소를 포함하고, 제1 네트워크의 개개의 브리징 노드에 발행된 각각의 제2 인증서는 상기 노드의 개개의 네트워크 주소를 포함하고, 방법은,

[0280] 상기 브리징 노드들의 세트 중 하나 이상에 메시지를 송신하는 단계를 포함하고, 메시지는 요청자의 네트워크 주소로부터, 메시지가 송신되는 하나 이상의 브리징 노드들의 개개의 네트워크 주소로 송신되는,

[0281] 컴퓨터-구현 방법.

[0282] 네트워크 주소는 IP 주소일 수 있다.

[0283] 스테이트먼트 21. 스테이트먼트 14 또는 스테이트먼트 18 내지 스테이트먼트 20 중 어느 하나의 방법에 있어서,

[0284] 요청자는 제1 네트워크의 디바이스들의 세트 중 하나인,

[0285] 컴퓨터-구현 방법.

[0286] 스테이트먼트 22. 스테이트먼트 14 내지 스테이트먼트 21 중 어느 하나의 방법에 있어서,

[0287] 요청자는 제1 네트워크의 노드들의 세트 중 하나인,

[0288] 컴퓨터-구현 방법.

[0289] 스테이트먼트 23. 스테이트먼트 22의 방법에 있어서,

[0290] 브리징 노드들의 세트는 마스터 노드 및 마스터 노드에 의해 제어 가능한 하나 이상의 중개 노드들을 포함하고, 요청자는 마스터 노드인,

[0291] 컴퓨터-구현 방법.

[0292] 스테이트먼트 24. 스테이트먼트 14 내지 스테이트먼트 23 중 어느 하나의 방법에 있어서,

[0293] 요청은 요청자의 하나 이상의 크리덴셜들을 포함하는,

[0294] 컴퓨터-구현 방법.

[0295] 스테이트먼트 25. 스테이트먼트 24의 방법에 있어서,

[0296] 하나 이상의 크리덴셜들은 요청자의 IP 주소를 포함하는,

[0297] 컴퓨터-구현 방법.

[0298] 스테이트먼트 26. 컴퓨터 장비로서,

[0299] 하나 이상의 메모리 유닛들을 포함하는 메모리; 및

[0300] 하나 이상의 프로세싱 유닛들을 포함하는 프로세싱 장치를 포함하고,

[0301] 메모리는 프로세싱 장치 상에서 실행되도록 배열된 코드를 저장하고, 코드는 프로세싱 장치 상에 있을 때, 스테이트먼트 1 내지 스테이트먼트 13 중 어느 하나의 방법을 수행하도록 구성되는,

[0302] 컴퓨터 장비.

[0303] 스테이트먼트 27. 컴퓨터-판독 가능 저장소 상에서 구체화되고 스테이트먼트 26의 컴퓨터 장비 상에서 실행될 때 스테이트먼트 1 내지 스테이트먼트 13 중 어느 한 항의 방법을 수행하도록 구성된 컴퓨터 프로그램.

[0304] 스테이트먼트 28. 컴퓨터 장비로서,

[0305] 하나 이상의 메모리 유닛들을 포함하는 메모리; 및

[0306] 하나 이상의 프로세싱 유닛들을 포함하는 프로세싱 장치를 포함하고,

[0307] 메모리는 프로세싱 장치 상에서 실행되도록 배열된 코드를 저장하고, 코드는 프로세싱 장치 상에 있을 때, 스테이트먼트 14 내지 스테이트먼트 25 중 어느 하나의 방법을 수행하도록 구성되는,

[0308] 컴퓨터 장비.

[0309] 스테이트먼트 29. 컴퓨터-판독 가능 저장소 상에서 구체화되고 제28 항의 컴퓨터 장비 상에서 실행될 때 스테이트먼트 14 내지 스테이트먼트 25 중 어느 한 항의 방법을 수행하도록 구성된 컴퓨터 프로그램.

[0310] 본원에서 개시된 교시들의 다른 양상에 따르면, 등록 기관 및 허가 요청자의 동작들을 포함하는 방법이 제공될 수 있다.

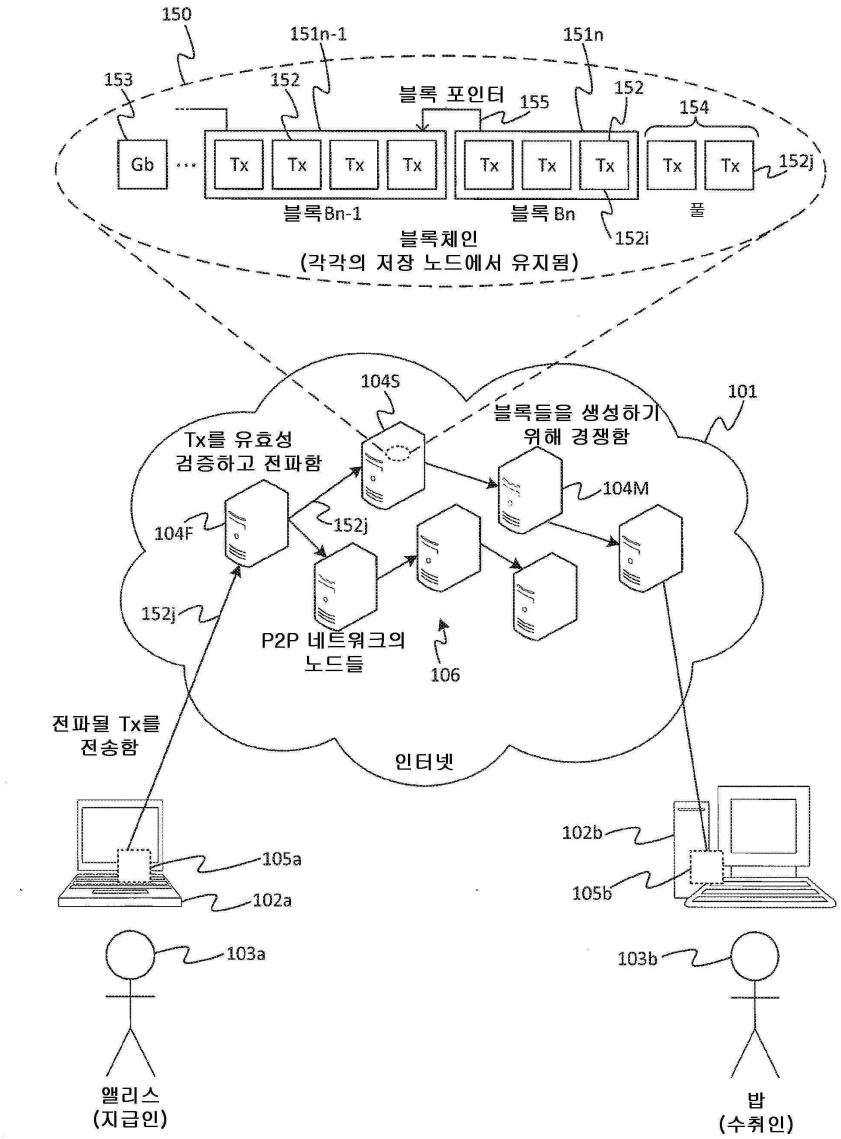
[0311] 본원에서 개시된 교시들의 다른 양상에 따르면, 등록 기관의 컴퓨터 장비 및 허가 요청자를 포함하는 시스템이 제공될 수 있다.

[0312] 본원에서 개시된 교시들의 다른 양상에 따르면, 제1 및 제2 블록체인 트랜잭션들을 포함하는 일 세트의 트랜잭션이 제공될 수 있다.

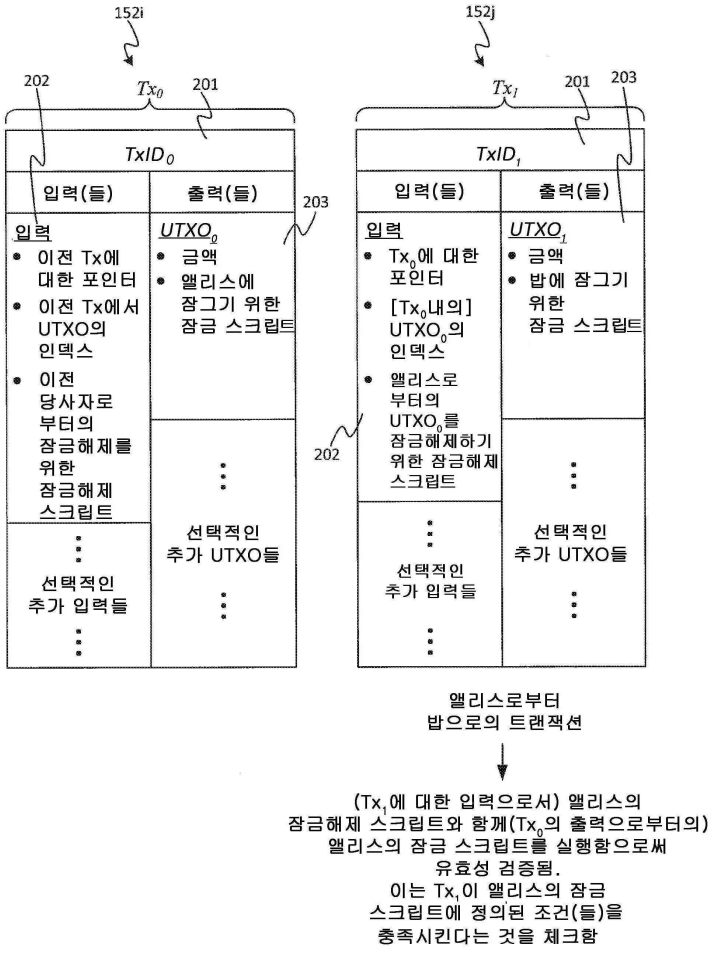
[0313] 개시된 기술들의 다른 변형들 또는 사용 사례들은 본원에서의 개시가 주어지면 당업자에게 명백해질 수 있다. 본 개시의 범위는 설명된 실시예에 의해 제한되는 것이 아니라 첨부된 스테이트먼트들에 의해서만 제한된다.

도면

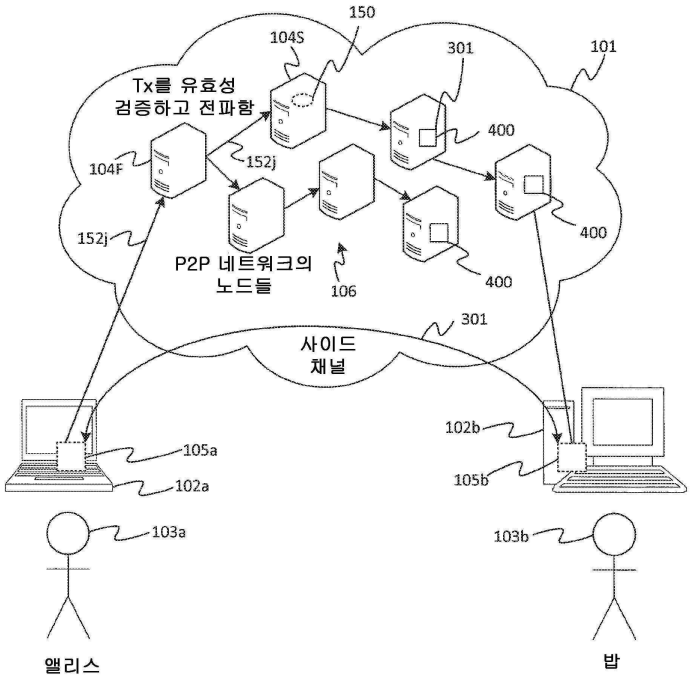
도면1


150
151n-1
151n
153
152
블록 포인터
155
152
154
Gb
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
Tx
152j
블록Bn-1
블록 Bn
풀
152i
블록체인
(각각의 저장 노드에서 유지됨)
104S
101
Tx를 유효성
검증하고 전파함
블록들을 생성하기
위해 경쟁함
104M
104F
152j
152j
P2P 네트워크의
노드들
106
전파될 Tx를
전송함
인터넷
102b
105a
105b
102a
103a
103b
앨리스
(지급인)
밥
(수취인)

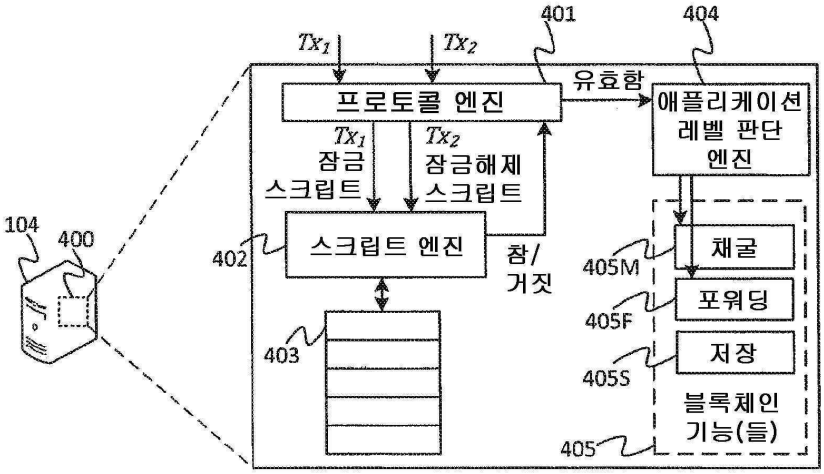
도면2


152i
202
Tx_0
201
$TxID_0$
입력(들)
출력(들)
203
입력
• 이전 Tx에 대한 포인터
• 이전 Tx에서 UTXO의 인덱스
• 이전 당사자로부터의 잠금해제를 위한 잠금해제 스크립트
$UTXO_0$
• 금액
• 앨리스에 잠그기 위한 잠금 스크립트
선택적인 추가 입력들
선택적인 추가 UTXO들
202
152j
Tx_1
201
203
$TxID_1$
입력(들)
출력(들)
입력
• Tx_0에 대한 포인터
• [Tx_0내의] $UTXO_0$의 인덱스
• 앨리스로부터의 $UTXO_0$를 잠금해제하기 위한 잠금해제 스크립트
$UTXO_1$
• 금액
• 밥에 잠그기 위한 잠금 스크립트
선택적인 추가 입력들
선택적인 추가 UTXO들
앨리스로부터 밥으로의 트랜잭션
(Tx_1에 대한 입력으로서) 앨리스의 잠금해제 스크립트와 함께(Tx_0의 출력으로부터의) 앨리스의 잠금 스크립트를 실행함으로써 유효성 검증됨.
이는 Tx_1이 앨리스의 잠금 스크립트에 정의된 조건(들)을 충족시킨다는 것을 체크함

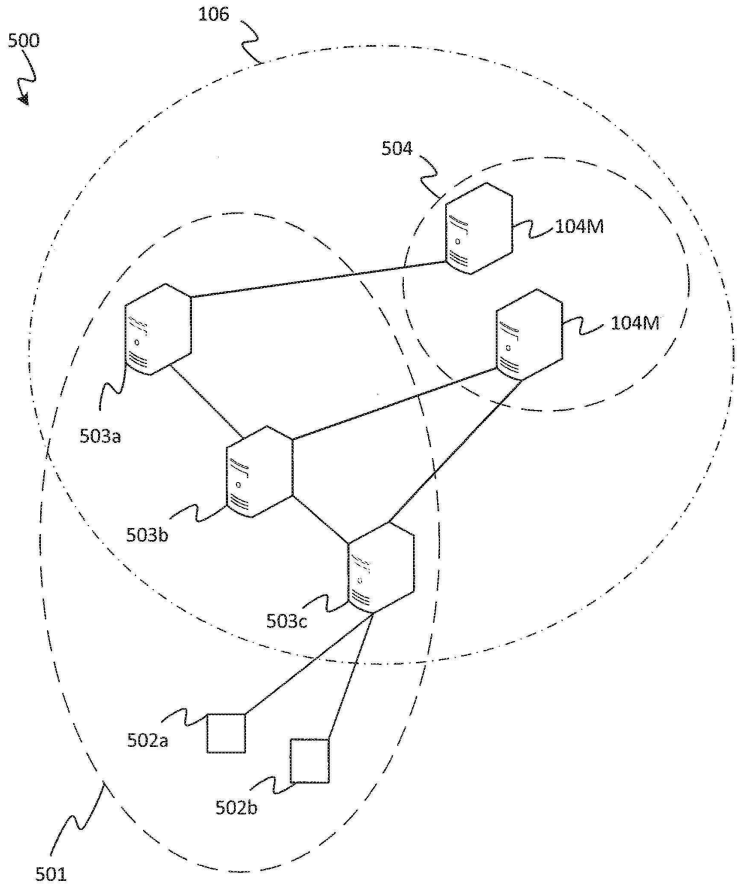
도면3


150
104S
301
101
Tx를 유효성 검증하고 전파함
400
104F
152j
152j
400
400
P2P 네트워크의 노드들
106
301
사이드 채널
102b
105a
105b
102a
103a
103b
앨리스
밥

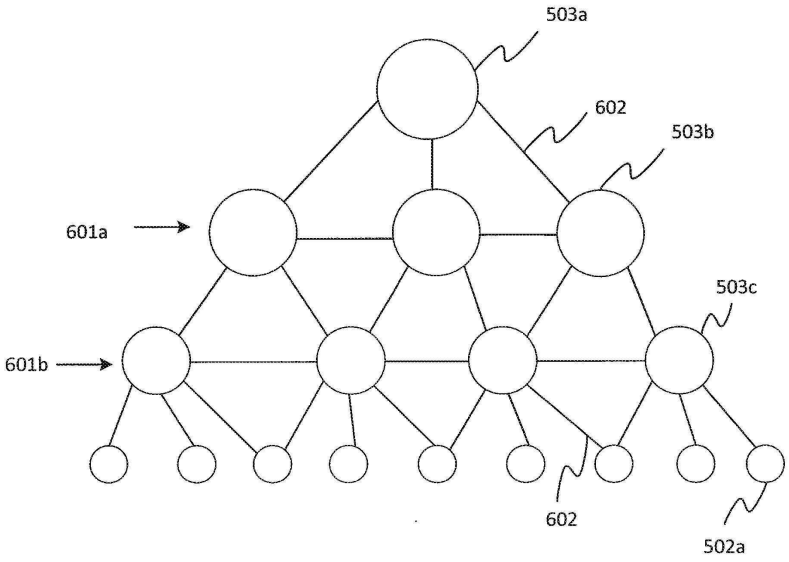
도면4


401
404
Tx1
Tx2
프로토콜 엔진
유효함
애플리케이션 레벨 판단 엔진
Tx1 잠금 스크립트
Tx2 잠금해제 스크립트
104
400
402
스크립트 엔진
참/거짓
403
405M
채굴
405F
포워딩
405S
저장
블록체인 기능(들)
405

도면5


106
500
504
104M
104M
503a
503b
503c
502a
502b
501

도면6



도면7a

Tx_1 (부분적)			
입력들		출력들	
값		값	
x	$<Sig_{PK0}>$ $<PK_0>$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >

701a

702a

도면7b

Tx_1 (완료)			
입력들		출력들	
값		값	
x	$< Sig_{PK0} > < PK_0 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
> 0	$< Sig_{PK1} > < PK_1 >$	x	OP_DUP OP_HASH160 $< H_{160}(PK_1) >$ OP_EQUALVERIFY OP_CHECKSIG

701b

702b

도면8a

Tx_1 (부분적)			
입력들		출력들	
값		값	
x	$< Sig_{PK0} > < PK_0 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
		$x + \delta$	OP_DUP OP_HASH160 $< H_{160}(PK_1) >$ OP_EQUALVERIFY OP_CHECKSIG

801a

802b

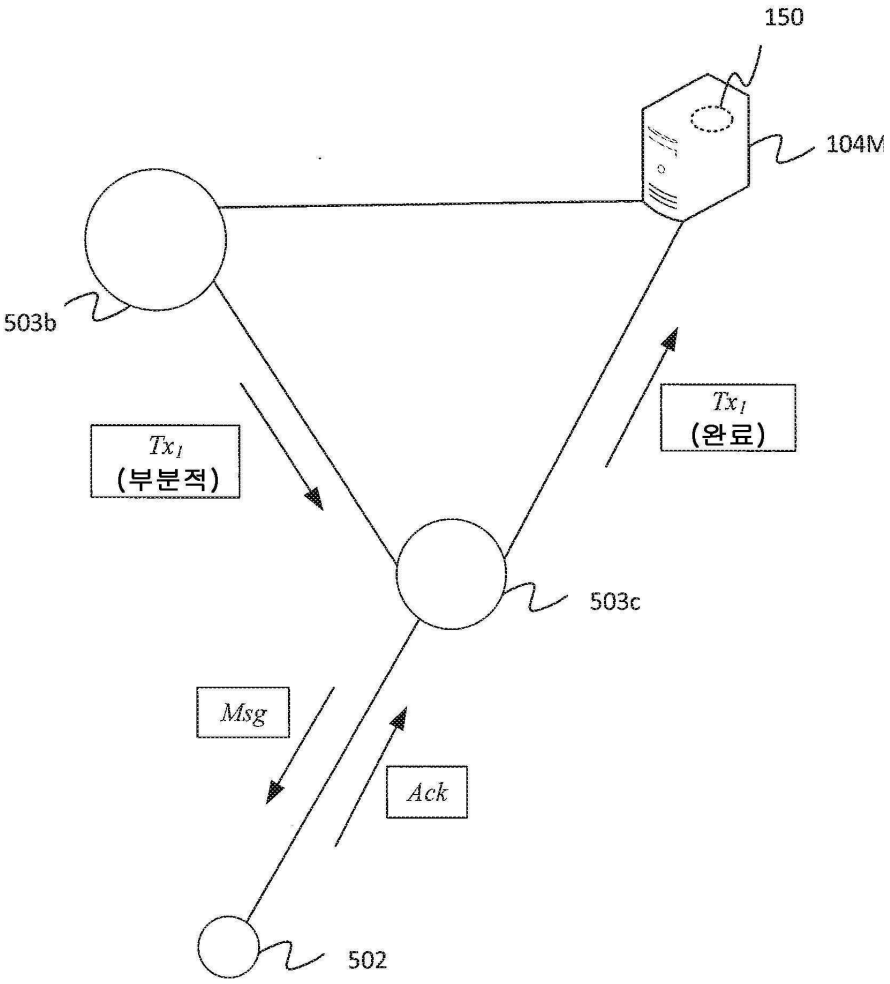
802a

도면8b

Tx_1 (완료)			
입력들		출력들	
값		값	
x	$< Sig_{PK0} > < PK_0 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
$> \delta$	$< Sig_{PK1} > < PK_1 >$	x	OP_DUP OP_HASH160 $< H_{160}(PK_1) >$ OP_EQUALVERIFY OP_CHECKSIG

801b

도면9



도면10a

Tx_1 (부분적)			
입력들		출력들	
값		값	
x	$< Sig_{PKserv} > < PK_{serv} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
		$x + \delta$	OP_DUP OP_HASH160 $< H_{160}(PK_{slave}) >$ OP_EQUALVERIFY OP_CHECKSIG

도면10b

Tx_1 (완료)			
입력들		출력들	
값		값	
x	$< Sig_{PKserv} > < PK_{serv} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
$> \delta$	$< Sig_{PKslave} > < PK_{slave} >$	x	OP_DUP OP_HASH160 $< H_{160}(PK_{slave}) >$ OP_EQUALVERIFY OP_CHECKSIG

도면11a

Tx_1 (요청)			
입력들		출력들	
값		값	
x_1	$< Sig_{PK0} > < PK_0 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
		y_1	OP_2 OP $< PK_1 > < PK_2 >$ OP_CHECKMULTISIG

1101a 1102b 1102a

도면11b

Tx_2 (승인)			
입력들		출력들	
값		값	
x_2	$< Sig_{PK1} > < PK_1 >$ $< Sig_{PK2} > < PK_2 >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Command data* >
		y_2	OP_DUP OP_HASH160 $< H_{160}(PK_3) >$ OP_EQUALVERIFY OP_CHECKSIG

1101b

1102c

도면12a

Tx_1 (부분적)			
입력들		출력들	
값		값	
x	$< Sig_{PKserv\_0} > < PK_{serv\_0} >$	0	OP_FALSE OP_RETURN 0x4d494f54 OP_PUSHDATA1 [*Payload length*] <*BIE1 Encrypted command data*> OP_PUSHDATA1 [*Payload length*] < *BIE1 Encrypted Encryption key* >
		$x + \delta$	OP_DUP OP_HASH160 $< H_{160}(PK_{slave\_0}) >$ OP_EQUALVERIFY OP_CHECKSIG

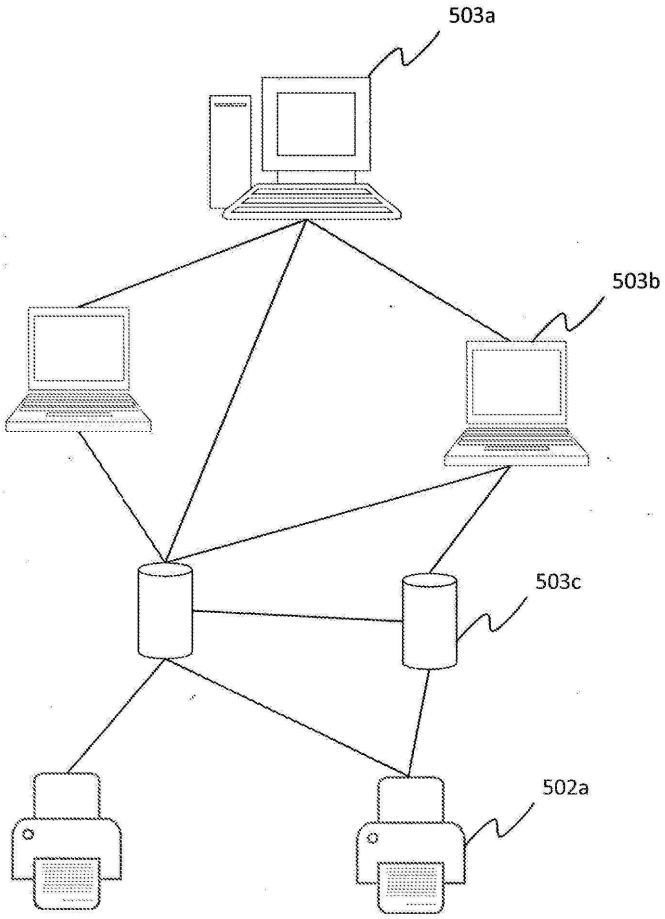
도면12b

Tx_1 (완료)			
입력들		출력들	
값		값	
x	$< Sig_{PKserv\_0} > < PK_{serv\_0} >$	0	OP_FALSE OP_RETURN 0x4d494f54 OP_PUSHDATA1 [*Payload length*] <*BIE1 Encrypted command data*> OP_PUSHDATA1 [*Payload length*] < *BIE1 Encrypted Encryption key* >
$> \delta$	$< Sig_{PKslave\_1} > < PK_{slave\_1} >$	$x + \delta$	OP_DUP OP_HASH160 $< H_{160}(PK_{slave\_0}) >$ OP_EQUALVERIFY OP_CHECKSIG

도면13

출력들	
값	
0	OP_FALSE OP_RETURN OP_PUSHDATA1 0x4d494f54 – *IoT*프로토콜 식별자 0x01 – 페이로드 유형 0x00000001 – *IoT* 소프트웨어 버전 번호 0x3dd5dfac...32 – 디바이스 ID 0x234a3789...22 – 디바이스 공개키 0x4d348912...87 – 디바이스 보증서 로케이션 데이터 0x3ad21fac – 커맨드 0x5665b456 – 상태 0x5665b456 – 이전 상태

도면14



도면15a

Tx_1 (인증서)			
입력들		출력들	
값		값	
x_1	$< Sig_{PKadmin} > < PK_{admin} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Certificate data* >
		y_1	OP_DUP OP_HASH160 $< H_{160}(PK_{admin}) >$ OP_EQUALVERIFY OP_CHECKSIG

도면15b

Tx_1 (부분적)			
입력들		출력들	
값		값	
x_2	$< Sig_{PKlaptop\_0} > < PK_{laptop\_0} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Encrypted Command data* >
		y_2	OP_DUP OP_HASH160 $< H_{160}(PK_{controller\_0}) >$ OP_EQUALVERIFY OP_CHECKSIG

도면15c

Tx_1 (제어기)			
입력들		출력들	
값		값	
x_2	$< Sig_{PKlaptop\_0} > < PK_{laptop\_0} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Encrypted Command data* >
x_3	$< Sig_{PKcontroller\_0} >$ $< PK_{controller\_0} >$	y_2	OP_DUP OP_HASH160 $< H_{160}(PK_{controller\_0}) >$ OP_EQUALVERIFY OP_CHECKSIG

도면16a

Tx_1 (인증서)			
입력들		출력들	
값		값	
x_1	$< Sig_{PKreg} > < PK_{reg} >$	0	OP_FALSE OP_RETURN 0x4d494f54 < *Certificate data* >
		y_1	OP_DUP OP_HASH160 $< H_{160}(PK_{reg}) >$ OP_EQUALVERIFY OP_CHECKSIG

1501a 1502b 1502a

도면16b

출력들	
값	
0	OP_FALSE OP_RETURN OP_PUSHDATA1 <페이로드 길이> 0x4d494f54 – *IoT* 프로토콜 식별자 0x02 – 페이로드 유형 0x00000001 – *IoT* 소프트웨어 버전 번호 0x3dd5dfac...32 – 새로운 디바이스ID 0x234a3789...22 – 새로운 디바이스 압축 공개키 0x3ad21fac – 디바이스 유형 0x00000004 – 디바이스 IoT 노드 유형 0x0000...06f – *IPv6* 주소 및 포트 번호 0x5664b456 – *UNIX* 시간 생성 날짜 0x5665b456 – *UNIX* 시간 인증서 만료 날짜 0x76d5335c.....45 – 부가적인 정보