(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2017년09월27일
(11) 등록번호 10-1781583
(24) 등록일자 2017년09월19일

(51) 국제특허분류(Int. Cl.)
G06F 17/30 (2006.01)
(52) CPC특허분류
G06F 17/30106 (2013.01)
G06F 17/3007 (2013.01)
(21) 출원번호 10-2016-0111914
(22) 출원일자 2016년08월31일
심사청구일자 2016년08월31일
(56) 선행기술조사문헌
US20160027229 A1*
US20150332283 A1*
KR1020160095720 A*
KR101637854 B1*
*는 심사관에 의하여 인용된 문헌

(73) 특허권자
서강대학교산학협력단
서울특별시 마포구 백범로 35 (신수동, 서강대학교)
(72) 발명자
장주욱
서울특별시 마포구 염리동 106동 401호
정문용
울산광역시 동구 봉수로 370 104동 2003호 (전하동, 현대홈타운아파트)
(74) 대리인
이지연

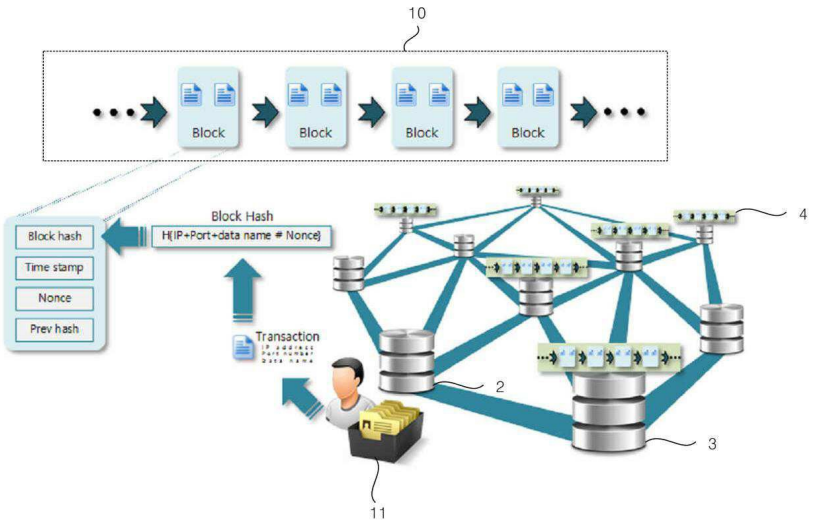
전체 청구항 수 : 총 15 항 심사관 : 김경완

(54) 발명의 명칭 **블록체인을 기반으로 한 파일 관리/검색 시스템 및 파일 관리/검색 방법**

(57) 요 약

본 발명은 데이터 명칭 및 소유주 확인이 가능한 블록체인을 기반으로 한 파일 관리 및 검색 시스템에 관한 것이다. 상기 파일 관리 및 검색 시스템은, 생성한 데이터를 저장하고 데이터 명칭을 상기 데이터가 저장된 노드의 IP address와 port number의 정보와 함께 트랜잭션 형태로 다른 노드에 전달하면, 이를 수신한 노드들은 해당 트랜잭션을 기반으로 블록을 생성하여 블록체인에 연결함으로써, 데이터 이름과 해당 데이터를 생성한 소유주 정보를 모든 사람들이 공유하게 된다. 이 정보는 블록체인으로 기록이 되어 있기 때문에 모든 노드가 같은 정보를 공유하여서 악의적 노드가 임의로 내용을 수정하기 위해서는 모든 노드가 가지고 있는 블록체인의 내용을 수정해야 하므로 임의적 수정이 사실상 불가능하기 때문에 블록체인 자체로도 보안성이 뛰어나다.

대 표 도 - 도1



(52) CPC특허분류
G06F 17/30091 (2013.01)

이 발명을 지원한 국가연구개발사업
과제고유번호 B0126161051
부처명 미래창조과학부
연구관리전문기관 정보통신기술진흥센터(IITP)
연구사업명 정보통신·방송 연구개발사업
연구과제명 초연결 자율형 IoT 서비스 지향 네트워크 인프라 기술 연구
기 여 율 1/1
주관기관 서강대학교 산학협력단
연구기간 2015.03.01 ~ 2018.02.28

명 세 서

청구범위

청구항 1

블록 체인(Block chain)을 활용하여 다수의 노드들에 저장된 데이터 또는 파일을 관리 및 검색할 수 있도록 하는 파일 관리 및 검색 시스템에 있어서,

각 노드들은,

적어도 블록 해쉬(block hash)값 및 논스(Nonce) 값을 포함하는 블록들이 연결되어 구성된 블록 체인;

상기 블록 체인을 갖는 모든 노드들에 대한 IP 주소 및 Port 번호를 저장한 IP 리스트;

사전에 설정된 폴더에 데이터 또는 파일이 저장되거나 삭제되면, 상기 데이터에 대한 정보 및 소유주 정보를 포함하는 트랜잭션을 생성하고, 상기 생성된 트랜잭션을 블록 체인을 갖는 다른 노드들로 전송하는 트랜잭션 전송 모듈; 및

블록 체인을 갖는 다른 노드로부터 트랜잭션을 수신하면, 수신된 트랜잭션에 대해 가장 먼저 블록을 생성하면 블록 해쉬값과 논스값을 다른 노드들에게 송신하며, 다른 노드로부터 블록 해쉬값과 논스값을 수신하면 수신된 정보를 이용하여 블록을 생성하여 블록 체인에 연결시키는 블록 체인 실행 모듈; 을 구비하고,

상기 블록 체인의 각 블록은 다수의 노드들에 저장된 데이터 또는 파일에 대하여 각각 생성된 것으로서, 각 블록의 블록 해쉬값은 데이터에 대한 정보, 데이터의 소유주 정보와 논스값에 대하여 사전 설정된 해쉬 알고리즘을 적용하여 생성한 해쉬값인 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 시스템.

청구항 2

제1항에 있어서, 상기 트랜잭션의 데이터에 대한 정보는 데이터 명칭을 포함하며,

상기 트랜잭션의 데이터의 소유주 정보는 해당 데이터가 저장된 노드의 IP 주소와 Port 번호를 포함하는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 시스템.

청구항 3

제1항에 있어서, 상기 트랜잭션 전송 모듈은

사전에 설정된 폴더내에 데이터 또는 파일이 저장되거나 삭제되면, 데이터 명칭을 설정하고,

상기 설정된 데이터 명칭 및 해당 노드의 IP 주소와 Port 번호를 포함하는 트랜잭션을 생성하고,

상기 생성된 트랜잭션을 블록 체인을 갖는 다른 노드들로 전송하는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 시스템.

청구항 4

제3항에 있어서, 상기 데이터 명칭은 아스키 코드(ASCII Code)의 형태로 변환되어 트랜잭션에 포함되고,

상기 IP 주소 및 Port 번호는 십진수의 형태로 변환되어 트랜잭션에 포함되는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 시스템.

청구항 5

제3항에 있어서, 상기 트랜잭션 전송 모듈은,

사전에 설정된 폴더내에 데이터 또는 파일이 저장되면, 데이터 생성을 나타내는 제1 플래그를 데이터에 대한 정보에 포함시켜 트랜잭션에 추가시키고,

사전에 설정된 폴더내의 데이터 또는 파일이 삭제되면, 데이터 삭제를 나타내는 제2 플래그를 데이터에 대한 정보에 포함시켜 트랜잭션에 추가시키는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 시스템.

청구항 6

제1항에 있어서, 블록 체인 실행 모듈은

다른 노드로부터 트랜잭션을 수신하면, 수신된 트랜잭션의 정보들에 대하여 proof of work를 실행하여 블록 해쉬값과 논스(Nonce)값을 생성하고, 생성된 블록 해쉬값과 논스값을 다른 노드들로 전송하며,

다른 노드로부터 블록 해쉬값과 논스값을 수신하면, 수신된 블록 해쉬값과 논스값을 이용하여 상기 수신된 트랜잭션에 대한 유효성을 검사하여 인증되면 블록을 생성하고,

상기 생성된 블록을 블록 체인에 연결시키는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 시스템.

청구항 7

제1항에 있어서, 상기 파일 관리 및 검색 시스템은 블록 체인을 이용하여 각 노드에 저장된 데이터와 소유주를 검색하는 파일 검색 모듈을 더 구비하고,

상기 파일 검색 모듈은,

검색하고자 하는 데이터 명칭이 입력되면,

블록 체인에 저장된 각 블록의 논스(nonce) 값을 판독하고, 상기 판독된 논스값, 상기 입력된 데이터 명칭 및 IP 리스트의 각 노드들에 대한 IP 주소와 Port 번호를 이용하여 해쉬값을 생성하고, 상기 생성된 해쉬값이 상기 논스값이 포함된 블록의 블록 해쉬값과 일치하는지를 판단하고,

상기 생성된 해쉬값과 일치하는 블록 해쉬값을 갖는 블록이 존재하면, 해당 IP 주소와 Port 번호를 갖는 노드가 소유주임을 결정하고,

상기 생성된 해쉬값과 일치하는 블록 해쉬값을 갖는 블록이 존재하지 않으면, 해당 데이터가 없음을 결정하는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 시스템.

청구항 8

제1항에 있어서, 상기 블록 체인 실행 모듈은,

블록 체인을 갖는 다른 노드들로부터 하나의 트랜잭션을 수신하면, 수신된 하나의 트랜잭션에 대해 하나의 블록을 생성하여 블록 체인에 연결시키는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 시스템.

청구항 9

블록 체인(Block chain)을 활용하여 다수의 노드들에 저장된 데이터를 관리 및 검색할 수 있도록 하기 위한 각 노드들에서의 파일 관리 및 검색 방법에 있어서,

(a) 적어도 블록의 해쉬(block hash)값 및 논스(Nonce) 값을 포함하는 블록들로 이루어진 블록 체인을 저장하는 단계;

(b) 상기 블록 체인을 갖는 모든 노드들에 대한 IP 주소 및 Port 번호를 저장한 IP 리스트를 구비하는 단계;

(c) 사전에 설정된 폴더에 데이터 또는 파일이 저장되거나 삭제되면, 상기 데이터에 대한 정보 및 소유주 정보를 포함하는 트랜잭션을 생성하고, 상기 생성된 트랜잭션을 블록 체인을 갖는 다른 노드들로 전송하는 단계; 및

(d) 블록 체인을 갖는 다른 노드들로부터 트랜잭션을 수신하면, 수신된 트랜잭션에 대한 블록을 생성하여 블록 체인에 연결시키는 단계; 을 구비하고,

상기 블록 체인의 각 블록은 다수의 노드들에 저장된 데이터 파일에 대하여 각각 생성된 것으로서, 각 블록의 블록 해쉬값은 데이터에 대한 정보, 데이터의 소유주 정보와 논스값에 대하여 사전 설정된 해쉬 알고리즘을 적용하여 생성한 해쉬값인 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 방법.

청구항 10

제9항에 있어서, 상기 트랜잭션의 데이터에 대한 정보는 데이터 명칭을 포함하며,

상기 트랜잭션의 데이터의 소유주 정보는 해당 데이터가 저장된 노드의 IP 주소와 Port 번호를 포함하는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 방법.

청구항 11

제9항에 있어서, 상기 (c) 단계는,

사전에 설정된 폴더내에 데이터 또는 파일이 저장되거나 삭제되면, 데이터 명칭을 설정하고,

상기 설정된 데이터 명칭 및 해당 노드의 IP 주소와 Port 번호, 전자 서명을 포함하는 트랜잭션을 생성하고,

상기 생성된 트랜잭션을 블록 체인을 갖는 다른 노드들로 전송하는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 방법.

청구항 12

제11항에 있어서, 상기 데이터 명칭은 아스키 코드(ASCII Code)의 형태로 변환되어 트랜잭션에 포함되고,

상기 IP 주소 및 Port 번호는 십진수의 형태로 변환되어 트랜잭션에 포함되는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 방법.

청구항 13

제9항에 있어서, 상기 (c) 단계는,

사전에 설정된 폴더내에 데이터 또는 파일이 저장되면, 데이터 생성을 나타내는 제1 플래그를 데이터에 대한 정보에 포함시켜 트랜잭션에 추가시키고,

사전에 설정된 폴더내의 데이터 또는 파일이 삭제되면, 데이터 삭제를 나타내는 제2 플래그를 데이터에 대한 정보에 포함시켜 트랜잭션에 추가시키는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 방법.

청구항 14

제9항에 있어서, 상기 (d) 단계는,

다른 노드로부터 트랜잭션을 수신하면, 수신된 트랜잭션의 정보들에 대하여 proof of work 를 수행하여 모든 노드들 중 가장 먼저 블록 해쉬값과 논스(Nonce)값을 생성하면, 이를 이용하여 블록을 생성하며, 상기 생성된 블록 해쉬값과 논스값을 다른 노드들로 전송하며,

다른 노드로부터 블록 해쉬값과 논스값을 수신하면, 수신된 블록 해쉬값과 논스값을 이용하여 상기 수신된 트랜잭션에 대한 유효성을 검사하여 인증되면 블록을 생성하고,

상기 생성된 블록을 블록 체인에 연결시키는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 방법.

청구항 15

제9항에 있어서, 상기 파일 관리 및 검색 방법은 (e) 외부로부터 요청된 데이터 명칭을 검색하여 제공하는 파일 검색 단계를 더 구비하고,

상기 (e) 파일 검색 단계는,

검색하고자 하는 데이터 명칭이 입력되면,

블록 체인에 저장된 각 블록의 논스(nonce) 값을 판독하고, 상기 판독된 논스값, 상기 입력된 데이터 명칭 및 IP 리스트의 각 노드들에 대한 IP 주소와 Port 번호를 이용하여 해쉬값을 생성하고, 상기 생성된 해쉬값이 상기 논스값이 포함된 블록의 블록 해쉬값과 일치하는지를 판단하고,

상기 생성된 해쉬값과 일치하는 블록 해쉬값을 갖는 블록이 존재하면, 해당 IP 주소와 Port 번호를 갖는 노드가 소유주임을 결정하고,

상기 생성된 해쉬값과 일치하는 블록 해쉬값을 갖는 블록이 존재하지 않으면, 해당 데이터가 없음을 결정하는 것을 특징으로 하는 블록 체인을 기반으로 한 파일 관리 및 검색 방법.

발명의 설명

기 술 분 야

[0001] 본 발명은 블록 체인을 기반으로 한 파일 관리/검색 시스템에 관한 것으로서, 더욱 구체적으로는 블록 체인을 기반으로 하여 데이터 명칭과 데이터의 소유주를 검색하여 제공할 수 있는 파일 관리/검색 시스템 및 파일 관리/검색 방법에 관한 것이다.

배 경 기 술

[0002] 블록 체인(Block chain)은 공공 거래 장부라고도 부르며 가상 화폐로 거래할 때 발생할 수 있는 해킹을 막는 기술이다. 기존 금융 회사의 경우 중앙 집중형 서버에 거래 기록을 보관하는 반면, 블록체인은 거래에 참여하는 모든 사용자에게 거래 내역을 보내 주며 거래 때마다 이를 대조해 데이터 위조를 막는 방식을 사용한다. 블록체인은 대표적인 온라인 가상 화폐인 비트코인에 적용되어 있다. 비트코인은 누구나 열람할 수 있는 장부에 거래 내역을 투명하게 기록하며, 비트코인을 사용하는 여러 컴퓨터가 10분에 한 번씩 이 기록을 검증하여 해킹을 막는다.

[0003] 한편, 종래의 일반적인 파일 공유 시스템은 중앙 집중형 서버를 이용한 클라우드 storage 에 데이터 또는 파일들을 저장 및 관리하며, 사용자들이 클라우드 storage 에 접속하여 데이터 또는 파일을 요청하거나 검색하게 된다. 이 경우 대용량 저장 장치가 요구될 뿐만 아니라, 이들을 관리하기 위한 중앙 집중형 서버가 요구되므로, 시스템 설계 및 유지가 복잡해지는 문제점이 발생한다.

[0004] 본 발명은 블록 체인을 활용하여 중앙 집중형 서버없이 데이터 관리 및 요청시스템을 구현하기 위한 플랫폼을 제안하고자 한다.

선행기술문헌

특허문헌

[0005] (특허문헌 0001) 한국등록특허공보 제 10-1628009호

(특허문헌 0002) 한국공개특허공보 제 10-2010-0002784호

(특허문헌 0003) 한국등록특허공보 제 10-1344465호

발명의 내용

해결하려는 과제

[0006] 전술한 문제점을 해결하기 위한 본 발명의 목적은 블록 체인(Block chain)을 활용하여 각 노드들에 저장된 데이터 파일들에 대한 정보를 블록 체인의 형태로 저장 및 관리하고, 각 데이터 파일에 대한 소유주를 확인하고 검색할 수 있도록 하는 블록 체인을 기반으로 한 파일 관리 및 검색 시스템을 제공하는 것이다.

[0007] 본 발명의 다른 목적은 전술한 파일 관리 및 검색 시스템에 사용되는 파일 관리 및 검색 방법을 제공하는 것이다.

과제의 해결 수단

[0008] 전술한 기술적 과제를 달성하기 위한 본 발명의 제1 특징에 따른 블록 체인(Block chain)을 활용하여 다수의 노드들에 저장된 데이터 또는 파일을 관리 및 검색할 수 있도록 하는 파일 관리 및 검색 시스템에 있어서,

[0009] 각 노드들은, 현재 블록의 해쉬(block hash)값, 논스(Nonce) 값, 이전 블록의 해쉬값 및 Time stamp로 이루어진 블록들이 연결되어 구성된 블록 체인; 상기 블록 체인을 갖는 모든 노드들에 대한 IP 주소 및 Port 번호를 저장한 IP 리스트; 사전에 설정된 폴더에 데이터가 저장되거나 삭제되면, 상기 데이터에 대한 정보, 소유주 정보 및 전자 서명을 포함하는 트랜잭션을 생성하고, 상기 생성된 트랜잭션을 블록 체인을 갖는 다른 노드들로 전송하는 트랜잭션 전송 모듈; 및 블록 체인을 갖는 다른 노드로부터 트랜잭션을 수신하면, 수신된 트랜잭션에 대해 가장 먼저 블록을 생성하면 블록 해쉬값과 논스값을 다른 노드들에게 송신하며, 다른 노드로부터 블록 해쉬값과 논스값을 수신하면 수신된 정보를 이용하여 블록을 생성하여 블록 체인에 연결시키는 블록 체인 실행 모듈; 을 구비하고,

[0010] 상기 블록 체인의 각 블록은 다수의 노드들에 저장된 데이터 파일에 대하여 각각 생성된 것으로서, 각 블록의 블록 해쉬값은 데이터에 대한 정보, 데이터의 소유주 정보와 논스값에 대하여 사전 설정된 해쉬 알고리즘을 적용하여 생성한 해쉬값이며, 상기 트랜잭션의 데이터에 대한 정보는 데이터 명칭을 포함하며, 상기 트랜잭션의 데이터의 소유주 정보는 해당 데이터가 저장된 노드의 IP 주소와 Port 번호를 포함한다.

[0011] 전술한 제1 특징에 따른 파일 관리 및 검색 시스템에 있어서, 상기 트랜잭션 전송 모듈은, 사전에 설정된 폴더내에 데이터가 저장되거나 삭제되면, 데이터 명칭을 설정하고, 상기 설정된 데이터 명칭 및 해당 노드의 IP 주소와 Port 번호, 전자 서명을 포함하는 트랜잭션을 생성하고, 상기 생성된 트랜잭션을 블록 체인을 갖는 다른 노드들로 전송하는 것이 바람직하다.

[0012] 전술한 제1 특징에 따른 파일 관리 및 검색 시스템에 있어서, 상기 트랜잭션 전송 모듈은, 사전에 설정된 폴더내에 데이터가 저장되면, 데이터 생성을 나타내는 제1 플래그를 데이터에 대한 정보에 포함시켜 트랜잭션에 추가시키고, 사전에 설정된 폴더내의 데이터가 삭제되면, 데이터 삭제를 나타내는 제2 플래그를 데이터에 대한 정보에 포함시켜 트랜잭션에 추가시키는 것이 바람직하다.

[0013] 전술한 제1 특징에 따른 파일 관리 및 검색 시스템에 있어서, 블록 체인 실행 모듈은, 다른 노드로부터 트랜잭션을 수신하면, 수신된 트랜잭션의 정보들에 대하여 proof of work를 실행하여 블록 해쉬값과 논스(Nonce)값을 생성하고, 생성된 블록 해쉬값과 논스값을 다른 노드들로 전송하며, 다른 노드로부터 블록 해쉬값과 논스값을 수신하면, 수신된 블록 해쉬값과 논스값을 이용하여 상기 수신된 트랜잭션에 대한 유효성을 검사하여 인증되면 블록을 생성하고, 상기 생성된 블록을 블록 체인에 연결시키는 것이 바람직하다.

[0014] 전술한 제1 특징에 따른 파일 관리 및 검색 시스템에 있어서, 상기 파일 관리 및 검색 시스템은 블록 체인을 이용하여 각 노드에 저장된 데이터와 소유주를 검색하는 파일 검색 모듈을 더 구비하고,

[0015] 상기 파일 검색 모듈은, 검색하고자 하는 데이터 명칭이 입력되면, 블록 체인에 저장된 각 블록의 논스(nonce)값을 판독하고, 상기 판독된 논스값, 상기 입력된 데이터 명칭 및 IP 리스트의 각 노드들에 대한 IP 주소와 Port 번호를 이용하여 해쉬값을 생성하고, 상기 생성된 해쉬값이 상기 논스값이 포함된 블록의 블록 해쉬값과 일치하는지를 판단하고, 상기 생성된 해쉬값과 일치하는 블록 해쉬값을 갖는 블록이 존재하면, 해당 IP 주소와 Port 번호를 갖는 노드가 소유주임을 결정하고, 상기 생성된 해쉬값과 일치하는 블록 해쉬값을 갖는 블록이 존재하지 않으면, 해당 데이터가 없음을 결정하는 것이 바람직하다.

[0016] 전술한 제1 특징에 따른 파일 관리 및 검색 시스템에 있어서, 상기 블록 체인 실행 모듈은, 블록 체인을 갖는 다른 노드들로부터 하나의 트랜잭션을 수신하면, 수신된 하나의 트랜잭션에 대해 하나의 블록을 생성하여 블록 체인에 연결시키는 것이 바람직하다.

[0017] 본 발명의 제2 특징에 따른 블록 체인(Block chain)을 활용하여 다수의 노드들에 저장된 데이터를 관리 및 검색할 수 있도록 하기 위한 각 노드들에서의 파일 관리 및 검색 방법은, (a) 현재 블록의 해쉬(block hash)값, 논스(Nonce) 값, 이전 블록의 해쉬값 및 Time stamp로 이루어진 블록들로 이루어진 블록 체인을 저장하는 단계; (b) 상기 블록 체인을 갖는 모든 노드들에 대한 IP 주소 및 Port 번호를 저장한 IP 리스트를 구비하는 단계; (c) 사전에 설정된 폴더에 데이터가 저장되거나 삭제되면, 상기 데이터에 대한 정보, 소유주에 대한 정보 및 전자 서명을 포함하는 트랜잭션을 생성하고, 상기 생성된 트랜잭션을 블록 체인을 갖는 다른 노드들로 전송하는 단계; 및 (d) 블록 체인을 갖는 다른 노드들로부터 트랜잭션을 수신하면, 수신된 트랜잭션에 대한 블록을 생성하여 블록 체인에 연결시키는 단계; 을 구비하고,

[0018] 상기 블록 체인의 각 블록은 다수의 노드들에 저장된 데이터 파일에 대하여 각각 생성된 것으로서, 각 블록의 블록 해쉬값은 데이터에 대한 정보, 데이터의 소유주 정보와 논스값에 대하여 사전 설정된 해쉬 알고리즘을 적용하여 생성한 해쉬값이며, 상기 트랜잭션의 데이터에 대한 정보는 데이터 명칭을 포함하며, 상기 트랜잭션의

데이터의 소유주 정보는 해당 데이터가 저장된 노드의 IP 주소와 Port 번호를 포함한다.

[0019] 전술한 제2 특징에 따른 파일 관리 및 검색 방법에 있어서, 상기 (c) 단계는, 사전에 설정된 폴더내에 데이터가 저장되거나 삭제되면, 데이터 명칭을 설정하고, 상기 설정된 데이터 명칭 및 해당 노드의 IP 주소와 Port 번호, 전자 서명을 포함하는 트랜잭션을 생성하고, 상기 생성된 트랜잭션을 블록 체인을 갖는 다른 노드들로 전송하는 것이 바람직하다.

[0020] 전술한 제2 특징에 따른 파일 관리 및 검색 방법에 있어서, 상기 데이터 명칭은 아스키 코드(ASCII Code)의 형태로 변환되어 트랜잭션에 포함되고, 상기 IP 주소 및 Port 번호는 십진수의 형태로 변환되어 트랜잭션에 포함되는 것이 바람직하다.

[0021] 전술한 제2 특징에 따른 파일 관리 및 검색 방법에 있어서, 상기 (c) 단계는, 사전에 설정된 폴더내에 데이터가 저장되면, 데이터 생성을 나타내는 제1 플래그를 데이터에 대한 정보에 포함시켜 트랜잭션에 추가시키고, 사전에 설정된 폴더내의 데이터가 삭제되면, 데이터 삭제를 나타내는 제2 플래그를 데이터에 대한 정보에 포함시켜 트랜잭션에 추가시키는 것이 바람직하다.

[0022] 전술한 제2 특징에 따른 파일 관리 및 검색 방법에 있어서, 상기 (d) 단계는, 다른 노드로부터 트랜잭션을 수신하면, 수신된 트랜잭션의 정보들에 대하여 proof of work 를 수행하여 모든 노드들 중 가장 먼저 블록 해쉬값과 논스(Nonce)값을 생성하면, 이를 이용하여 블록을 생성하며, 상기 생성된 블록 해쉬값과 논스값을 다른 노드들로 전송하며, 다른 노드로부터 블록 해쉬값과 논스값을 수신하면, 수신된 블록 해쉬값과 논스값을 이용하여 상기 수신된 트랜잭션에 대한 유효성을 검사하여 인증되면 블록을 생성하고, 상기 생성된 블록을 블록 체인에 연결시키는 것이 바람직하다.

[0023] 전술한 제2 특징에 따른 파일 관리 및 검색 방법에 있어서, 상기 파일 관리 및 검색 방법은 (e) 외부로부터 요청된 데이터 명칭을 검색하여 제공하는 파일 검색 단계를 더 구비하고,

[0024] 상기 (e) 파일 검색 단계는, 검색하고자 하는 데이터 명칭이 입력되면, 블록 체인에 저장된 각 블록의 논스(nonce) 값을 판독하고, 상기 판독된 논스값, 상기 입력된 데이터 명칭 및 IP 리스트의 각 노드들에 대한 IP 주소와 Port 번호를 이용하여 해쉬값을 생성하고, 상기 생성된 해쉬값이 상기 논스값이 포함된 블록의 블록 해쉬값과 일치하는지를 판단하고, 상기 생성된 해쉬값과 일치하는 블록 해쉬값을 갖는 블록이 존재하면, 해당 IP 주소와 Port 번호를 갖는 노드가 소유주임을 결정하고, 상기 생성된 해쉬값과 일치하는 블록 해쉬값을 갖는 블록이 존재하지 않으면, 해당 데이터가 없음을 결정하는 것이 바람직하다.

발명의 효과

[0025] 본 발명에 따른 파일 관리 및 검색 시스템은 블록 체인(Block chain)을 활용하여 데이터 명칭을 검색하고, 해당 데이터에 대한 소유주를 확인할 수 있으며, 소유주의 확인에 의해 소유주의 IP 주소 및 Port 번호를 획득하여 해당 데이터를 요청할 수 있게 된다.

[0026] 또한, 본 발명에 따른 파일 관리 및 검색 시스템은 블록 체인을 활용함으로써, 중앙 관리 서버가 필요치 않은 파일 관리 및 검색 시스템의 플랫폼으로 활용 가능하며, 블록체인 검색을 통한 데이터 위치 파악으로 파일 공유 요청 시스템으로 활용 가능하다.

[0027] 한편, 본 발명에 따른 파일 관리 및 검색 시스템은, 블록 체인의 블록 해쉬값이 데이터의 명칭과 소유주 정보를 포함하도록 함으로써 데이터의 위치를 다른 사용자가 쉽게 확인 가능하다.

[0028] 또한, 본 발명에 따른 파일 관리 및 검색 시스템은, 블록체인 생성과정에서 신뢰성이 보장되므로 보안 측면에서 전자 서명을 통해 인증(Authentication), 부인방지(Non-repudiation)와 블록 hash 생성의 message digest를 통해 무결성(Integrity)을 가진다.

도면의 간단한 설명

[0029] 도 1은 본 발명의 바람직한 실시예에 따른 블록 체인을 기반으로 한 파일 관리 및 검색 시스템을 전체적으로 도시한 구성도이며, 도 2는 본 발명에 따른 파일 관리 및 검색 시스템에 있어서, 각 노드의 구조를 도시한 블록도이다.

도 3은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, 각 노드들에 구비된 블록 체인 및 블록 체인을 구성하는 블록들을 예시적으로 도시한 개념도이다.

도 4는 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, 각 노드들에 의해 수행되는 블록 체인에 데이터 명칭 및 소유주 정보를 포함하는 블록을 연결시키는 과정을 순차적으로 도시한 흐름도이다.

도 5는 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, (a)는 각 노드의 트랜잭션 전송 모듈이 트랜잭션을 생성하는 과정을 설명하기 위하여 도시한 구성도이며, (b) 트랜잭션에 포함되는 데이터 명칭 및 소유주 정보를 예시적으로 출력한 것이다.

도 6은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, (a)는 데이터 또는 파일을 생성한 데이터 소유주인 트랜잭션 송신 노드가 트랜잭션 송신 결과를 출력한 것이며, (b) 트랜잭션 수신 노드의 트랜잭션 수신 결과를 출력한 것이다.

도 7은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, (a)는 전자 서명에 대한 암호화 및 복호화 알고리즘을 설명하기 위하여 도시한 개념도이며, (b)는 전자 서명 및 공개키(public key)를 포함한 트랜잭션을 도시한 것이다.

도 8은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, 트랜잭션 수신 노드가 공개키를 이용하여 복호화한 트랜잭션 유효 검증 결과를 도시한 것이다.

도 9는 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, (a)는 트랜잭션 수신 노드가 proof of work를 통해 블록 해쉬값을 생성하고 전송하는 과정을 도시한 개념도이며, (b)는 상기 블록 해쉬값을 수신한 노드들에 의한 블록 유효 검증 과정을 도시한 것이다.

도 10은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, 각 노드들에 의해 새롭게 생성된 블록이 json 파일 형태로 저장된 것을 도시하는 도면이다.

도 11은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, 각 노드의 파일 검색 모듈의 동작을 도시한 흐름도이며, 도 12는 각 노드들의 파일 검색 모듈을 이용하여 블록체인에 저장된 데이터 명칭 검색 후 소유주를 확인하는 과정을 도시한 개념도이다.

도 13은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, (a)는 블록 체인의 블록 Json 파일을 열어 IP 리스트의 정보와 비교하는 것을 도시하는 도면이고, (b)는 검색한 데이터 명칭을 블록체인 내에서 찾아낸 결과를 도시하는 도면이다.

발명을 실시하기 위한 구체적인 내용

[0030] 본 발명은 데이터 명칭 및 소유주 확인이 가능한 블록체인을 기반으로 한 파일 관리 및 검색 시스템에 관한 것으로서, 생성한 데이터를 저장하고 데이터 명칭을 상기 데이터가 저장된 노드의 IP address와 port number의 정보와 함께 트랜잭션 형태로 다른 노드에 전달하면, 이를 수신한 노드들은 해당 트랜잭션을 기반으로 블록을 생성하여 블록체인에 연결함으로써, 데이터 이름과 해당 데이터를 생성한 소유주 정보를 모든 사람들이 공유하게 된다. 이 정보는 블록체인으로 기록이 되어 있기 때문에 모든 노드가 같은 정보를 공유하여서 악의적 노드가 임의로 내용을 수정하기 위해서는 모든 노드가 가지고 있는 블록체인의 내용을 수정해야 하므로 임의적 수정이 사실상 불가능하기 때문에 블록체인 자체로도 보안성이 뛰어나다.

[0031] 이하, 첨부된 도면을 참조하여 본 발명의 바람직한 실시예에 따른 블록 체인을 기반으로 한 파일 관리 및 검색 시스템의 구조 및 동작에 대하여 구체적으로 설명한다.

[0032] 도 1은 본 발명의 바람직한 실시예에 따른 블록 체인을 기반으로 한 파일 관리 및 검색 시스템을 전체적으로 도시한 구성도이며, 도 2는 본 발명에 따른 파일 관리 및 검색 시스템에 있어서, 각 노드의 구조를 도시한 블록도이다.

[0033] 도 1 및 도 2를 참조하면, 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템(1)은 별도의 중앙 관리 서버없이 블록 체인(Block chain)을 활용하여 다수의 노드들에 저장된 데이터를 관리 및 검색할 수 있도록 하는 시스템으로서, 각 노드들(2,3,4, ..., n)은, 블록 체인(10), 데이터 저장 모듈(12), IP 리스트(11), 트랜잭션 전송 모듈(13), 블록 체인 실행 모듈(14) 및 파일 검색 모듈(15)을 구비한다.

[0034] 상기 블록 체인(10)은 다수 개의 블록들이 사슬 형태로 연결되어 이루어진 것으로서, 상기 블록들은 Json 파일 형태로 이루어지며, 현재 블록의 해쉬(Hash) 값, 논스(Nonce) 값, 이전 블록의 해쉬값, 타임 스탬프(Time Stamp)를 포함한다. 상기 해쉬값과 논스값은 데이터 명칭과 소유주의 IP 주소 및 Port 번호에 대하여 사전 설정

된 해쉬 함수를 이용하여 구한 값들이다.

[0035] 도 3은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, 각 노드들에 구비된 블록 체인 및 블록 체인을 구성하는 블록들을 예시적으로 도시한 개념도이다. 도 3을 참조하면, 블록 체인의 블록들은 자신의 블록 해쉬값 뿐만 아니라 바로 이전의 블록에 대한 블록 해쉬값도 함께 포함하고 있으므로, 블록 체인의 각 블록들은 사슬처럼 서로 연결되어 이어져 나가게 된다.

[0036] 상기 IP 리스트(11)는 상기 블록 체인을 갖는 모든 노드들에 대한 IP 주소 및 Port 번호를 저장한 것으로서, 모든 노드들이 구비한다.

[0037] 상기 데이터 저장 모듈(12)은 데이터 또는 파일들이 생성되어 사전 설정된 폴더에 저장되는 메모리 영역으로서, 각 노드들은 데이터 저장 영역에 다른 노드들과 공유하고자 하는 데이터 또는 파일을 저장하기 위한 폴더를 사전 지정해 둔다. 본 발명에 따른 시스템에 의하여 모든 노드들이 상기 데이터 저장 모듈에 저장된 데이터들을 공유할 수 있는 파일 공유 플랫폼을 제공할 수 있게 된다.

[0038] 상기 트랜잭션 전송 모듈(13)은 상기 데이터 저장 모듈(12)의 사전 설정된 폴더에 데이터 또는 파일이 저장되거나 삭제되면, 데이터 명칭을 설정하고, 상기 데이터 명칭을 포함하는 상기 데이터 또는 파일에 대한 정보, 소유주 정보 및 전자 서명을 포함하는 트랜잭션을 생성하고, 상기 생성된 트랜잭션을 블록 체인을 갖는 다른 노드들로 전송한다. 본 명세서에서는, 설명의 편의상, 트랜잭션을 전송한 노드를 "트랜잭션 송신 노드"라고 표시한다.

[0039] 상기 데이터 명칭은 아스키 코드(ASCII Code)의 형태로 변환되어 트랜잭션에 포함되고, 상기 IP 주소 및 Port 번호는 십진수의 형태로 변환되어 트랜잭션에 포함되는 것이 바람직하다.

[0040] 상기 트랜잭션 전송 모듈은, 사전에 설정된 폴더내에 데이터 또는 파일이 저장되면, 데이터 또는 파일의 생성을 나타내는 제1 플래그를 데이터에 대한 정보의 일부로서 트랜잭션에 추가시키고, 사전에 설정된 폴더내의 데이터 또는 파일이 삭제되면, 데이터 삭제를 나타내는 제2 플래그를 데이터에 대한 정보의 일부로서 트랜잭션에 추가시키는 것이 바람직하다.

[0041] 상기 블록 체인 실행 모듈(14)은 블록 체인을 갖는 다른 노드들로부터 트랜잭션을 수신하면, 수신된 트랜잭션에 대해 블록을 생성하여 블록 체인에 연결시켜 블록 체인을 확장시키게 된다. 본 명세서에서는, 설명의 편의상, 트랜잭션 송신 노드로부터 트랜잭션을 수신한 노드를 "트랜잭션 수신 노드"라고 표시한다. 이하, 상기 블록 체인 실행 모듈(14)의 동작을 보다 구체적으로 설명한다.

[0042] 상기 블록 체인 실행 모듈은 다른 노드로부터 트랜잭션을 수신하면, 자신이 블록을 생성하기 위하여 필요한 블록 해쉬값을 만들기 위하여 proof-of-work 를 실행한다. 상기 proof of work 는 사전 설정된 해쉬 함수를 사용하여 랜덤한 논스값을 상기 수신된 트랜잭션과 연산하여 정해진 '0'의 개수를 충족시키는 16진수의 블록 해쉬값을 만드는 작업이다. 본 발명에 따른 시스템에서는 SHA 256 해쉬 함수를 사용하며, 그 외의 다른 해쉬 함수도 사용가능하다. 이와 같이 proof of work를 하는 이유는, 블록 체인에 참여한 노드들 중 어떠한 노드가 블록을 생성할지 모르게 만들기 위한 것으로서, 악의적 노드가 현재 블록을 생성할 노드를 판단하지 못하게 하여 악의적 노드의 공격(attack)을 방지하게 된다. 예컨대, 비트 코인의 블록 체인의 높이는 417453개이고, 이 블록 해쉬값의 '0'의 개수는 17개인데, 비트 코인은 '0'의 개수를 늘리면서 블록 생성 난이도를 조절하는데, 현재 가장 높은 CPU 또는 GPU 성능을 갖는 시스템을 기준으로 하여 10분에 1개의 블록을 생성하는 '0'의 개수를 취하는 식으로 블록 생성 난이도를 설정하고 있다.

[0043] 상기 블록 체인 실행 모듈은, 상기 트랜잭션 수신 노드들 중 proof-of-work 를 가장 먼저 성공하면, 블록 해쉬값과 랜덤한 논스(Nonce) 값을 찾아내고, 이를 이용하여 블록을 생성하며, 블록 생성 사실과 상기 찾아낸 블록 해쉬값과 랜덤한 논스값을 전체 노드들에게 전송한다. 본 명세서에서는, 설명의 편의상, 트랜잭션 수신 노드들 중 proof-of-work를 가장 먼저 성공한 노드를 "proof of work 성공 노드"라고 표시한다.

[0044] 상기 블록 체인 실행 모듈은, 상기 proof of work 성공 노드로부터 블록 해쉬값과 논스값을 수신하면, 유효 검정 알고리즘을 이용하여 트랜잭션과 상기 수신한 블록 해쉬값과 논스값에 대하여 유효성을 판단한 후, 유효성 검증이 완료되면 상기 수신한 블록 해쉬값과 논스값을 이용하여 추가의 블록을 생성하고 상기 블록 체인에 상기 추가의 블록을 연결시킨다.

[0045] 상기 파일 검색 모듈(15)은 블록 체인(10)과 IP 리스트(11)를 이용하여 각 노드에 저장된 데이터와 소유주를 검색할 수 있도록 한다.

[0046] 상기 파일 검색 모듈은, 검색하고자 하는 데이터 명칭이 입력되면, 블록 체인에 저장된 각 블록의 논스(nonce)

값을 판독하고, 상기 판독된 논스값, 상기 입력된 데이터 명칭 및 IP 리스트의 각 노드들에 대한 IP 주소와 Port 번호를 이용하여 블록 해쉬값을 생성하고, 상기 생성된 블록 해쉬값이 상기 논스값이 포함된 블록의 블록 해쉬값과 일치하는지를 판단한다. 만약, 상기 생성된 블록 해쉬와 일치하는 블록 해쉬값을 갖는 블록이 존재하면, 해당 IP 주소와 Port 번호를 갖는 노드가 소유주임을 결정하고, 해당 노드로 해당 데이터를 요청한다. 만약, 상기 생성된 블록 해쉬값과 일치하는 블록 해쉬값을 갖는 블록이 존재하지 않으면, 해당 데이터는 없다고 결정하게 된다.

[0047] 한편, 본 발명에 따른 시스템에 있어서, 상기 블록 체인 실행 모듈은, 블록 체인을 갖는 다른 노드들로부터 하나의 트랜잭션을 수신하면, 수신된 하나의 트랜잭션에 대해 하나의 블록을 생성하여 블록 체인에 연결시키는 것을 특징으로 한다.

[0048] 이하, 전술한 구성을 갖는 블록 체인을 기반으로 한 파일 관리 및 검색 시스템에 있어서, 각 노드에서의 파일 관리 및 검색 방법을 보다 구체적으로 설명한다.

[0049] 도 4는 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, 각 노드들에 의해 수행되는 블록 체인에 데이터 명칭 및 소유주 정보를 포함하는 블록을 연결시키는 과정을 순차적으로 도시한 흐름도이다. 도 4를 참조하여, 각 노드들에 의해 수행되는 블록 체인에 데이터 명칭 및 소유주 정보를 포함하는 블록을 연결시키는 과정을 구체적으로 설명한다.

[0050] 각 노드들은 블록 체인 및 IP 리스트를 구비하며, 각 노드들은 데이터 저장 영역에 다른 노드들과 공유하고자 하는 데이터 또는 파일을 저장하기 위한 폴더를 사전 지정해 둔다.

[0051] 먼저, 각 노드들은, 트랜잭션 전송 모듈에 의해, 상기 사전 설정된 폴더에 데이터 또는 파일을 생성하여 저장하거나 삭제하면, 상기 데이터 또는 파일에 대한 명칭을 설정하고, 상기 데이터 명칭, 상기 데이터에 대한 소유주인 데이터가 저장된 노드의 IP 주소와 Port 번호, 전자 서명을 포함하는 트랜잭션을 생성하고, 생성된 트랜잭션을 다른 노드들에게 전체적으로 전송한다(단계 400). 이때, 상기 트랜잭션은 데이터의 생성을 제1 플래그 및 데이터의 삭제를 나타내는 제2 플래그 중 하나를 더 구비하여, 향후 검색시에 해당 데이터가 생성 또는 삭제되었음을 판단할 수 있도록 하는 것이 바람직하다.

[0052] 한편, 다른 노드들은 트랜잭션 송신 노드로부터 트랜잭션을 수신하게 된다(단계 410).

[0053] 상기 트랜잭션 수신 노드들은, 블록 체인 실행 모듈을 통해, 상기 수신된 트랜잭션에 대하여 자신이 블록을 생성하기 위하여 필요한 블록 해쉬값을 만들기 위하여 proof-of-work 를 실행한다(단계 420).

[0054] 상기 트랜잭션 수신 노드들 중 proof-of-work 를 가장 먼저 성공한 노드는, 블록 체인 실행 모듈을 통해, 블록 해쉬값과 랜덤한 논스(Nonce) 값을 찾아내고, 이를 이용하여 블록을 생성하며, 블록 생성 사실과 상기 찾아낸 블록 해쉬값과 랜덤한 논스값을 전체 노드들에게 전송한다(단계 430).

[0055] 상기 proof of work 성공 노드로부터 블록 해쉬값과 논스값을 수신한 다른 노드들은 유효 검정 알고리즘을 이용하여 트랜잭션과 상기 수신한 블록 해쉬값과 논스값에 대하여 유효성을 판단한 후(단계 440), 유효성 검증이 완료되면 상기 수신한 블록 해쉬값과 논스값을 이용하여 추가의 블록을 생성하고 상기 블록 체인에 상기 추가의 블록을 연결시킨다(단계 450).

[0056] 도 5는 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, (a)는 각 노드의 트랜잭션 전송 모듈이 트랜잭션을 생성하는 과정을 설명하기 위하여 도시한 구성도이며, (b) 트랜잭션에 포함되는 데이터 명칭 및 소유주 정보를 예시적으로 출력한 것이다.

[0057] 도 5를 참조하면, 소유주는 사전 지정된 폴더내에 데이터 또는 파일을 생성하고, 데이터 명칭을 a1.txt 라 명명한다. 블록 체인에 처음 참여하는 노드는 블록 체인 실행 프로그램과 같은 디렉토리에 임의의 폴더를 생성하여 해당 폴더에 블록 체인의 블록들을 저장하게 된다. 한편, 각 노드는 Python의 Watchdog API를 사용하여 사용자 지정 폴더의 모든 파일의 생성, 삭제 및 수정을 감시하여 파일이 생성되었을 때만 Python 프로그램을 진행하는 것이 바람직하다. 이때, 트랜잭션에 사전 설정된 플래그를 포함시킴으로써, 해당 명칭의 데이터 또는 파일의 현재 상태를 알려주는 것이 바람직하다. 예컨대, 데이터 또는 파일의 생성시에는 제1 플래그를 트랜잭션에 포함시키고 데이터 또는 파일의 삭제시에는 제2 플래그를 트랜잭션에 포함시킴으로써, 해당 명칭의 데이터를 필요로 하는 노드가 검색할 때에 데이터의 존재 여부를 확인할 수 있도록 하게 된다. 한편, 생성 및 삭제된 데이터 명칭은 문자를 숫자로 표현가능한 아스키(ASCII) 코드를 이용하여 단순 나열을 통한 데이터 명칭을 표현하고 IP 주소는 이진수로 변환시킨 후 십진수로 변환시키는 것이 바람직하다. 이러한 과정을 통해 추출된 데이터 명칭,

소유주의 IP 주소와 Port 번호와 전자 서명(Signature)과 함께 트랜잭션을 생성하게 된다.

[0058] 도 6은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, (a)는 데이터 또는 파일을 생성한 데이터 소유주인 트랜잭션 송신 노드가 트랜잭션 송신 결과를 출력한 것이며, (b) 트랜잭션 수신 노드의 트랜잭션 수신 결과를 출력한 것이다.

[0059] 도 6의 (a)를 참조하면, 블록 체인에 참여하는 노드들은 모든 노드들에 대한 IP 리스트를 구비하고 있으며, IP 리스트에는 블록 체인 참여 노드들의 IP 주소, Port 번호가 저장되어 있다. 데이터 생성후 트랜잭션을 생성한 노드는 IP 리스트에 저장된 IP 주소와 Port 번호를 이용하여 모든 노드들에게 트랜잭션을 전송하며, 이때 전송 방식은 TCP Socket 통신을 사용한다. 도 6의 (b)를 참조하면, 트랜잭션을 생성한 노드가 아닌 나머지 노드들이 트랜잭션을 수신한 결과로서, 트랜잭션을 성공적으로 수신한 것을 확인할 수 있다. 종래의 비트코인에서 사용되는 블록 체인은 트랜잭션 내부에 송신 주소, 수신 주소, 거래량 등의 많은 정보가 필요하지만, 본 발명에 따른 블록 체인은 모든 노드들이 공유한 데이터의 위치를 확인할 수 있도록 하기 위하여 트랜잭션에 데이터 명칭과 소유주 정보만을 전송함으로써, 송수신되는 패킷의 크기를 줄여 링크의 bandwidth를 최소화시킬 수 있게 된다.

[0060] 도 7은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, (a)는 전자 서명에 대한 암호화 및 복호화 알고리즘을 설명하기 위하여 도시한 개념도이며, (b)는 전자 서명 및 공개키(public key)를 포함한 트랜잭션을 도시한 것이다.

[0061] 도 7의 (a)를 참조하면, 본 발명에 따른 시스템에 있어서, 데이터 소유주인 트랜잭션 송신 노드는 트랜잭션을 송신할 때 private key를 이용하여 전자서명을 생성하고 전자서명을 트랜잭션에 포함시켜 송신하며, 트랜잭션 수신 노드는 public key를 이용하여 수신된 트랜잭션의 전자서명을 복호화하여 트랜잭션을 송신한 소유주 IP 및 port와 트랜잭션에 포함된 IP address 및 port number가 일치하는지 확인하게 된다.

[0062] 본 발명에 따른 시스템은, private key와 public key를 이용하여 전자 서명(signature)를 생성하게 되는데, 전자서명 시 사용한 암호 이론은 ECDSA이며, 트랜잭션 송신 노드의 IP address와 TCP socket 통신에 사용할 소유주의 port number를 나열하여 자신의 private key와 암호화함으로써, 전자서명을 생성하게 된다. 해당 전자서명은 트랜잭션에 첨부되고 ECDSA로 생성한 public key를 동봉하여 다른 노드에 전송한다. 트랜잭션 수신 노드는 동봉된 public key를 이용해 복호화시켜 출력된 결과가 True 혹은 False인지를 확인하여 해당 트랜잭션이 유효함을 검증하게 된다. 전자서명의 확인으로 spoofing을 방지하여 임의의 악의적 노드가 잘못된 트랜잭션을 전송하여 블록에 포함시키는 것을 방지한다.

[0063] 도 7의 (b)는 데이터 생성 소유주, 트랜잭션 송신 노드가 전송한 트랜잭션에 포함된 5가지 정보인 데이터 명칭, IP address, port number, public key, 전자서명을 도시한다.

[0064] 도 8은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, 트랜잭션 수신 노드가 공개키를 이용하여 복호화한 트랜잭션 유효 검증 결과를 도시한 것이다. 도 8을 참조하면, 트랜잭션 수신 노드는 트랜잭션 송신 노드의 public key를 이용해, 전자서명을 복호화하고, 만약 트랜잭션 유효 검증을 위한 복호화의 결과가 true임을 확인하면 유효성 검증 이후 블록생성단계를 진행하고, 만약 복호화의 결과가 false일 경우 해당 트랜잭션을 버리게 된다.

[0065] 도 9는 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, (a)는 트랜잭션 수신 노드가 proof of work를 통해 블록 해쉬값을 생성하고 전송하는 과정을 도시한 개념도이며, (b)는 상기 블록 해쉬값을 수신한 노드들에 의한 블록 유효 검증 과정을 도시한 것이다.

[0066] 도 9의 (a)를 참조하면, 실제 비트코인 블록체인 블록 hash의 0의 개수는 17개가 필요한 난이도로 설정되어 있으며 그에 맞게 난이도를 설정한다. 기존의 블록체인과 달리 여러 트랜잭션을 포함한 블록이 아닌 하나의 트랜잭션만 포함하는데, 즉 하나의 블록에 하나의 데이터 명칭, 소유주 정보가 입력된다. 이 정보만 이용하여 블록 hash로 만들어 추후에 데이터 검색 시 블록 hash 분석을 통해 데이터 명칭과 소유주 정보를 확인 가능하게 한다. Hash 알고리즘을 적용하기 위해 트랜잭션으로 수신한 IP, port, 데이터 명칭을 나열하고, 랜덤하게 nonce 값을 생성해서 SHA256 hash 알고리즘을 사용해 블록 hash를 생성한다(1). 설정한 기준 값인 0이 17개보다 많은 0을 가진 hash값을 찾기까지 nonce를 계속 변화시켜가고, 조건에 타당한 hash를 찾을 시 블록 hash로 받아들인다. 임의의 노드가 생성한 블록 hash값과 nonce값을 블록체인에 참여한 모든 노드들에게 전송하며(2) 수신한 노드는 자신이 받은 블록 hash값과 nonce값을 분석하여 유효성 검증을 진행한다(3). 도 9의 (b)는 블록 유효 검증을 실시하여 hash 알고리즘 결과와 수신한 블록 hash가 일치하는 것을 확인 가능한 도면이다. 각 노드들은 블록

생성 위한 트랜잭션 정보를 모두 가지고 있으므로, 트랜잭션 정보인 IP 주소, port 번호, 데이터 명칭을 나열하고 수신한 nonce값을 이용해 SHA256 해쉬 알고리즘을 진행하여 출력된 hash 값이 수신한 블록 hash값과 일치했을 때 트랜잭션의 내용이 변하지 않았다는 사실을 노드가 인지하여, 악의적 노드의 블록 생성을 차단할 수 있다. 블록 유효성 검증이 완료된 블록은 해당 내용들이 Json 파일 형태로 저장된 후(4), 블록 체인에 연결된다(5).

[0067] 도 10은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, 각 노드들에 의해 새롭게 생성된 블록이 Json 파일 형태로 저장된 것을 도시하는 도면이다.

[0068] 도 10을 참조하면, 블록은 Json 파일 형태로 저장되며, 타임 스탬프('time', 현재 블록의 해쉬값('hash'), 논스값(nonce'), 이전 블록의 해쉬값('prev hash')을 포함하게 된다.

[0069] 도 11은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, 각 노드의 파일 검색 모듈의 동작을 도시한 흐름도이며, 도 12는 각 노드들의 파일 검색 모듈을 이용하여 블록체인에 저장된 데이터 명칭 검색 후 소유주를 확인하는 과정을 도시한 개념도이다.

[0070] 도 12는 블록체인에 참여한 사용자가 필요한 데이터를 요청받기 위해 데이터 명칭을 검색하고 해당 데이터를 가진 소유주를 찾는 과정이다. 도 11 및 도 12를 참조하면, 먼저 사용자가 데이터 명칭(a)을 텍스트로 입력하면(단계 900), ASCII code로 변환 후 IP 리스트(11)에 있는 각 노드들의 IP 주소와 port 번호의 정보와 함께 나열한다(단계 910). Json 파일 형태로 저장된 블록체인(10)을 읽어 해당 블록의 nonce 값을 판독하고(단계 920), 상기 판독된 nonce 값과 앞서 나열한 정보들과 함께 SHA256 hash 알고리즘을 적용시켜 해쉬값을 구하며(단계 930), 이렇게 구한 해쉬값이 Json 파일의 블록 hash값과 일치하는지를 확인한다(단계 940). Json 블록 하나를 열어 IP 리스트의 모든 노드들의 IP 주소와 port 번호를 이용해 nonce값과 함께 hash값을 찾아내고 일치하는 hash값이 존재하지 않는다면 다음 블록을 열어 같은 방식으로 반복하여 진행한다. 블록의 hash값과 일치하는 hash 값을 찾아내면, 해당 IP 주소와 port 번호를 가진 노드가 검색한 데이터의 소유주임을 확인 가능하다(단계 950). 만약 검색한 데이터 명칭이 블록체인에 존재하지 않을 경우 "not found"를 return한다(단계 960).

[0071] 도 13은 본 발명의 바람직한 실시예에 따른 파일 관리 및 검색 시스템에 있어서, (a)는 블록 체인의 블록 Json 파일을 열어 IP 리스트의 정보와 비교하는 것을 도시하는 도면이고, (b)는 검색한 데이터 명칭을 블록체인 내에서 찾아낸 결과를 도시하는 도면이다.

[0072] 도 13의 (a)를 참조하면, 블록 체인을 구성하는 각 블록에 대한 Jason 파일을 열어 논스(nonce) 값을 판독하고, IP 리스트를 구성하는 각 노드들에 대한 IP 주소와 Port 번호를 판독하고, 사용자로부터 입력된 데이터 명칭과 상기 판독된 각 노드에 대한 IP 주소와 Port 번호, 그리고 각 블록에 대한 논스 값을 이용하여 해쉬값을 구한다. 이렇게 구한 해쉬값이 상기 논스 값이 포함된 블록의 해쉬값과 일치하는 경우, 해당 노드가 데이터의 소유주로 확인된다.

[0073] 이와 같이, 본 발명에 따른 파일 관리 및 검색 시스템은, 데이터 명칭을 블록체인의 hash를 검색하여 소유주(File owner)의 IP address 및 port 정보를 찾아냄으로서 공유 가능한 파일의 리스트를 확인 가능한 플랫폼을 구현할 수 있고, 블록체인 상에서 데이터의 소유주 정보를 확인했을 시, 검색한 노드는 소유주 노드에 해당 데이터를 요청하게 되고, 소유주 노드의 confirm 후 파일을 TCP socket 통신을 통해 전송함으로서 파일 공유 시스템 또한 구현 가능하다.

[0074] 한편, 본 발명에 따른 파일 관리 및 검색 시스템은 블록체인을 통한 데이터 명칭과 소유주의 정보만 저장하여 블록체인 검색을 통해 필요한 데이터를 공유 요청하는 형식이기 때문에 기존의 클라우드 방식에서 공유하기 위한 파일 전체를 저장하여 많은 저장 용량 소모의 문제를 해결할 수 있으며, 블록체인에 속한 블록의 정보를 블록의 높이, 블록의 개수, 트랜잭션 정보 등을 제외시키고 데이터 검색에 필요한 블록 hash 및 nonce 등의 정보만 저장하므로 블록체인 자체가 가지는 저장 용량 과다 소모 문제 또한 해결 가능하다.

[0075] 이상에서 본 발명에 대하여 그 바람직한 실시예를 중심으로 설명하였으나, 이는 단지 예시일 뿐 본 발명을 한정하는 것이 아니며, 본 발명이 속하는 분야의 통상의 지식을 가진 자라면 본 발명의 본질적인 특성을 벗어나지 않는 범위에서 이상에 예시되지 않은 여러 가지의 변형과 응용이 가능함을 알 수 있을 것이다. 그리고, 이러한 변형과 응용에 관계된 차이점들은 첨부된 청구 범위에서 규정하는 본 발명의 범위에 포함되는 것으로 해석되어야 할 것이다.

산업상 이용가능성

[0076] 본 발명에 따른 시스템 및 방법은 파일 공유 시스템에 널리 사용될 수 있다.

부호의 설명

[0077] 1 : 파일 관리 및 검색 시스템

2,3,4 : 노드

10 : 블록 체인

12 : 데이터 저장 모듈

11 : IP 리스트

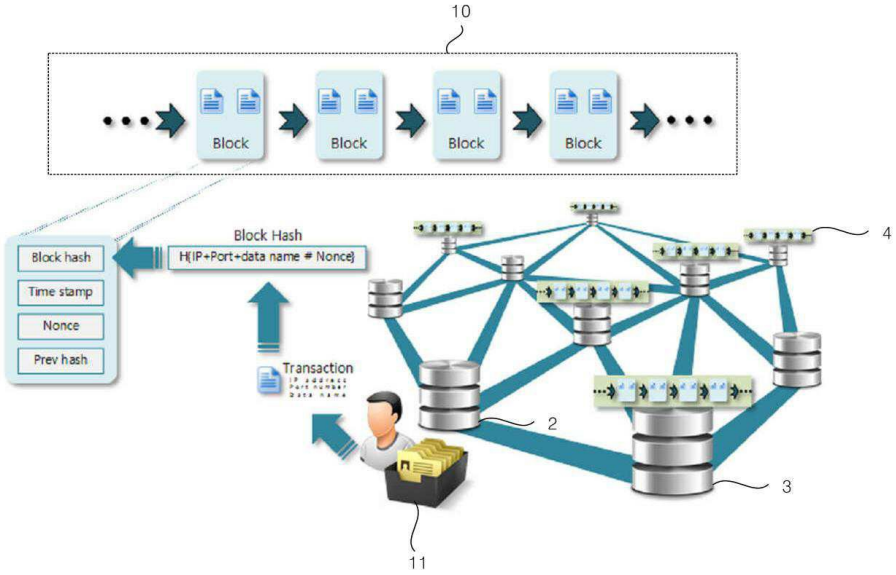
13 : 트랜잭션 전송 모듈

14 : 블록 체인 실행 모듈

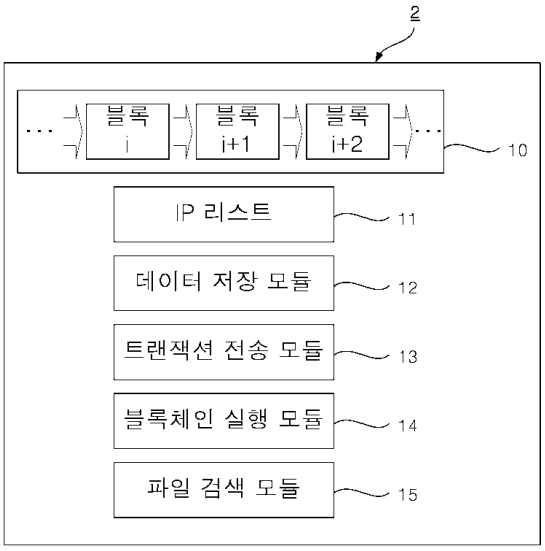
15 : 파일 검색 모듈

도면

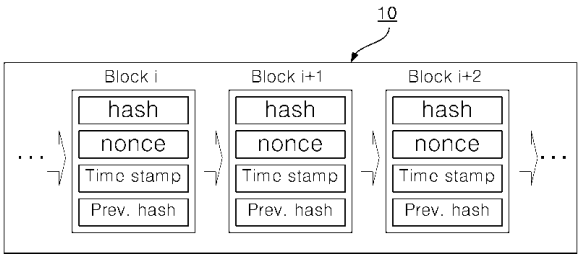
도면1



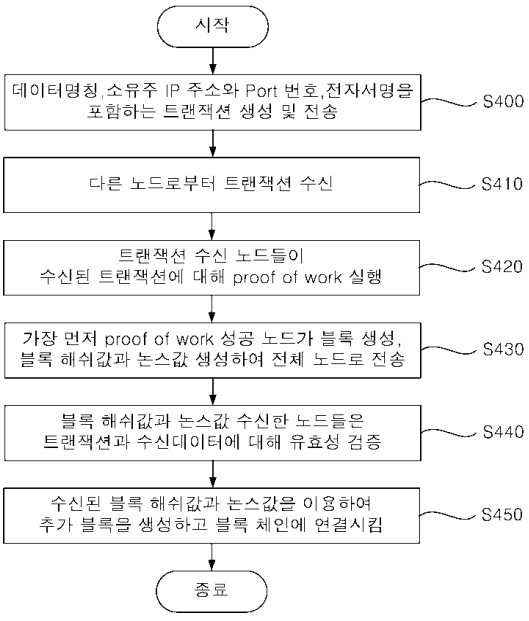
도면2


2
블록 i
블록 i+1
블록 i+2
10
IP 리스트
11
데이터 저장 모듈
12
트랜잭션 전송 모듈
13
블록체인 실행 모듈
14
파일 검색 모듈
15

도면3

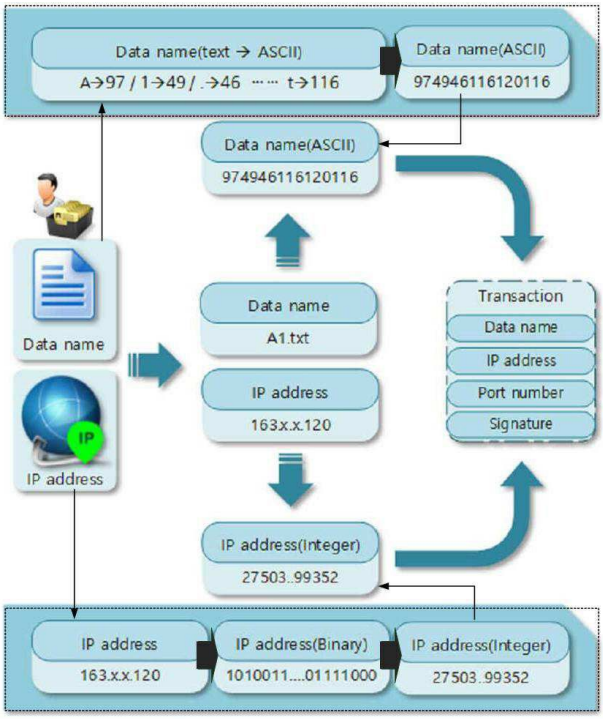

10
Block i
Block i+1
Block i+2
hash
nonce
Time stamp
Prev. hash

도면4

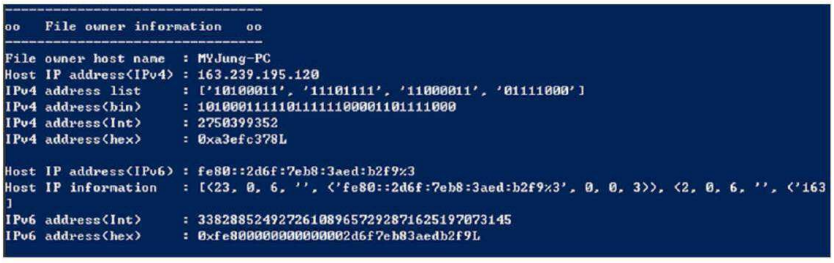

시작
데이터명칭,소유주 IP 주소와 Port 번호,전자서명을 포함하는 트랜잭션 생성 및 전송
S400
다른 노드로부터 트랜잭션 수신
S410
트랜잭션 수신 노드들이 수신된 트랜잭션에 대해 proof of work 실행
S420
가장 먼저 proof of work 성공 노드가 블록 생성, 블록 해쉬값과 논스값 생성하여 전체 노드로 전송
S430
블록 해쉬값과 논스값 수신한 노드들은 트랜잭션과 수신데이터에 대해 유효성 검증
S440
수신된 블록 해쉬값과 논스값을 이용하여 추가 블록을 생성하고 블록 체인에 연결시킴
S450
종료

도면5

(a)

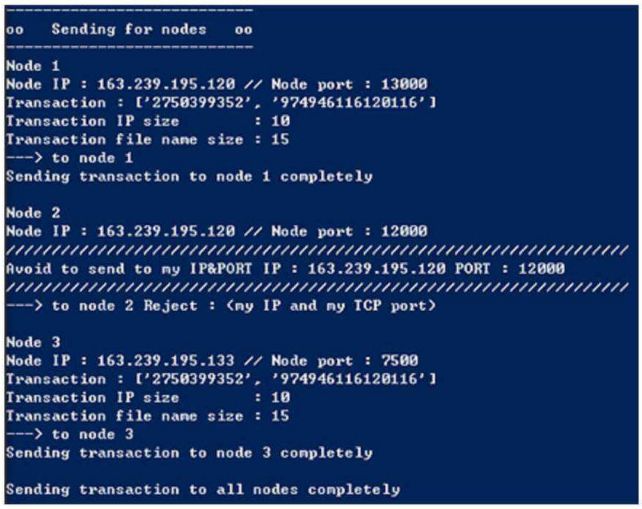

Data name(text → ASCII)
A→97 / 1→49 / .→46 ····· t→116
Data name(ASCII)
974946116120116
Data name(ASCII)
974946116120116
Data name
Data name
A1.txt
IP address
IP address
163.x.x.120
Transaction
Data name
IP address
Port number
Signature
IP address(Integer)
27503..99352
IP address
163.x.x.120
IP address(Binary)
1010011....01111000
IP address(Integer)
27503..99352

(b)

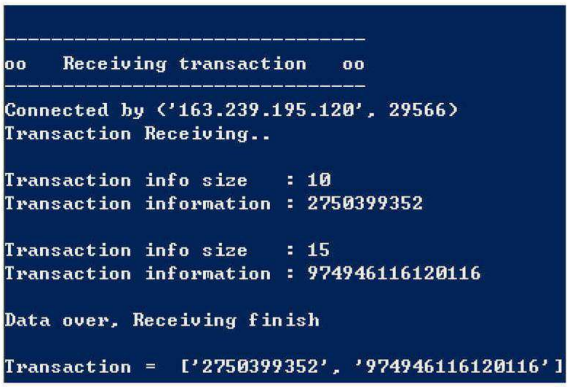

oo File owner information oo
File owner host name : MYJung-PC
Host IP address(IPv4) : 163.239.195.120
IPv4 address list : ['10100011', '11101111', '11000011', '01111000']
IPv4 address(bin) : 10100011111011111100001101111000
IPv4 address(Int) : 2750399352
IPv4 address(hex) : 0xa3efc378L
Host IP address(IPv6) : fe80::2d6f:7eb8:3aed:b2f9%3
Host IP information : [(23, 0, 6, '', ('fe80::2d6f:7eb8:3aed:b2f9%3', 0, 0, 3)), (2, 0, 6, '', ('163
]
IPv6 address(Int) : 338288524927261089657292871625197073145
IPv6 address(hex) : 0xfe800000000000002d6f7eb83aedb2f9L

도면6

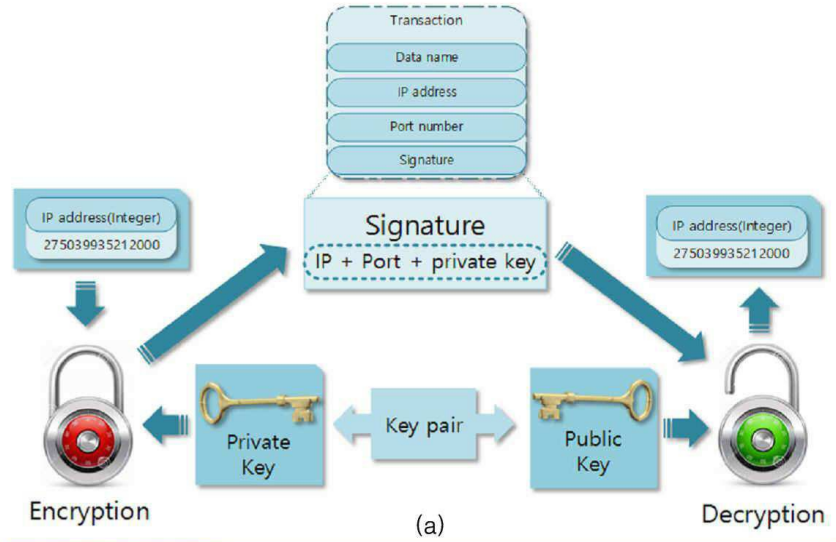
(a)


oo Sending for nodes oo
Node 1
Node IP : 163.239.195.120 // Node port : 13000
Transaction : ['2750399352', '974946116120116']
Transaction IP size : 10
Transaction file name size : 15
---> to node 1
Sending transaction to node 1 completely
Node 2
Node IP : 163.239.195.120 // Node port : 12000
Avoid to send to my IP&PORT IP : 163.239.195.120 PORT : 12000
---> to node 2 Reject : (my IP and my TCP port)
Node 3
Node IP : 163.239.195.133 // Node port : 7500
Transaction : ['2750399352', '974946116120116']
Transaction IP size : 10
Transaction file name size : 15
---> to node 3
Sending transaction to node 3 completely
Sending transaction to all nodes completely

(b)


oo Receiving transaction oo
Connected by ('163.239.195.120', 29566)
Transaction Receiving..
Transaction info size : 10
Transaction information : 2750399352
Transaction info size : 15
Transaction information : 974946116120116
Data over, Receiving finish
Transaction = ['2750399352', '974946116120116']

도면7

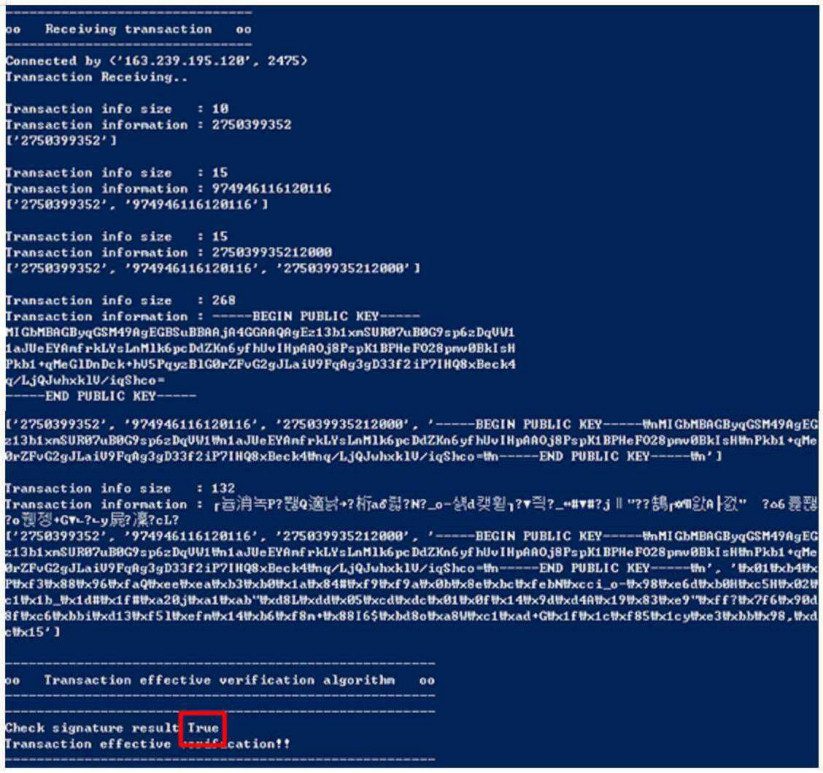


(a)

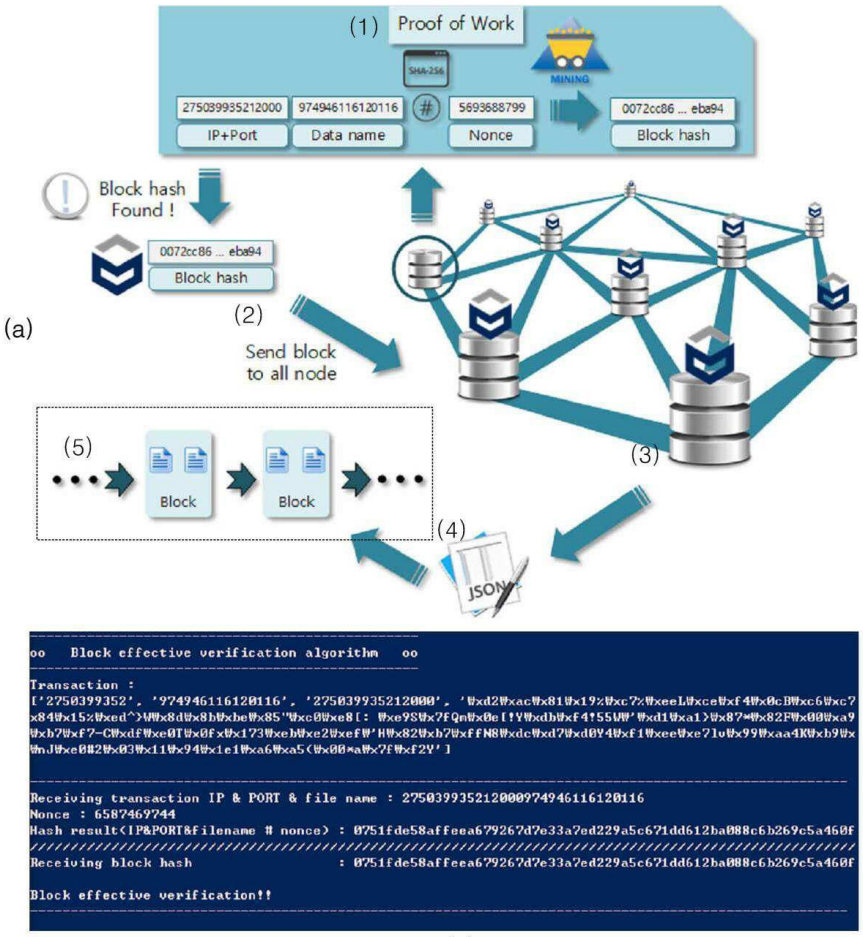
```
oo   Sending for nodes   oo
Node 1
Node IP : 163.239.195.120 // Node port : 13000
SocketList : [<socket._socketobject object at 0x0000000002FE1EE8>]
Transaction : ['2750399352', '9749461161201l6', '275039935212000', u'-----BEGIN PUBLIC KEY-----\nMIGbM
BBQAjA4GGAAQAgEz13b1xmSUR07uB0G9sp6zDqVV1\n1aJUeEVAmfrkLVsLnM1k6pcDdZKn6yFhUvIHpAAOj8PspK1BPHeFO28pmv0
nDck+hV5PqyzB1G0rZPoG2gJLaiU9FqAg3gD33f2iP7IHQ8xBeck4Vnq/LjQJvhxklU/iqShco=\n-----END PUBLIC KEY-----\
xe1\xbc\xb3\xecP\xf3\x88\x96\xfaQ\xee\xea\xb3\xb0\x1a\x84\x9f\xf9a\x0b\x8e\xbc\xfebN\xcci_o-\x98\xe6d
\x1f\x1f\xa3q\xci\x1b_\x1d#\x1f##\xa20j\xaI\xab'\xd8L\xdd#\x05\xcd\xdc\x01\x0f\x14\x9d\xd4A\x19\x83\xe9"
6\x91\xf0\xb0\x0f\xc6\xbhi\xd13\xf51\xefm\x14\xb6\xf8n+\x88I6$\xbd8o\xa8U\xci\xad+G\x1f\x1c\xf85\xicy#
f2\x88\x00cL\x8c\x15']
Transaction IP size         : 10
Transaction file name size : 15
Transaction PORT size       : 15
Transaction public key size       : 268
Transaction signature size       : 132
---> to node 1
Sending transaction to node 1 completely
```

(b)

도면8


oo Receiving transaction oo
Connected by ('163.239.195.120', 2475)
Transaction Receiving..
Transaction info size : 10
Transaction information : 2750399352
['2750399352']
Transaction info size : 15
Transaction information : 974946116120116
['2750399352', '974946116120116']
Transaction info size : 15
Transaction information : 275039935212000
['2750399352', '974946116120116', '275039935212000']
Transaction info size : 268
Transaction information : -----BEGIN PUBLIC KEY-----
MIGbMBAGByqGSM49AgEGBSuBBAAjA4GGAAQAgEz13b1xnSUR07uB0G9sp6zDqVU1
iaJUeEYAnfrkLYsLnMlk6pcDdZKn6yfhUvIHpAAOj8PspK1BPHeFO28pnv0BkIsH
Pkb1+qMeGlDnDck+hV5PqyzB1G0rZFvG2gJLaiV9FqAg3gD33f2iP7IHQ8xBeck4
q/LjQJwhxklV/iqShco=
-----END PUBLIC KEY-----
Transaction info size : 132
oo Transaction effective verification algorithm oo
Check signature result True
Transaction effective verification!!

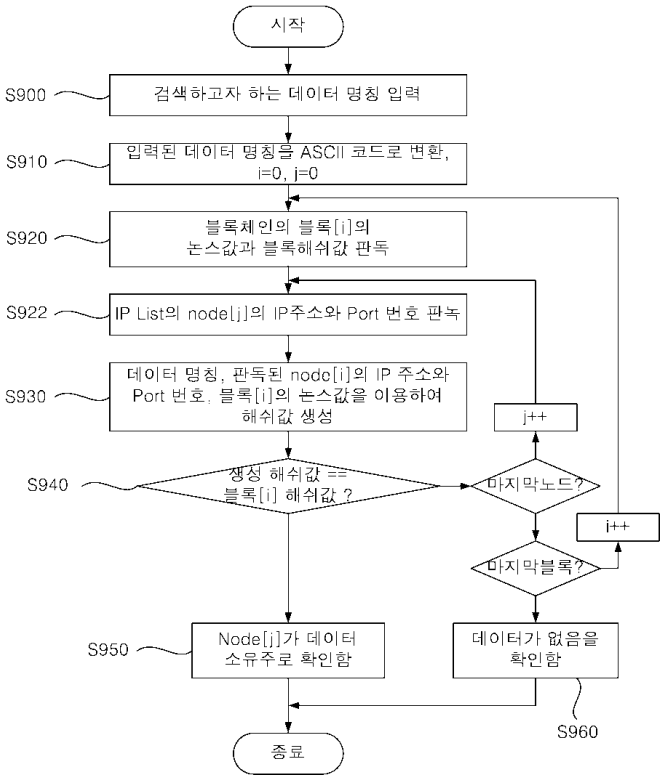
도면9



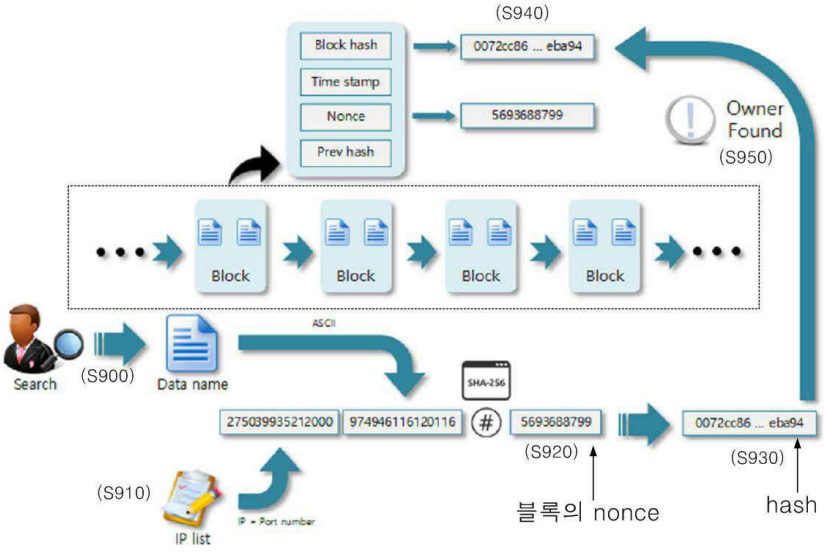
도면10


```
{
    "time      ": "1463486342.25",
    "hash      ": "0d8dc97b8d8891f0920558a6c765fef4f0b156ac1d8044494b8f7f986fdeb3a7",
    "nonce     ": "1536241736",
    "prev hash": "031988190c80e3c570734c03cd0c1be90e92f37c7e36a0f4a3e75b3e77aaefd7"
}
```

도면11

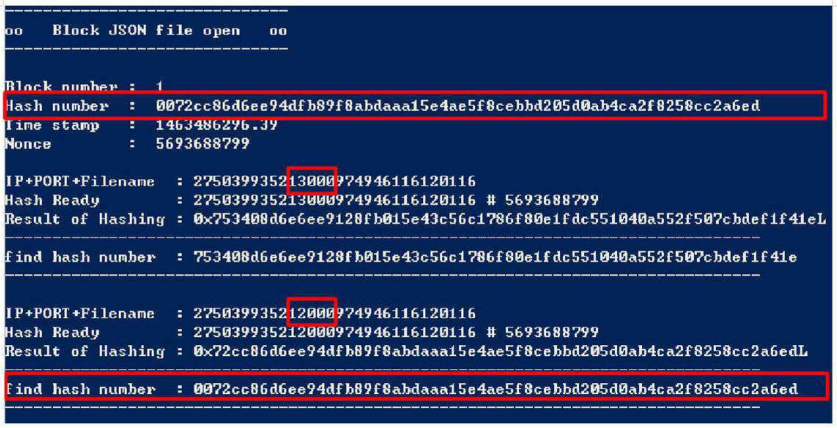

시작
S900
검색하고자 하는 데이터 명칭 입력
S910
입력된 데이터 명칭을 ASCII 코드로 변환, i=0, j=0
S920
블록체인의 블록[i]의 논스값과 블록해쉬값 판독
S922
IP List의 node[j]의 IP주소와 Port 번호 판독
S930
데이터 명칭, 판독된 node[i]의 IP 주소와 Port 번호, 블록[i]의 논스값을 이용하여 해쉬값 생성
S940
생성 해쉬값 == 블록[i] 해쉬값 ?
마지막노드?
j++
i++
마지막블록?
S950
Node[j]가 데이터 소유주로 확인함
데이터가 없음을 확인함
S960
종료

도면12

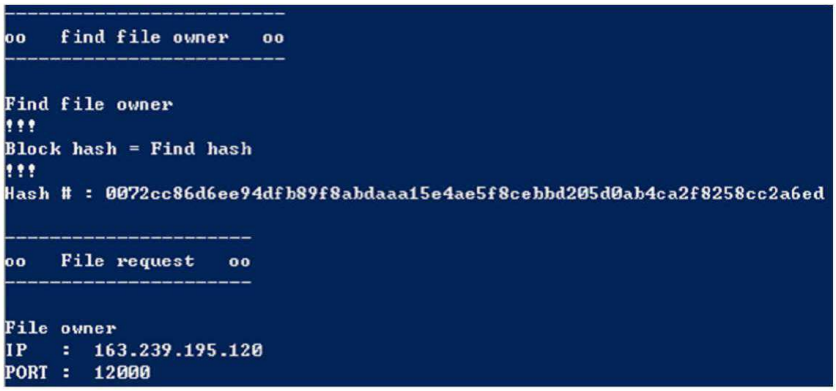

(S940)
Block hash
Time stamp
Nonce
Prev hash
0072cc86 ... eba94
5693688799
Owner Found
(S950)
Block
Block
Block
Block
Search
(S900)
Data name
ASCII
SHA-256
275039935212000
974946116120116
#
5693688799
0072cc86 ... eba94
(S920)
(S930)
(S910)
IP list
IP + Port number
블록의 nonce
hash

도면13

(a)


oo Block JSON file open oo
Block number : 1
Hash number : 0072cc86d6ee94dfb89f8abdaaa15e4ae5f8cebbd205d0ab4ca2f8258cc2a6ed
Time stamp : 1463486296.39
Nonce : 5693688799
IP+PORT+Filename : 27503993521300097494611612O116
Hash Ready : 27503993521300097494611612O116 # 5693688799
Result of Hashing : 0x753408d6e6ee9128fb015e43c56c1786f80e1fdc551040a552f507cbdef1f41eL
find hash number : 753408d6e6ee9128fb015e43c56c1786f80e1fdc551040a552f507cbdef1f41e
IP+PORT+Filename : 27503993521200097494611612O116
Hash Ready : 27503993521200097494611612O116 # 5693688799
Result of Hashing : 0x72cc86d6ee94dfb89f8abdaaa15e4ae5f8cebbd205d0ab4ca2f8258cc2a6edL
find hash number : 0072cc86d6ee94dfb89f8abdaaa15e4ae5f8cebbd205d0ab4ca2f8258cc2a6ed

(b)


oo Find file owner oo
Find file owner
!!!
Block hash = Find hash
!!!
Hash # : 0072cc86d6ee94dfb89f8abdaaa15e4ae5f8cebbd205d0ab4ca2f8258cc2a6ed
oo File request oo
File owner
IP : 163.239.195.120
PORT : 12000