(19) 대한민국특허청(KR)
(12) 등록특허공보(B1)

(45) 공고일자 2020년04월02일
(11) 등록번호 10-2096639
(24) 등록일자 2020년03월27일

(51) 국제특허분류(Int. Cl.)
H04L 9/06 (2006.01) *H04L 9/08* (2006.01)
(52) CPC특허분류
H04L 9/0618 (2013.01)
H04L 9/0825 (2013.01)
(21) 출원번호 10-2018-0174311(분할)
(22) 출원일자 2018년12월31일
심사청구일자 2018년12월31일
(65) 공개번호 10-2019-0139744
(43) 공개일자 2019년12월18일
(62) 원출원 특허 10-2018-0065964
원출원일자 2018년06월08일
심사청구일자 2018년06월08일
(56) 선행기술조사문헌
A. Menezes 외 2명, Handbook of Applied Cyrptography, Chapter 10,12, CRC Press (1996.)
KR1020080093635 A
KR1020170129866 A
KR1020170137388 A

(73) 특허권자
주식회사 미탭스플러스
서울특별시 강남구 테헤란로 134, 4층(역삼동, 포스코피앤에스타워)
(72) 발명자
김승연
서울특별시 서초구 신반포로 15길 19, 109동 204호 (반포동, 아크로리버파크)
임지순
경기도 성남시 중원구 도촌북로 78 ,510동703호(도촌동,휴먼시아섬마을)
이일영
서울특별시 송파구 올림픽로4길 42, 21동 601호 (잠실동, 우성아파트)
(74) 대리인
해움특허법인

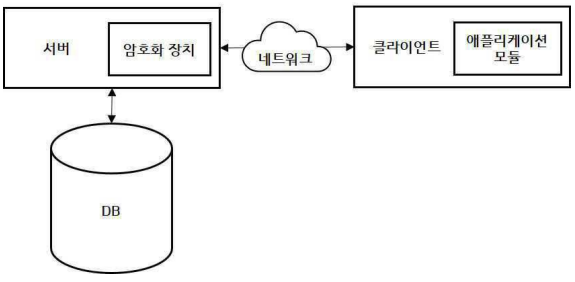
전체 청구항 수 : 총 1 항 심사관 : 양종필

(54) 발명의 명칭 **UUID를 이용한 블록체인에서 정보 조회 기록의 무결성을 위한 분산 원장 장치**

(57) 요 약

본 발명은 UUID를 이용한 블록체인에서 정보 조회 기록의 무결성을 위한 분산 원장 장치에 관한 것이다. 이를 위하여, 클라이언트로 송신할 특정 정보에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤, 클라이언트의 공개키 또는 클라이언트의 사용자의 공개키로 병합 정보를 암호화하여 암호 정보를 생성하는 암호화 모듈; 및 암호 정보를 클라이언트로 송신하고, 공개키에 대응되는 개인키를 이용한 암호 정보의 복호화를 통해 클라이언트에서 생성된 조회 정보가 포함된 리턴 쿼리를 수신하는 통신 모듈;이 제공될 수 있다.

대 표 도 - 도1



(52) CPC특허분류
H04L 9/0869 (2013.01)
H04L 2209/38 (2013.01)

명 세 서

청구범위

청구항 1

특정 계정의 디바이스로서 특정 정보를 이용하는 애플리케이션 모듈을 포함하는 클라이언트로 송신할 상기 특정 정보에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤 상기 클라이언트의 공개키 또는 상기 특정 계정의 공개키로 상기 병합 정보를 암호화하여 암호 정보를 생성하는 암호화 모듈; 및

상기 암호 정보를 상기 클라이언트로 송신하고, 상기 클라이언트에서 상기 공개키에 대응되는 상기 특정 계정의 개인키를 이용한 상기 암호 정보의 복호화를 통해 생성한 조회 정보를 수신하는 통신 모듈;

상기 조회 정보가 포함된 트랜잭션인 조회 정보 트랜잭션을 기록하는 블록체인을 저장하는 메모리 모듈; 및

상기 메모리 모듈과 동작 가능하도록 결합되고, 상기 조회 정보를 기록하는 컴퓨터 판독 가능한 프로그램 코드를 실행하는 처리 모듈;

을 포함하고,

상기 프로그램 코드는,

상기 조회 정보를 상기 특정 계정에 대해 수신하는 조회 정보 수신 단계;

상기 조회 정보가 포함된 트랜잭션인 상기 조회 정보 트랜잭션을 생성하는 트랜잭션 생성 단계; 및

상기 조회 정보 트랜잭션을 상기 블록체인이 포함되는 블록체인 분산 네트워크에 브로드캐스트 하는 배포 단계;

를 수행하는 컴퓨터 판독 가능한 프로그램 코드를 포함하고,

상기 조회 정보는, 상기 특정 계정의 상기 특정 정보의 조회에 관한 정보로서 상기 클라이언트의 UUID(universally unique identifier) 및 상기 특정 계정에 대한 정보를 포함하고,

상기 리턴 쿼리는, 상기 클라이언트의 상기 애플리케이션 모듈에서 상기 암호 정보가 복호화되는 경우 곧바로 실행되어 상기 조회 정보를 생성하고, 상기 통신 모듈로 상기 조회 정보를 송신하도록 구성된 코드를 포함하고,

상기 리턴 쿼리 및 상기 조회 정보에 의해 상기 특정 정보의 정보 조회 기록에 무결성이 확보되고,

상기 조회 정보 수신 단계에서 상기 클라이언트의 UUID가 기존과 달라지는 경우, 상기 암호화 모듈은 상기 특정 정보를 대칭키 암호화 방식인 비밀키(Secret key)로 암호화 한 뒤에 상기 리턴 쿼리를 병합하여 상기 병합 정보를 생성한 뒤 상기 공개키로 상기 병합 정보를 암호화하여 상기 암호 정보를 생성하는 하이브리드 암호 방식을 수행하도록 구성되고, 상기 통신 모듈은 상기 조회 정보가 수신되면 상기 비밀키를 상기 클라이언트에 송신하는 것을 특징으로 하는,

UUID를 이용한 블록체인에서 정보 조회 기록의 무결성을 위한 분산 원장 장치.

발명의 설명

기 술 분 야

[0001] 본 발명은 UUID를 이용한 블록체인에서 정보 조회 기록의 무결성을 위한 분산 원장 장치에 관한 것이다.

배 경 기 술

[0002] 기존의 암호 알고리즘 방식은 크게 대칭형 암호와 비대칭형 암호의 2가지로 나눌 수 있으며, 암호화에 사용하는 키 값과 복호화에 사용하는 키 값이 같은지 다른지를 기준으로 구분한다. 암호화할 때 사용한 키로 복호화를 할 수 있으면 대칭형 암호, 암호화 키와 복호화 키를 따로 구분하면 비대칭형 암호이다.

[0003] 대칭형 암호는 암호화할 때 사용하는 키와 복호화할 때 사용하는 키가 동일한 암호화 기법이다. 현재 사용되는 대칭형 암호화 알고리즘은 주로 파이스텔 네트워크(Feistel Network) / S-Box를 통하여 블럭 암호로 만들어졌지만 AES처럼 파이스텔 네트워크를 사용하지 않는 알고리즘도 있다. 현재 가장 보편적으로 쓰이는 암호화 방식은 현 미국 표준 방식인 AES이다. AES는 128~256비트 키를 적용할수 있어 보안성이 뛰어나며 공개된 알고리즘이라 누구나 사용할 수 있다. 그전에는 DES(Data Encryption Standard)라는 알고리즘이 1975년부터 사용되고 있었으나 너무 오래되어 취약점이 발견됨에 따라 이를 대체하기 위해 등장한 것이 바로 AES이다. 그 외에는 RC4, Twofish, Serpent, Blowfish, CAST5, 3DES, IDEA 등의 암호화 알고리즘이 존재하며 국내에서 개발된 SEED와 ARIA라는 알고리즘도 있다. AES는 현재 일반인들이 쉽게 접근할 수 있는 암호화 방식들 중에서 충분히 안전성이 있다고 알려진 암호로 암호화 키와 복호화 키가 동일하다. 즉, ABCDE라는 문자열을 QWERTY라는 키로 암호화했으면 복호화도 반드시 QWERTY로 해야 한다.

[0004] 대칭형 암호는 훌륭한 암호화 방식이기는 하지만 결정적인 문제가 존재한다. 바로 '키 배송'에 관한 문제로, 어떻게든 송신 측에서는 수신 측에 암호 키를 전달해야만 하고, 이 키가 배송 과정에서 털리면 아무리 뛰어난 암호화 알고리즘을 사용했더라도 속절없이 평문이 털리게 된다. 안전하게 평문을 전달하기 위해 만든 것이 암호문인데, 정작 키는 안전하게 전달할 방법이 없는 것. 이 키 배송에 대한 방법은 여러 가지 방법이 연구되었지만 발상의 전환으로 키 배송 문제를 해결한 방식이 비대칭형 암호이다.

[0006] 비대칭형 암호는 이름 그대로 암호화 키와 복호화 키가 다르다. 암호화를 하면 하나의 키 쌍이 생기고 이 두 개의 키는 수학적으로 밀접한 관계를 가지고 있다. 두 개의 키를 각각 키 A, 키 B라고 했을 때 키 A로 암호화한 암호문은 키 B로만 복호화할 수 있다. 따라서 이 중 하나의 키(B)만 비밀로 보호하고(이를 '개인키'라고 한다) 다른 하나의 키(A)는 공중에게 공개해도 관계가 없다(이를 '공개키'라고 부른다). 이렇게 둘 중 하나의 키는 반드시 공개되어야 통상적인 사용이 가능하므로 공개키 암호라고도 불린다. 공개키로 암호화한 암호문은 어차피 개인키를 가진 사람만이 풀어볼 수 있으므로 상호간에 공개키만 교환하고 상대의 공개키로 암호화를 해서 데이터를 교환하면 상대는 자신의 개인키로 복호화를 한다. 따라서 키 배송 문제는 근본적으로 발생하지 않는 것이다. 다만, 비대칭형 암호는 암/복호화가 대칭형 암호에 비해 현저하게 느리다는 문제점이 있다. 따라서 현실적으로는 비대칭형 암호를 이용해서 대칭형 암호의 키를 배송하고 실제 암호문은 대칭형 암호를 사용하는 식으로 상호보완적으로 이용하는 것이 일반적이다. 그리고 비대칭형 암호라고 약점이 없는 것은 아니어서 중간자 공격(MITM : Man In The Middle Attack)에는 취약하다. 해커가 중간에서 통신을 가로채어 수신자에게는 송신자인 척하고 송신자에게는 수신자인 척 해서 양쪽의 공개키와 실제 암호화에 사용되는 대칭키를 모두 얻어내는 기법. 참고로 이 중간자 공격을 미리 차단하기 위해서 사이트 인증서라는 것이 존재한다. 대표적인 비대칭형 암호에는 Diffie-Hellman 키 교환, DSS, ElGamal, ECC, RSA 등이 있다.

[0008] 비트코인, 이더리움 등과 같은 암호화 화폐 및 블록체인 기술도 이러한 기존의 암호 알고리즘 방식 및 Linked list 기술을 P2P 기술에 접목하여 개발된 것이다. 따라서, 블록체인 기술도 기존의 암호 알고리즘 방식에 기반하고 있다고 할 수 있다.

선행기술문헌

특허문헌

[0010] (특허문헌 0001) 등록특허 10-1825838, 데이터의 부분 암호화 방법, 데이터의 부분 복호화 방법 및 부분 암호화된 데이터를 복호화하는 프로그램을 저장하는 저장매체, 영남대학교 산학협력단

발명의 내용

해결하려는 과제

[0011] 하지만, 기존의 암호 알고리즘 방식은, 대칭키 또는 비대칭키가 해커에게 유포되지 않음을 전제로 하여 해커가 특정 평문을 조회하여 복호화하는 것을 방지하는데 집중하고 있고, 대칭키 또는 비대칭키가 해커에게 유포된 경우 평문의 조회에 대해 트래킹 하는 것은 매우 어려운 실정이었다.

[0012] 또한, 블록체인 기술을 이용하는 경우에도 상태의 변경만을 블록에 기록하여 linked list를 통해 기록의 무결성을 확보할 뿐, 상태나 정보의 조회에 대해 무결성이 확보된 기록이 수행되는 것은 매우 어려운 실정이었다. 오히려 블록체인 기술에서는 상태나 정보의 조회는 누구나 수행할 수 있도록 하는 것이 아이디어의 핵심이었다.

예를 들어, 이더리움의 경우, Ether scan이라는 웹페이지를 통해 이더리움 블록체인에 기록되는 모든 트랜잭션을 조회할 수 있도록 구성되어있다.

[0013] 따라서, 본 발명의 목적은 위와 같은 문제를 해결하기 위해, 정보 조회 기록의 무결성을 위한 암호화 장치 및 방법을 제공하는 데에 있다.

과제의 해결 수단

[0015] 이하 본 발명의 목적을 달성하기 위한 구체적 수단에 대하여 설명한다.

[0016] 본 발명의 목적은, 클라이언트로 송신할 특정 정보에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤, 상기 클라이언트의 공개키 또는 상기 클라이언트의 사용자의 공개키로 상기 병합 정보를 암호화하여 암호 정보를 생성하는 암호화 모듈; 및 상기 암호 정보를 상기 클라이언트로 송신하고, 상기 공개키에 대응되는 개인키를 이용한 상기 암호 정보의 복호화를 통해 상기 클라이언트에서 생성된 조회 정보가 포함된 상기 리턴 쿼리를 수신하는 통신 모듈;를 포함하는, 정보 조회 기록의 무결성을 위한 암호화 장치를 제공하여 달성될 수 있다.

[0017] 또한, 상기 암호화 모듈은, 상기 특정 정보를 대칭키 암호 방식인 비밀키로 암호화 한 뒤에 상기 리턴 쿼리를 병합하여 상기 병합 정보를 생성하고, 상기 통신 모듈은, 상기 조회 정보가 포함된 상기 리턴 쿼리를 수신한 뒤, 상기 클라이언트에 상기 비밀키를 송신하는 것을 특징으로 할 수 있다.

[0018] 또한, 상기 비밀키는, 상기 암호화 모듈에 의해 상기 특정 정보의 조회 요청시마다 random nonce로 생성되거나, 특정 시간 간격을 두고 random nonce로 생성되는 것을 특징으로 할 수 있다.

[0019] 또한, 상기 통신 모듈에서 상기 조회 정보를 포함한 상기 리턴 쿼리가 수신되는 경우, 상기 암호화 모듈에서 상기 조회 정보에 조회 시각을 포함시키고 메모리 모듈에 상기 조회 시각이 포함된 상기 조회 정보를 저장하는 것을 특징으로 할 수 있다.

[0020] 또한, 상기 암호화 모듈은, 분산 원장 장치의 프론트엔드 또는 상기 분산 원장 장치 내에 포함되고, 상기 조회 정보는 상기 분산 원장 장치에 의해 상기 조회 정보를 포함한 트랜잭션의 형태인 조회 정보 트랜잭션으로 생성되고, 상기 조회 정보 트랜잭션은 상기 분산 원장 장치의 블록체인에 기록되는 것을 특징으로 할 수 있다.

[0022] 본 발명의 다른 목적은, 암호화 모듈이, 클라이언트로 송신할 특정 정보에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤, 상기 클라이언트의 공개키 또는 상기 클라이언트의 사용자의 공개키로 상기 병합 정보를 암호화하여 암호 정보를 생성하는 암호화 단계; 통신 모듈이, 상기 암호 정보를 상기 클라이언트로 송신하는 암호 정보 송신 단계; 및 상기 통신 모듈이, 상기 공개키에 대응되는 개인키를 이용한 상기 암호 정보의 복호화를 통해 상기 클라이언트에서 생성된 조회 정보가 포함된 상기 리턴 쿼리를 수신하는 리턴 쿼리 수신 단계;를 포함하는, 정보 조회 기록의 무결성을 위한 암호화 방법을 제공하여 달성될 수 있다.

[0024] 본 발명의 다른 목적은, 클라이언트로 송신할 특정 계정의 특정 정보에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤, 상기 클라이언트의 공개키 또는 상기 특정 계정의 공개키로 상기 병합 정보를 암호화하여 암호 정보를 생성하는 암호화 모듈; 및 상기 암호 정보를 상기 클라이언트로 송신하고, 상기 공개키에 대응되는 개인키를 이용한 상기 암호 정보의 복호화를 통해 상기 클라이언트에서 생성된 조회 정보가 포함된 메시지 트랜잭션 형태의 상기 리턴 쿼리를 수신하는 통신 모듈; 상기 특정 계정을 포함하고, 상기 조회 정보가 포함된 메시지 트랜잭션인 조회 정보 메시지 트랜잭션을 수신하여 조회 정보 트랜잭션을 생성하는 블록체인을 저장하는 메모리 모듈; 및 상기 메모리 모듈과 동작 가능하도록 결합되고, 상기 특정 계정의 컴퓨터 판독 가능한 프로그램 코드를 실행하는 처리 모듈;을 포함하고, 상기 특정 계정의 상기 프로그램 코드는, 상기 조회 정보 메시지 트랜잭션을 상기 블록체인 내의 상기 특정 계정에 대해 수신하는 조회 정보 메시지 트랜잭션 수신 단계; 및 상기 조회 정보 트랜잭션을 생성하는 트랜잭션 생성 단계;를 수행하는 컴퓨터 판독 가능한 프로그램 코드를 포함하고, 상기 조회 정보 트랜잭션을 상기 블록체인이 포함되는 블록체인 분산 네트워크에 브로드캐스트 하는 것을 특징으로 하는, 블록체인에서 정보 조회 기록의 무결성을 위한 분산 원장 장치를 제공하여 달성될 수 있다.

[0026] 본 발며의 다른 목적은, 특정 계정의 디바이스로서 특정 정보를 이용하는 애플리케이션 모듈을 포함하는 클라이언트로 송신할 상기 특정 정보에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤 상기 클라이언트의 공개키 또는 상기 특정 계정의 공개키로 상기 병합 정보를 암호화하여 암호 정보를 생성하는 암호화 모듈; 및 상기 암호 정보를 상기 클라이언트로 송신하고, 상기 클라이언트에서 상기 공개키에 대응되는 상기 특정 계정의 개인키를 이용한 상기 암호 정보의 복호화를 통해 생성한 조회 정보를 수신하는 통신 모듈; 상기 조회 정보가 포함된 트랜잭션인 조회 정보 트랜잭션을 기록하는 블록체인을 저장하는 메모리 모듈; 및 상기 메모리 모듈과 동작 가능하

도록 결합되고, 상기 조회 정보를 기록하는 컴퓨터 판독 가능한 프로그램 코드를 실행하는 처리 모듈;을 포함하고, 상기 프로그램 코드는, 상기 조회 정보를 상기 특정 계정에 대해 수신하는 조회 정보 수신 단계; 상기 조회 정보가 포함된 트랜잭션인 상기 조회 정보 트랜잭션을 생성하는 트랜잭션 생성 단계; 및 상기 조회 정보 트랜잭션을 상기 블록체인이 포함되는 블록체인 분산 네트워크에 브로드캐스트 하는 배포 단계;를 수행하는 컴퓨터 판독 가능한 프로그램 코드를 포함하고, 상기 조회 정보는, 상기 특정 계정의 상기 특정 정보의 조회에 관한 정보로서 상기 클라이언트의 UUID(universally unique identifier) 및 상기 특정 계정에 대한 정보를 포함하고, 상기 리턴 쿼리는, 상기 클라이언트의 상기 애플리케이션 모듈에서 상기 암호 정보가 복호화되는 경우 곧바로 실행되어 상기 조회 정보를 생성하고, 상기 통신 모듈로 상기 조회 정보를 송신하도록 구성된 코드를 포함하고, 상기 리턴 쿼리 및 상기 조회 정보에 의해 상기 특정 정보의 정보 조회 기록에 무결성이 확보되고, 상기 조회 정보 수신 단계에서 상기 클라이언트의 UUID가 기존과 달라지는 경우, 상기 암호화 모듈은 상기 특정 정보를 대칭키 암호화 방식인 비밀키(Secret key)로 암호화 한 뒤에 상기 리턴 쿼리를 병합하여 상기 병합 정보를 생성한 뒤 상기 공개키로 상기 병합 정보를 암호화하여 상기 암호 정보를 생성하는 하이브리드 암호 방식을 수행하도록 구성되고, 상기 통신 모듈은 상기 조회 정보가 수신되면 상기 비밀키를 상기 클라이언트에 송신하는 것을 특징으로 하는, UUID를 이용한 블록체인에서 정보 조회 기록의 무결성을 위한 분산 원장 장치를 제공하여 달성될 수 있다.

발명의 효과

[0028] 상기한 바와 같이, 본 발명에 의하면 이하와 같은 효과가 있다.

[0029] 첫째, 본 발명의 일실시예에 따르면, 평문과 같은 정보의 조회 및 복호화 여부가 무결성이 유지된 채로 기록될 수 있으므로, 스니핑 등에 의해 개인키가 탈취된 상황에서도 정보 조회가 기록될 수 있는 효과가 발생될 수 있다.

도면의 간단한 설명

[0031] 본 명세서에 첨부되는 다음의 도면들은 본 발명의 바람직한 실시예를 예시하는 것이며, 발명의 상세한 설명과 함께 본 발명의 기술사상을 더욱 이해시키는 것이므로, 본 발명은 그러한 도면에 기재된 사항에만 한정되어 해석되어서는 아니 된다.

도 1은 본 발명의 일실시예에 따른 암호화 장치가 구성된 시스템을 도시한 모식도,

도 2, 3은 본 발명의 일실시예에 따른 암호화 방법을 도시한 흐름도,

도 4, 5는 본 발명의 일실시예에 따른 암호화 방법을 도시한 흐름도,

도 6은 본 발명의 일실시예에 따른 카드 게임 애플리케이션의 암호화 방법을 도시한 흐름도,

도 7은 본 발명의 일실시예에 따른 분산 원장 장치를 도시한 모식도,

도 8은 다양한 노드의 형태를 도시한 모식도,

도 9는 본 발명의 일실시예에 따른 블록체인을 이용한 Dapp 시스템을 도시한 모식도,

도 10은 본 발명의 일실시예에 따른 데이터의 흐름을 도시한 흐름도,

도 11은 본 발명의 일실시예에 따른 정보 조회 기록 무결성의 확보를 위한 분산 원장 방법을 도시한 흐름도,

도 12는 본 발명의 일실시예에 따른 IPFS가 이용되는 경우 블록체인의 정보 조회 기록 암호화 방법을 도시한 흐름도,

도 13은 본 발명의 일실시예에 따른 조회 정보 메시지 트랜잭션 정보의 데이터 구조를 도시한 모식도,

도 14, 15는 본 발명의 일 실시예에 따른 블록체인 통신 장치(2)와 분산 원장 장치(1)의 HTTP 통신을 도시한 모식도이다.

발명을 실시하기 위한 구체적인 내용

[0032] 이하 첨부된 도면을 참조하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 본 발명을 쉽게 실시할 수 있는 실시예를 상세히 설명한다. 다만, 본 발명의 바람직한 실시예에 대한 동작원리를 상세하게 설명함에 있어서 관련된 공지기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되

는 경우에는 그 상세한 설명을 생략한다.

[0033] 또한, 도면 전체에 걸쳐 유사한 기능 및 작용을 하는 부분에 대해서는 동일한 도면 부호를 사용한다. 명세서 전체에서, 특정 부분이 다른 부분과 연결되어 있다고 할 때, 이는 직접적으로 연결되어 있는 경우뿐만 아니라, 그 중간에 다른 소자를 사이에 두고, 간접적으로 연결되어 있는 경우도 포함한다. 또한, 특정 구성요소를 포함한다는 것은 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라, 다른 구성요소를 더 포함할 수 있는 것을 의미한다.

[0035] **정보 조회 기록의 무결성을 위한 암호화 장치 및 방법**

[0036] 도 1은 본 발명의 일실시예에 따른 암호화 장치가 구성된 시스템을 도시한 모식도이다. 도 1에 따른 본 발명의 일실시예에 따른 암호화 장치가 구성된 시스템은, 특정 애플리케이션을 서비스하는 서버, 암호화 장치, DB, 클라이언트, 애플리케이션 모듈을 포함할 수 있다.

[0037] 서버는, 특정 애플리케이션을 클라이언트에 서비스하기 위해 해당 클라이언트와 통신하여 정보를 송수신하는 구성이다. 서버에는 본 발명의 일실시예에 따른 암호화 장치가 구성되어 클라이언트로 송신할 정보(특정 정보, 예를 들어 평문 또는 이미지와 같은 기타 정보)를 암호화하여 암호 정보를 생성되고, 서버는 암호화된 특정 정보인 암호 정보를 클라이언트로 송신하게 된다. 본 발명의 일실시예에 따른 암호화 장치의 설명을 위해 암호화 장치가 서버에 구성된다고 기재하였으나, 본 발명의 범위는 이에 한정되지 않고 클라이언트의 애플리케이션 모듈에 암호화 장치가 구성되어 다른 클라이언트 또는 서버에 정보를 암호화하여 송신하는 경우를 포함할 수 있다.

[0038] 암호화 장치는 클라이언트로 송신할 정보(특정 정보)에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤, 해당 클라이언트의 공개키(public key) 또는 해당 클라이언트의 사용자의 공개키(public key)로 암호화를 하여 암호 정보를 생성하는 구성이다. 생성된 암호 정보는 서버에 의해 클라이언트로 송신된다. 본 발명의 일실시예에 따른 리턴 쿼리는 클라이언트의 애플리케이션 모듈에서 암호 정보가 복호화되는 경우 곧바로 실행되어 서버로 쿼리를 송신하도록 구성된 코드이다. 본 발명의 일실시예에 따른 암호화 장치 및 리턴 쿼리가 병합된 병합 정보에 의하면 복호화가 이루어진 즉시, 특정 정보가 생성된 서버로 클라이언트에서 리턴 쿼리가 송신되도록 구성되기 때문에, 해커와 같은 제3자가 암호 정보를 스니핑(Sniffing)하여 취득하더라도 특정 정보의 조회 여부를 서버가 파악하고 기록할 수 있게 되는 효과가 발생된다. 또한, 리턴 쿼리가 서버로 송신될 때 클라이언트의 UUID(universally unique identifier) 또는 사용자 id와 함께 서버로 송신되도록 구성될 수 있다. 이에 따르면, 어떤 클라이언트 디바이스 또는 어떤 사용자가 해당 특정 정보를 조회하였는지에 대해 트래킹하고 기록할 수 있게 되는 효과가 발생된다.

[0039] DB는, 서버와 연결되어 특정 애플리케이션에 대한 데이터, 사용자에 대한 데이터 등을 저장하고 출력하는 구성이다.

[0040] 클라이언트는, 사용자의 디바이스이고 특정 정보를 이용하는 애플리케이션 모듈을 포함하는 구성이다. 클라이언트는 암호 정보 및 리턴 쿼리를 송수신하는 통신 모듈, 애플리케이션 모듈 및 여러 정보를 저장하는 메모리 모듈, 애플리케이션 모듈의 프로그램 코드를 처리하는 처리 모듈을 포함할 수 있다.

[0041] 애플리케이션 모듈은, 클라이언트에 구성되고 특정 정보를 이용하여 특정 서비스를 사용자에게 제공하는 프로그램 코드를 포함한 구성이다. 또한, 본 발명의 일실시예에 따른 애플리케이션 모듈은 서버에서 암호 정보를 수신하면 이를 클라이언트 또는 사용자의 개인키(private key)로 복호화하여 특정 정보 및 리턴 쿼리의 병합인 병합 정보를 생성하고, 조회 정보(클라이언트 UUID, 사용자 계정 정보, 조회 시각, 특정 정보)를 포함한 리턴 쿼리를 서버로 송신하여 복호화 여부를 서버에 전달하도록 구성될 수 있다.

[0043] 도 2, 3은 본 발명의 일실시예에 따른 암호화 방법을 도시한 흐름도이다. 도 2, 3에 도시된 바와 같이, 본 발명의 일실시예에 따른 암호화 방법은, 특정 정보 암호화 단계(S10), 암호 정보 송신 단계(S11), 복호화 단계(S12), 리턴 쿼리 수신 단계(S13)를 포함할 수 있다.

[0044] 특정 정보 암호화 단계(S10)는 암호화 장치가 클라이언트로 송신할 정보(특정 정보)에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤, 해당 클라이언트의 공개키(public key) 또는 해당 클라이언트의 사용자의 공개키(public key)로 암호화를 하여 암호 정보를 생성하는 단계이다.

[0045] 암호 정보 송신 단계(S11)는 서버가 생성된 암호 정보를 클라이언트로 송신하는 단계이다.

[0046] 복호화 단계(S12)는 암호 정보를 수신한 클라이언트의 애플리케이션 모듈에서 암호 정보를 클라이언트 또는 사

용자의 개인키(private key)로 복호화하여 특정 정보 및 리턴 쿼리의 병합인 병합 정보를 생성하는 단계이다.

[0047] 리턴 쿼리 수신 단계(S13)는 클라이언트가 조회 정보(클라이언트 UUID, 사용자 계정 정보, 조회 시각, 특정 정보)를 포함한 리턴 쿼리를 서버로 송신하여 복호화 여부를 서버에 전달하는 단계이다. 본 발명의 다른 실시예에 따르면, 조회 시각의 경우 최초에 조회 정보에 포함되지 않았다가, 서버가 조회 정보를 수신한 시각을 기초로 서버가 조회 정보 내에 조회 시각을 포함시키도록 하는 것이 가능하다. 이에 따르면, 서버의 시각에 맞게 조회 시각이 표준화되어 조회 시각의 timestamping이 가능해지는 효과가 발생된다.

[0048] 본 발명의 일실시예에 따른 암호화 방법에 따르면 정보 조회 기록에도 무결성이 확보되는 효과가 발생된다.

[0050] 본 발명의 다른 실시예에 따르면, 클라이언트에서의 복호화 이후 리턴 쿼리가 클라이언트에서 서버로 송신되지 않으면 특정 정보가 조회되지 못하도록, 복호화 이후 리턴 쿼리의 송신과 특정 정보의 조회 사이에 의존관계를 형성할 수 있다. 도 4, 5는 본 발명의 일실시예에 따른 암호화 방법을 도시한 흐름도이다. 도 4, 5에 도시된 바와 같이, 본 발명의 일실시예에 따른 암호화 방법은, 특정 정보 암호화 단계(S20), 암호 정보 송신 단계(S21), 복호화 단계(S22), 리턴 쿼리 수신 단계(S23), 비밀키 송신 단계(S24)를 포함할 수 있다.

[0051] 특정 정보 암호화 단계(S20)는 암호화 장치가 클라이언트로 송신할 정보(특정 정보)를 대칭키 암호화 방식인 비밀키(Secret key)로 암호화 한 뒤에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤, 해당 클라이언트의 공개키(public key) 또는 해당 클라이언트의 사용자의 공개키(public key)로 암호화를 하여 암호 정보를 생성하는 단계이다.

[0052] 암호 정보 송신 단계(S21)는 서버가 생성된 암호 정보를 클라이언트로 송신하는 단계이다.

[0053] 복호화 단계(S22)는 암호 정보를 수신한 클라이언트의 애플리케이션 모듈에서 암호 정보를 클라이언트 또는 사용자의 개인키(private key)로 복호화하여 비밀키로 암호화된 특정 정보 및 리턴 쿼리의 병합인 병합 정보를 생성하는 단계이다.

[0054] 리턴 쿼리 수신 단계(S23)는 클라이언트가 조회 정보(클라이언트 UUID, 사용자 계정 정보, 조회 시각, 특정 정보)를 포함한 리턴 쿼리를 서버로 송신하여 복호화 여부를 서버에 전달하는 단계이다.

[0055] 비밀키 송신 단계(S24)는 암호화 장치가 클라이언트로 비밀키를 송신하여 비밀키로 암호화된 특정 정보를 복호화할 수 있도록 하는 단계이다.

[0056] 본 발명의 실시예에 따른 비밀키는 대칭키 암호 방식이며, 정보 조회시마다 random nonce로 생성되거나 특정 시간 간격을 두고 random nonce로 생성될 수 있다. 비밀키가 random nonce로 구성되는 경우, 해커가 비밀키를 재차 사용하는 것이 불가능해지는 효과가 발생된다.

[0057] 본 발명의 다른 실시예에 따르면, 대칭키 암호 방식과 비대칭키 암호 방식이 하이브리드로 적용되면서 해커가 클라이언트 또는 사용자의 개인키를 확보하여 암호 정보를 스니핑한 뒤 인터넷 연결을 끊는 다던지 애플리케이션 모듈의 코드를 변조한다던지의 행위를 통해 특정 정보 조회 후 리턴 쿼리가 서버로 송신되지 않도록 하는 경우를 방지할 수 있게 되는 효과가 발생된다. 즉, 본 발명의 다른 실시예에 따르면 해커가 암호 정보 복호화 후 리턴 쿼리가 서버로 송신되지 않도록 하면 특정 정보를 조회하는 것이 불가능하게 되는, 복호화 이후 리턴 쿼리의 송신과 특정 정보의 조회 사이에 의존관계가 형성된다.

[0058] 또한, 본 발명의 다른 실시예에 따른 암호화 방법의 경우, 본 발명의 일실시예에 따른 암호화 방법의 보안 강화 버전으로 이용될 수 있다. 즉, 본 발명의 일실시예에 따른 암호화 방법으로 특정 정보의 조회가 기록되다가, 특정 정보의 조회를 요청한 클라이언트의 UUID가 기존과 달라지는 경우 보안 단계가 상승하여 본 발명의 다른 실시예에 따른 암호화 방법으로 특정 정보의 조회가 기록 되도록 구성될 수 있다. 이에 따르면, 암호화 방법이 합리적으로 구성되게 되므로 트래픽의 속도나 복잡도를 크게 악화되지 않으면서도 보안성이 향상되는 효과가 발생된다.

[0060] 본 발명의 일실시예에 따른 암호화 방법을 이용한 애플리케이션의 예시로 카드 게임 애플리케이션(예를 들면, 세븐 포커)을 예로 들 수 있다.

[0061] 도 6은 본 발명의 일실시예에 따른 카드 게임 애플리케이션의 암호화 방법을 도시한 흐름도이다. 도 6에 도시된 바와 같이, 카드 게임 애플리케이션의 코드에 의해 특정 사용자에 대한 카드 정보가 카드 게임 제공 서버에서 생성되면, 암호화 장치에 의해 카드 정보에 리턴 쿼리가 병합된 병합 정보가 생성되고, 암호화 장치는 생성된 병합 정보를 사용자의 공개키로 암호화하여 암호 정보를 생성하게 된다. 생성된 암호 정보는 카드 게임 제공 서

버에서 사용자의 클라이언트로 송신되게 되고, 암호 정보를 수신한 클라이언트의 카드 게임 애플리케이션 모듈은 자신의 개인키를 이용하여 암호 정보를 복호화하게 된다. 클라이언트의 카드 게임 애플리케이션 모듈은 암호 정보의 복호화를 통해 사용자의 카드 정보 및 리턴 쿼리를 생성하게 되고, 조회 정보(클라이언트 UUID, 조회 시각, 카드 정보)를 포함하여 리턴 쿼리를 카드 게임 제공 서버에 송신할 수 있다.

[0062] 이에 따르면, 개인키를 획득한 해커가 카드 정보를 스니핑한다고 하더라도, 해커의 클라이언트에서 카드 정보의 조회가 이루어지는 즉시 리턴 쿼리가 서버로 송신되어 카드 정보 조회 기록의 무결성이 확보되는 효과가 발생된다.

[0064] **정보 조회 기록의 무결성을 위한 분산 원장 장치**

[0065] 정보 조회 기록의 무결성을 위한 분산 원장 장치에 관하여, 도 7은 본 발명의 일실시예에 따른 분산 원장 장치를 도시한 모식도이다. 도 7에 도시된 바와 같이, 본 발명의 일실시예에 따른 분산 원장 장치(1)는 블록체인 분산 네트워크(100)의 노드 중 하나를 의미할 수 있고, 통신 모듈(10), 처리 모듈(11), 메모리 모듈(12)을 포함할 수 있다.

[0066] 블록체인 분산 네트워크(100)는 블록체인의 분산 원장을 저장하고 있는 복수대의 노드로 구성된 P2P 분산 네트워크를 의미한다. 블록체인 분산 네트워크(100)는 적어도 일부의 노드가 동일한 처리를 할 수 있도록 구성되기 때문에 일부의 노드가 Shut down 되거나 변조되더라도 시스템 전체에는 영향을 주지 않는 특징을 가진다. 노드는 블록체인 분산 네트워크(100)에 연결된 모든 컴퓨팅 장치를 의미할 수 있다. 이러한 노드는 디지털 지갑, 블록체인 복사본, 검증 엔진, 채굴 엔진, P2P 네트워크 배포 기능(브로드캐스트) 등을 포함할 수 있으며, 조금 더 Light한 기능들로만 구성된 클라이언트들도 포함될 수 있다. 본 발명의 일실시예에 따른 블록체인 분산 네트워크(100)는 Pure P2P와 슈퍼 노드(Super Node)를 포함하는 Hybrid P2P를 포함할 수 있다. 특히 본 발명의 일실시예에 따른 노드는 PC/모바일 애플리케이션의 소스코드와 함께 클라이언트에 설치될 수 있다.

[0067] 노드와 관련하여, 도 8은 다양한 노드의 형태를 도시한 모식도이다. 도 8에 도시된 바와 같이, 레퍼런스 클라이언트(Reference client)는 사용자들의 디지털 지갑 관리 모듈, 합의 알고리즘이 작업 증명(POW, Proof of Work)인 경우 블록 채굴(Block Mining)을 위한 마이닝 모듈(Mining Module), 전체 블록체인 중 전부 또는 적어도 일부의 블록을 저장하는 블록체인 데이터베이스(Blockchain Database), 트랜잭션(Transaction)을 블록체인 분산 네트워크에 브로드캐스트(Broadcast)하는 네트워크 라우팅 모듈(Network Routing Module)을 포함하는 노드를 의미할 수 있고, 예를 들어 Bitcoin Core의 Client 등을 의미할 수 있다. 풀노드(Full Node)는 블록체인 데이터베이스, 네트워크 라우팅 모듈을 포함하는 노드를 의미할 수 있다. 솔로 마이너 노드(Solo Miner Node)는, 마이닝 모듈, 블록체인 데이터베이스, 네트워크 라우팅 모듈을 포함하는 노드를 의미할 수 있다. 마이닝 노드(Mining Node)는, 마이닝 풀(Mining Pool)의 노드를 의미하는 풀 마이닝 노드(Pool Mining Node)에 연결되는 게이트웨이 라우터(Gateway Router)와 마이닝 모듈을 포함하는 가벼운 노드를 의미할 수 있다. 라이트웨이트 월렛 노드(Lightweight Wallet Node)는 일반적으로 블록체인의 헤더정보만 저장하고, 디지털 지갑 관리 모듈을 보유하여 사용자의 디지털 지갑을 저장하며, 네트워크 라우팅 모듈을 포함하긴 하지만 블록체인 데이터베이스를 포함하지 않아 트랜잭션 생성이나 블록체인 분산 네트워크에 접근하기 위해서는 제3자가 소유한 서버에 의존하는 가벼운 노드를 의미할 수 있다. 본 발명의 일실시예에 따른 분산 원장 장치(1)는 위의 노드들 중 블록체인 데이터베이스와 네트워크 라우팅 모듈을 포함하는 노드들을 의미할 수 있다. 또한, 본 발명의 일실시예에 따른 노드가 마이닝 모듈을 포함하는 경우에는 POW(Proof of Work) 또는 POS(Proof of Stake) 방식으로 사용자 클라이언트가 마이닝을 수행하도록 구성할 수 있다.

[0068] 통신 모듈(10)은 트랜잭션(Transaction)을 블록체인 분산 네트워크에 브로드캐스트(Broadcast)하는 네트워크 라우팅 모듈(Network Routing Module)이나 풀 마이닝 노드(Pool Mining Node)에 연결되는 게이트웨이 라우터(Gateway Router) 등을 의미할 수 있다.

[0069] 처리 모듈(11)은 메모리 모듈(12)에 저장된 분산 원장인 블록체인의 블록에 저장되어 있는 트랜잭션의 내용을 처리하는 모듈이다. 본 발명의 일실시예에 따른 처리 모듈(11)은 Ethereum Virtual Machine과 같은 가상 머신(VM)으로 구성될 수 있다. 이러한 가상 머신은 예를 들어, Mutan, LLL, Serpent, Solidity 등과 같은 상위 레벨 언어로 만들어진 코드(스마트 계약, Smart Contract)가 컴파일되어 생성되는 Byte Code를 실행하기 위한 Runtime이고, OPCODE 및 Stack 외에 Memory 및 Storage를 사용하는 주체이기도 하다.

[0070] 메모리 모듈(12)은 분산 원장을 저장하는 모듈로서, 전체 블록체인 중 전부 또는 일부의 블록을 저장하는 블록체인 데이터베이스(Blockchain Database)를 의미할 수 있다.

[0072] 도 9는 본 발명의 일실시예에 따른 블록체인을 이용한 Dapp 시스템을 도시한 모식도이다. 도 9에 도시된 바와 같이, 본 발명의 일실시예에 따른 블록체인을 이용한 Dapp 시스템은 클라이언트(200), Dapp(400), IPFS(410), 서버(420), DB(430), 블록체인(30)을 포함할 수 있고, 본 발명의 일실시예에 따른 암호화 장치가 Dapp 내에 구성될 수 있다.

[0073] 클라이언트(200)는 사용자가 Dapp(Decenteralized Application)을 이용하기 위한 스마트폰과 같은 클라이언트를 의미한다. 클라이언트를 통해 사용자는 블록체인 상의 특정 정보를 수신할 수 있게 된다. 본 발명의 일실시예에 따른 클라이언트는 본 발명의 일실시예에 다른 애플리케이션 모듈을 포함하고, 암호 정보를 수신하여 개인키로 복호화하는 방식으로 특정 정보를 확보하게 되고, 암호 정보를 복호화하여 특정 정보와 함께 생성된 리턴 쿼리를 다시 Dapp으로 송신하게 된다. 이에 따라, 정보 조회 기록의 무결성이 확보되는 효과가 발생된다.

[0074] Dapp(400)은 블록체인 애플리케이션의 프론트엔드(Web Frontend)를 의미하며, Dapp의 웹 프론트엔드는 HTML, CSS, 자바스크립트(web3js의 비중이 큼)의 조합으로 구성될 수 있다. 클라이언트(200)는 이 프론트엔드 애플리케이션을 통해 블록체인(30), IPFS(410), nodeJS 서버(420)와 상호작용하게 된다. Dapp은 본 발명의 일실시예에 따른 암호화 장치를 포함하고, 클라이언트(200)에서 특정 정보에 대한 조회를 요청(query)하는 경우 Dapp의 암호화 장치에서 특정 정보에 리턴 쿼리를 병합하여 병합 정보를 생성하고, 생성된 병합 정보를 클라이언트의 사용자 계정 공개키로 암호화 하여 암호 정보를 생성한 뒤, 암호 정보를 Dapp에서 클라이언트로 송신하게 된다.

[0075] IPFS(410, InterPlanetary File System)는 모든 컴퓨팅 장치를 동일한 파일 시스템으로 연결하려고 하는 P2P 분산 파일 시스템이다. Git 레포지트리의 오브젝트를 교환하는 단일 비트토렌트 스웜 (파일의 고유 식별자와 파일을 소유하고 있는 피어 목록)으로도 볼 수 있다. 즉, IPFS는 Cotent의 해시값을 주소로 활용하는 content addressed 하이퍼 링크를 이용하여 높은 처리량을 가진 content addressed 블록 스토리지 모델을 제공한다. 이는 버전 관리되는 파일 시스템, 블록 체인, 영구적인 웹 페이지를 구축할 수 있는 자료 구조인 일반화된 Markle DAG의 형태를 가질 수 있다. IPFS는 분산 해시 테이블과 인센티브화된 블록 교환과 자체 인증 네임스페이스를 합친 것이다. IPFS를 활용하면 특정 노드의 예상치 못한 종료나 파괴에 의해서 단일 실패 지점이 발생되지 않으며, 각자의 노드들은 서로 신뢰할 필요가 없게 되는 효과가 발생된다. 본 발명의 일실시에에 따라 사용자가 Dapp(400)에 특정 파일을 올리면, 프론트엔드인 Dapp(400)은 특정 파일을 IPFS(410)에 업로드한 후 업로드된 파일의 해시를 블록체인(30) 및 DB(430)에 저장하게 된다.

[0076] 서버(420)는 프론트엔드인 Dapp(400)이 DB(430)와 통신하기 위한 Off Chain의 백엔드 서버이다. API를 통해 프론트엔드에서 데이터베이스의 정보를 쿼리하고 받을 수 있게 구성된다. 또한, DB(430)에서 다양한 조건으로 등록된 정보의 검색 및 리스팅(Listing)을 수행하는 구성이다.

[0077] DB(430)는 서버(420)와 연결되어 정보를 저장하고 인덱싱하는 Off Chain 데이터베이스이다. 예를 들어, 몽고DB 등의 데이터베이스로 구성될 수 있다. 블록체인에 모든 정보를 올린다 해도, 정보를 Dapp(400) 상에서 디스플레이하고 다양한 필터를 적용할 때마다(특정 카테고리 상품만 보거나, 곧 마감되는 상품을 조회하는 등) 블록체인에 쿼리를 날리는 것은 비효율적이다. 따라서, 본 발명의 일실시예에서는 블록체인(30) 대신 Off Chain의 DB(430)를 사용해서 정보를 저장하고 쿼리하게 된다.

[0078] 블록체인(30)은 분산 원장 장치(1)의 메모리 모듈(12)에 저장되는 Linked List인 복수개의 블록으로 구성되어 있고, 특정 블록 이후부터는 특정 계정을 포함한다. 본 발명의 일실시예에 따른 블록체인(30)에 대해서는 이하에서 설명한다.

[0080] 도 10은 본 발명의 일실시예에 따른 데이터의 흐름을 도시한 흐름도이다. 도 10에 도시된 바와 같이, 블록체인(30)에는 특정 계정(20)이 포함될 수 있다. 특정 계정(20)은 EoA(External Owned Accouunt) 또는 CA(Contract Account)로 구성될 수 있고, 특정 정보의 조회 요청이 Dapp에 수신되면 본 발명의 일실시예에 따른 암호화 방법으로 특정 정보를 암호화하여 클라이언트에 송신하는 계정을 의미한다.

[0082] 이러한 스마트 계약의 Code는 예를 들어, Ethereum 의 경우, Solidity, Serpent, LLL, Mutan의 언어로 쓰여질 수 있는데, 현재는 Solidity가 주로 사용되고 있으며 문법은 JavaScript와 유사하다. Smart Contract 는 "변수", "구조체" 및 "함수"를 포함하여 처리 모듈(11)에 의해 처리되는 프로그램 코드이다. 이러한 Smart Contract Code 는 Compile 과정을 거쳐 Byte Code로 변환될 수 있다. Byte code는 Solidity Realtime Compiler를 통해 컴파일될 수 있다. Solidity의 Byte code는 모두 16진수로 된 코드이며, Solidity에서 이 Byte code를 수신 주소 없이 Payload (data:) 로 할당하여 Blockchain에 Transaction을 배포하면, Miner에 의해 Block이 생성되고, 이러한 Transaction은 Contract Creation Transaction으로 간주되어, Transaction Receipt의

contractAddress: 필드에 생성(배포)된 Contract의 주소를 넣어서 리턴해주게 된다.

[0083] Smart Contract 개발환경은 개발도구와 Compiler 까지를 포함한 범위를 포함할 수 있다. 예를 들어, Solidity의 경우, Code를 작성하고 컴파일하면 모든 컴파일러는 [Byte Code] 와 [Function Signature], [ABI]를 출력하게 된다.

[0084] Byte Code는 이미 위에서 설명한 것과 같이, Smart Contract Code를 컴파일 한 결과이며, Blockchain에 Contract Creation Transaction을 발생시켜 배포하는 경우, Contract로의 Message Tx(Transaction의 줄임말)이 발생되는 경우, Contract로의 Call/Query가 발생되는 경우를 통해 분산 원장 장치(1, 이더리움의 경우 EVM) 위에서 실행된다.

[0085] FunctionSignature는 Contract 함수 호출 시 인터페이스로 이용되는 것이고, Contract 함수 이름을 SHA3 암호 해시 함수로 해시한 4바이트 값의 Hash값이다.

[0086] ABI(Application Binary Interface)는 특정 언어나 플랫폼에 종속되지 않은 방식으로 기술된 Application Interface를 의미한다. ABI 정의를 컴파일러 혹은 ABI Generator가 출력해내는데, ABI 에는 Smart Contract의 함수와 Parameter에 대한 Meta data가 정의되어있다. ABI 를 갖고 JavaScript 언어 기반의 어플리케이션을 만들 때 객체를 만들게 할 수 있고, 쉽게 그 객체의 Method를 호출하는것 만으로 Contract의 함수가 호출되도록 할 수 있는 것이다. 현재 Ethereum 은 web3.js와 함께 JavaScript 응용에서 쉽게 ABI 로 객체를 만들어 사용하도록 지원하며, 1.4.0 이후의 go-ethereum 에서는 Go Native 언어 기반의 응용에서 Smart Contract 를 쉽게 Binding 가능하도록, ABI 기반으로 Go Code를 생성해주는 ABIGen을 제공하고 있다.

[0087] 이러한 Smart Contract 개발 환경은 Blockchain Engine과 연결되어 Contract Creation/Deployment, Message Tx, Call/Query를 전달할 수 있는데, 이러한 Blockchain Engine은 본 발명의 일실시예에 따른 분산 원장 장치(1)를 의미하고, 이더리움의 예에서는 geth나 parity, eth와 같은 Ethereum Node를 의미한다. 결국 모든 Smart Contract 와 관련한 Transaction 처리와 Contract 실행을 위한 EVM과 같은 가상 머신은 분산 원장 장치(1)와 같은 Node 위에 구성되어 있다.

[0088] 이러한 Smart Contract 개발 환경에서 ABI는 Applications와 연결될 수 있다. Smart Contract는 Logic만을 갖고 있고, 사용자나 외부 시스템과의 상호작용을 위해서는 Application이 필요하다. HTML+CSS+JavaScript, Application Server, Wallet 등의 Application은 예를 들어, Ethereum과의 Interface를 통해 Smart Contract와 상호작용하는 구성이다.

[0089] Smart Contract Code는 크게 [Creation/Deployment] [Invoke by Message] [Call]로 구분될 수 있다. 본 발명의 일실시예에 따른 특정 계정의 코드도 계정 생성, 메시지 트랜잭션, 조회 요청(Query) 등으로 구분될 수 있다.

[0091] 특정 계정(20)과 연결된 클라이언트(블록체인 통신 장치)에서 블록체인의 특정 계정에 특정 정보의 조회를 요청(Query)하면, 블록체인의 가상머신 또는 Dapp에서 해당 특정 정보를 특정 계정에서 조회하여 리턴 쿼리와 병합하여 병합 정보를 생성하고, 생성된 병합 정보를 특정 계정의 공개키로 암호화한 뒤, 암호화된 병합 정보인 암호 정보를 특정 계정(20)과 연결된 클라이언트에 송신하게 된다. 암호화된 병합 정보인 암호 정보를 수신한 클라이언트는 특정 계정의 개인키(private key)로 암호 정보를 복호화하여 특정 정보 및 리턴 쿼리를 확보한 뒤, 리턴 쿼리를 메시지 트랜잭션의 형태로 블록체인의 특정 계정에 송신하게 된다. 특정 계정이 리턴 쿼리를 수신하게 되면 특정 정보의 조회에 대한 정보인 조회 정보(특정 정보, 조회 시간, 조회 클라이언트 UUID 등)을 포함하여 트랜잭션 정보를 생성하고, 새롭게 생성되는 블록에 저장하게 되며, 블록체인 분산 네트워크에 global 하게 배포하여 블록체인 분산 네트워크의 모든 노드가 새로운 공유재화 등록에 의한 상태변경을 공유할 수 있다.

[0092] 조회 정보의 트랜잭션 정보에는, 특정 계정 주소 정보, 매개변수(parameter) 중 하나인 조회 정보(예를 들어, 특정 정보, 조회 시간, 조회 클라이언트 UUID)를 포함할 수 있다. 특정 계정(20)이 조회 정보의 트랜잭션 정보를 수신하면 블록에 저장하고 블록체인 분산 네트워크에 global 하게 배포하여 블록체인 분산 네트워크의 모든 노드가 새로운 조회 정보에 의한 상태변경을 공유할 수 있다.

[0094] 도 11은 본 발명의 일실시예에 따른 정보 조회 기록 무결성의 확보를 위한 분산 원장 방법을 도시한 흐름도이다. 도 11에 도시된 바와 같이, 본 발명의 일실시예에 따른 처리 모듈(11)의 정보 조회 기록 무결성의 확보를 위한 분산 원장 방법은, 조회 요청 수신 단계(S30), 암호 정보 생성 단계(S31), 암호 정보 송신 단계(S32), 조회 정보 트랜잭션 생성 단계(S33), 배포 단계(S34)를 포함하여 트랜잭션 정보를 처리할 수 있다.

[0095] 조회 요청 수신 단계(S30)는 Dapp(400)이 특정 계정(20)의 특정 정보에 대한 클라이언트(200)의 조회 요청을 수

신하는 단계이다.

[0096] 암호 정보 생성 단계(S31)는 Dapp(400) 또는 블록체인 가상머신(예를 들어, EVM)의 암호화 장치가 블록체인(30)에서 특정 정보를 검색 및 수신하고, 특정 정보에 리턴 쿼리를 병합하여 병합 정보를 생성하며, 병합 정보를 사용자 또는 클라이언트(200)의 공개키로 암호화하여 암호 정보를 생성하는 단계이다.

[0097] 암호 정보 송신 단계(S32)는 생성된 암호 정보를 Dapp(400)이 클라이언트(200)에 송신하는 단계이다.

[0098] 조회 정보 트랜잭션 생성 단계(S33)는 클라이언트(200)에서 암호 정보를 사용자 또는 클라이언트의 개인키로 복호화하여 특정 정보를 조회하고 리턴 쿼리가 생성되면, 클라이언트(200)가 조회 정보(클라이언트 UUID, 특정 정보 해시값 및 조회 시간)를 포함하여 Dapp(400)으로 메시지 트랜잭션의 형태로 송신하게 되며, Dapp(400)은 조회 정보 메시지 트랜잭션을 수신하여 조회 정보에 대한 트랜잭션을 생성하고 특정 계정(20)의 스토리지(Storage) 또는 메모리(Memory)에 저장하는 단계이다. Contract Account 는 Storage라고 불리는 Persistent 저장소를 포함할 수 있다. 스토리지(Storage)에서는 Key-Value 맵 구조로 32바이트 키를 32바이트 값으로 맵핑하도록 되어있다. 특정 Smart Contract는 자기 자신 이외의 Contract의 Storage를 읽거나 쓸 수 없을 수 있다. Memory는 Smart Contract가 Message Call이 있을 때마다 최신의 Instance를 얻을 수 있는 공간으로 구성될 수 있다. 메모리(Memory)에서는 Byte 레벨로 읽고 쓸 수 있으나 32바이트 단위 Chunk로 저장될 수 있다. 즉, 메모리(Memory)에서는 1이라는 값을 저장하면 32바이트 (256비트) 공간에 저장될 수 있다. 예를 들어, EVM은 총 1024개의 Instruction Set (OPCODE) 를 담을 수 있는 Stack을 포함하며, 256비트의 word (값)을 가질 수 있다.

[0099] 배포 단계(S34)는 조회 정보 트랜잭션 생성 단계(S13)에 의해 변경된 상태(State)인 조회 정보 트랜잭션 정보를 분산 원장 장치(1)가 블록체인 분산 네트워크(100)의 전체 노드로 배포하여 브로드캐스트하는 단계이다.

[0100] 본 발명의 일실시예에 따른 분산 원장 방법에 따르면, 정보 조회 기록에 무결성이 확보되게 되므로, 정보 변조뿐만 아니라 정보 조회까지도 무결성을 확보할 수 있는 블록체인을 구성할 수 있게 되는 효과가 발생된다.

[0101] 본 발명의 다른 실시예에 따르면, 조회 시각의 경우 최초에 조회 정보에 포함되지 않았다가, Dapp이 조회 정보를 수신한 시각을 기초로 Dapp이 조회 정보 내에 조회 시각을 포함시키도록 하는 것이 가능하다. 이에 따르면, Dapp의 시각에 맞게 조회 시각이 표준화되어 조회 시각의 timestamping이 가능해지는 효과가 발생된다.

[0103] 도 12는 본 발명의 일실시에에 따른 IPFS가 이용되는 경우 블록체인의 정보 조회 기록 암호화 방법을 도시한 흐름도이다. 도 12에 도시된 바와 같이, 본 발명의 일실시예에 따르면 IPFS가 이용되는 경우에도 아래의 단계로 정보 조회가 기록 될 수 있다.

[0104] (1) 클라이언트(200)에서 Dapp(400)으로 특정 정보(이미지, 영상, 음성, 텍스트 등)를 조회하는 요청을 송신

[0105] (2) Dapp(400)에서 블록체인(30)에 특정 정보의 해시값을 요청하고, 블록체인(30)에서 특정 정보의 해시값을 수신

[0106] (3) Dapp(400)에서 특정 정보의 해시값을 토대로 IPFS에 특정 정보를 요청 및 수신

[0107] (4) Dapp(400) 또는 블록체인 가상머신(예를 들어, EVM)에 구성된 암호화 장치가 특정 정보에 리턴 쿼리를 병합하여 병합 정보 생성

[0108] (5) 암호화 장치가 병합 정보를 사용자의 공개키로 암호화하여 암호 정보 생성

[0109] (6) Dapp(400)에서 클라이언트(200)로 암호 정보를 송신

[0110] (7) 클라이언트(200)에서 암호 정보를 수신하고, 암호 정보를 사용자의 개인키로 복호화하여 병합 정보 생성

[0111] (8) 클라이언트(200)에서 조회 정보(클라이언트 UUID, 특정 정보 해시값 및 조회 시간)를 포함한 리턴 쿼리를 Dapp(400)으로 송신

[0112] (9) Dapp(400)에서 조회 정보를 블록체인(30)에 저장하도록 메시지 트랜잭션 송신

[0113] 위와 같은 단계에 의해, 승인되지 않은 해커의 클라이언트가 사용자의 개인키를 스니핑하여 암호 정보를 복호화하고 특정 정보를 획득하게 된다고 하더라도 어떤 클라이언트가 어떤 정보를 언제 조회하였는지에 대해 블록체인 내에 조회 정보가 무결성이 확보된 채로 기록되는 효과가 발생된다. 특히, IPFS나 Swarm 등의 P2P 스토리지를 이용하는 경우에도 정보 조회 기록의 무결성이 확보되는 효과가 발생된다.

[0115] 도 13은 본 발명의 일실시예에 따른 조회 정보 메시지 트랜잭션 정보의 데이터 구조를 도시한 모식도이다. 도

13에 도시된 바와 같이, 본 발명의 일실시예에 따른 메모리 모듈(12)은 전부 또는 일부의 블록체인을 저장할 수 있고, 각 블록 헤더에는 앞 블록 헤더의 해시 값(hash), 해답 값(nonce), 트랜잭션 그룹의 해시 값과 그 밖에 생성된 시간에 대한 정보인 Timestamp(미도시), 채굴 난이도를 의미하는 Difficulty(미도시), 블록의 넘버를 의미하는 Block Number(미도시) 등이 포함될 수 있다. 각 블록 바디(Contents)에는 적어도 하나 이상의 트랜잭션 정보(특히, 본 발명의 일실시예에 따른 조회 정보 트랜잭션 정보)가 포함될 수 있다.

[0116] 본 발명의 일실시예에 따른 조회 정보 트랜잭션 정보에는 특정 계정 주소 정보, 클라이언트 UUID, 사용자 정보, 조회 시각, 특정 정보 등이 포함될 수 있다.

[0119] *본 발명의 일실시예에 따른 블록체인 통신 장치와 관련하여, 도 14, 15는 본 발명의 일 실시예에 따른 블록체인 통신 장치(2)와 분산 원장 장치(1)의 HTTP 통신을 도시한 모식도이다. 도 14, 15에 도시된 바와 같이, 본 발명의 일실시예에 따른 트랜잭션 정보 브로드캐스트 장치는, 예를 들어, Solidity로 코딩되는 경우, JavaScript 기반의 Web3.Js API를 사용하고, 내부적으로는 JSON-RPC API를 사용할 수 있다. 이를 통해, Browser, Node.JS, Mist 등으로 블록체인 통신 장치(2)를 구성할 수 있다. 블록체인 통신 장치(2)의 주체는 EOA나 CA인 특정 계정이 될 수 있다.

[0120] 도 14에 도시된 바와 같이, 본 발명의 일실시예에 따른 블록체인 통신 장치(2)에서 특정 계정에 조회 정보 트랜잭션을 저장하도록 조회 정보 메시지 트랜잭션을 분산 원장 장치(1)에 JSON-RPC API를 이용하여 요청할 수 있다. 조회 정보 메시지 트랜잭션을 수신한 분산 원장 장치(1)는 본 발명의 일실시예에 따른 암호화 방법에 따라 생성되는 조회 정보 트랜잭션을 특정 계정의 스토리지 또는 메모리에 저장한 뒤에 블록체인 분산 네트워크(100)에 배포(broadcast, propagation) 할 수 있다.

[0121] 또한, 도 15에 도시된 바와 같이, 블록체인 통신 장치(2)에서 개인키를 통해 조회 정보 메시지 트랜잭션의 트랜잭션 정보가 서명되어 블록체인 분산 네트워크(100)의 각 노드들(분산 원장 장치, 1)에 브로드캐스트 될 수 있다. 특정 노드에서 블록체인 분산 네트워크(100)로 배포된 조회 정보 트랜잭션은 특정 계정의 메모리에 저장되어 있다가 해당 특정 노드 또는 다른 특정 노드가 새로운 블록(New block)을 마이닝(mining)하게 되면 해당 블록에 기록되게 된다.

[0123] 이상에서 설명한 바와 같이, 본 발명이 속하는 기술 분야의 통상의 기술자는 본 발명이 그 기술적 사상이나 필수적 특징을 변경하지 않고서 다른 구체적인 형태로 실시될 수 있다는 것을 이해할 수 있을 것이다. 그러므로 상술한 실시예들은 모든 면에서 예시적인 것이며 한정적인 것이 아닌 것으로서 이해해야만 한다. 본 발명의 범위는 상세한 설명보다는 후술하는 특허청구범위에 의하여 나타내어지며, 특허청구범위의 의미 및 범위 그리고 등가 개념으로부터 도출되는 모든 변경 또는 변형된 형태가 본 발명의 범위에 포함하는 것으로 해석되어야 한다.

[0124] 본 명세서 내에 기술된 특징들 및 장점들은 모두를 포함하지 않으며, 특히 많은 추가적인 특징들 및 장점들이 도면들, 명세서, 및 청구항들을 고려하여 당업자에게 명백해질 것이다. 더욱이, 본 명세서에 사용된 언어는 주로 읽기 쉽도록 그리고 교시의 목적으로 선택되었고, 본 발명의 주제를 묘사하거나 제한하기 위해 선택되지 않을 수도 있다는 것을 주의해야 한다.

[0125] 본 발명의 실시예들의 상기한 설명은 예시의 목적으로 제시되었다; 이는 개시된 정확한 형태로 본 발명을 제한하거나, 빠뜨리는 것 없이 만들려고 의도한 것이 아니다. 당업자는 상기한 개시에 비추어 많은 수정 및 변형이 가능하다는 것을 이해할 수 있다.

[0126] 본 설명의 일부는 연산의 기호 표현 및 알고리즘에 관한 본 발명의 실시예들을 기술한다. 이러한 알고리즘적 설명 및 표현은, 일반적으로 그들의 작업의 핵심을 효율적으로 다른 당업자에게 전달하기 위해 데이터 처리 분야의 당업자에 의해 사용된다. 이러한 동작은 기능적, 연산적, 또는 논리적으로 설명되지만, 컴퓨터나 이와 동등한 전기 회로, 마이크로코드 등에 의해 구현될 것으로 이해된다. 나아가, 또한 이것은 모듈로서의 이러한 동작의 배열을 나타내기 위해, 때때로 일반성의 상실 없이 편리하게 입증된다. 상기 기술된 동작 및 그들의 연관된 모듈은 소프트웨어, 펌웨어, 하드웨어, 또는 이들의 임의의 조합 내에서 구현될 수 있다.

[0127] 여기서 기술된 임의의 단계, 동작, 또는 프로세스는, 하나 이상의 하드웨어 또는 소프트웨어 모듈과 함께 단독으로 또는 다른 장치와 조합하여 수행되거나 구현될 수 있다. 일 실시예에서, 소프트웨어 모듈은 컴퓨터 프로그램 코드를 포함하는 컴퓨터-판독 가능 매체로 구성되는 컴퓨터 프로그램 제품과 함께 구현되고, 컴퓨터 프로그램 코드는 기술된 임의의 또는 모든 공정, 단계, 또는 동작을 수행하기 위한 컴퓨터 프로세서에 의해 실행될 수 있다.

[0128] 또한, 본 발명의 실시예들은, 여기서의 동작을 수행하기 위한 장치와 관련될 수 있다. 이들 장치는 요구되는 목적을 위해 특별히 제작될 수 있고/있거나, 컴퓨터 내에 저장된 컴퓨터 프로그램에 의해 선택적으로 활성화되거나 재구성되는 일반적-목적의 연산 장치를 포함할 수 있다. 이러한 컴퓨터 프로그램은, 유형의 컴퓨터 판독가능 저장 매체 또는 전자 명령어를 저장하기 위해 적합한 임의의 유형의 미디어 내에 저장될 수 있고, 컴퓨터 시스템 버스에 결합될 수 있다. 나아가, 본 명세서에 참조되는 임의의 연산 시스템은 단일 프로세서를 포함할 수 있거나, 증가한 연산 능력을 위한 다중 프로세서 디자인을 채택한 구조가 될 수 있다.

[0129] 마지막으로, 본 명세서에 사용된 언어는 주로 읽기 쉽도록 그리고 교시의 목적으로 선택되었고, 본 발명의 주제를 묘사하거나 제한하기 위해 선택되지 않을 수 있다.

[0130] 그러므로 본 발명의 범위는 상세한 설명에 의해 한정되지 않고, 이를 기반으로 하는 출원의 임의의 청구항들에 의해 한정된다. 따라서, 본 발명의 실시예들의 개시는 예시적인 것이며, 이하의 청구항에 기재된 본 발명의 범위를 제한하는 것은 아니다.

부호의 설명

[0132] 1: 분산원장 장치

2: 블록체인 통신 장치

10: 통신 모듈

11: 처리 모듈

12: 메모리 모듈

20: 특정 계정

30: 블록체인

100: 블록체인 분산 네트워크

200: 클라이언트

400: Dapp

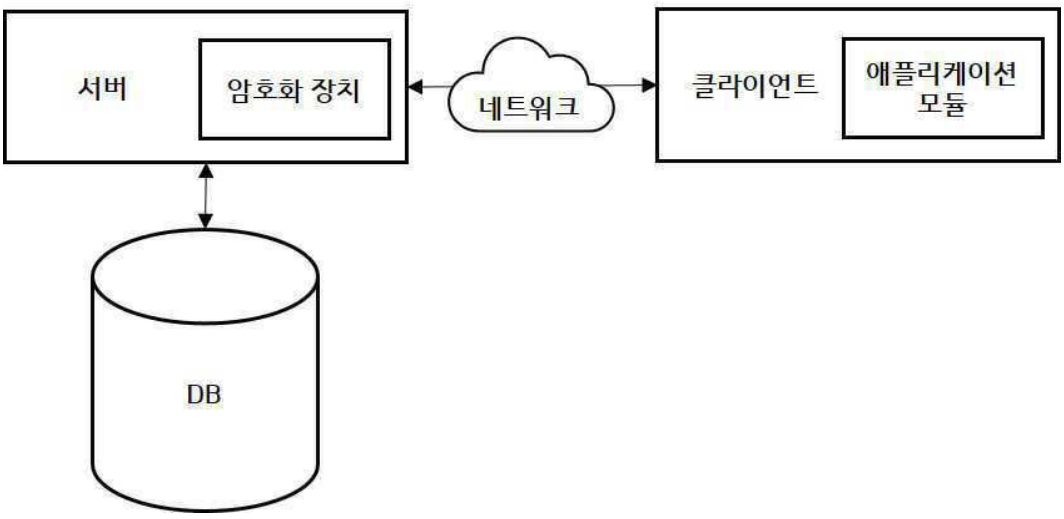
410: IPFS

420: 서버

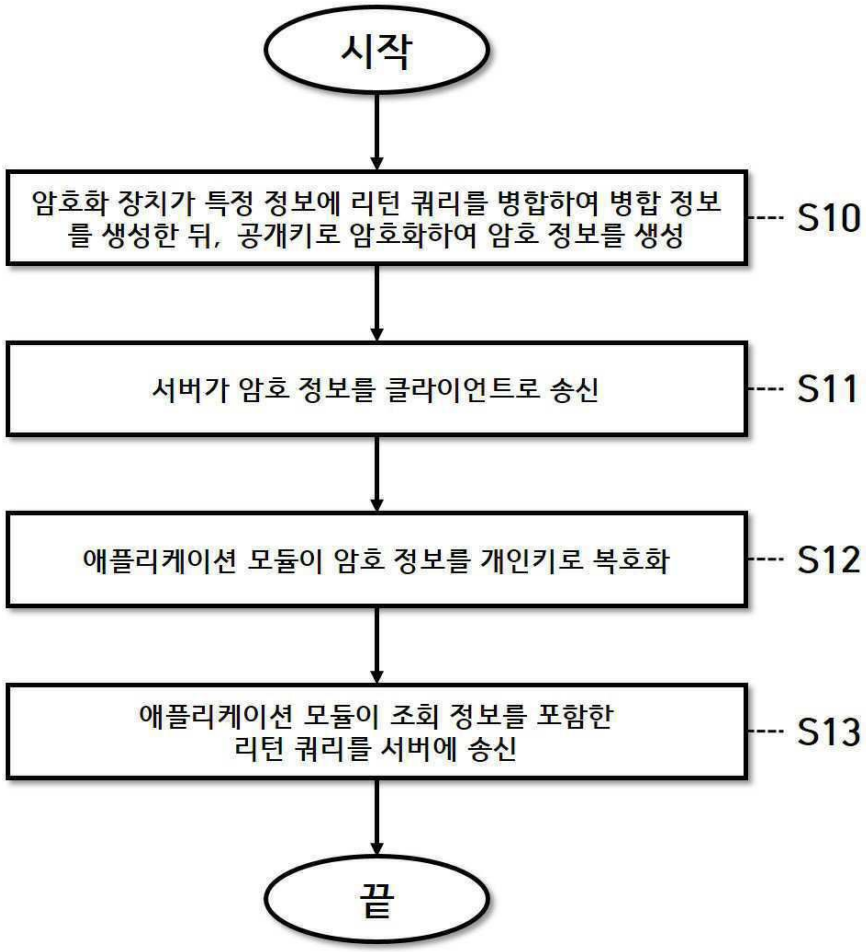
430: DB

도면

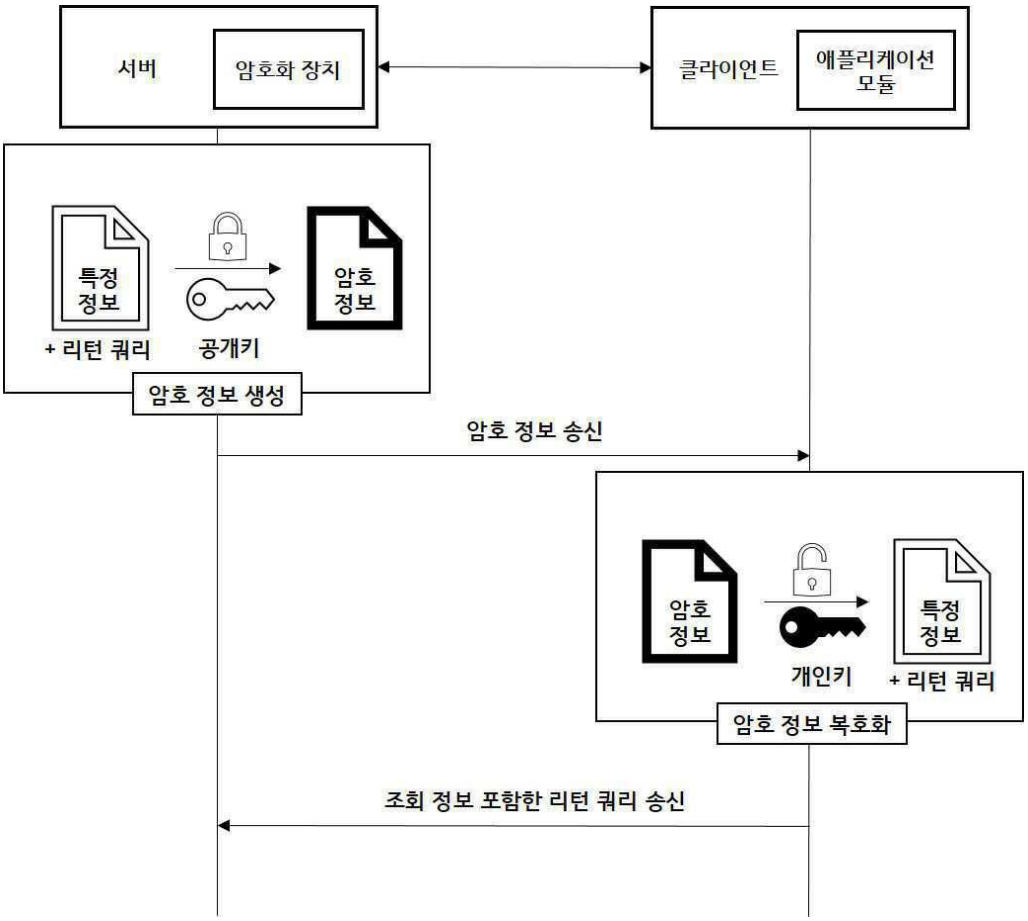
도면1



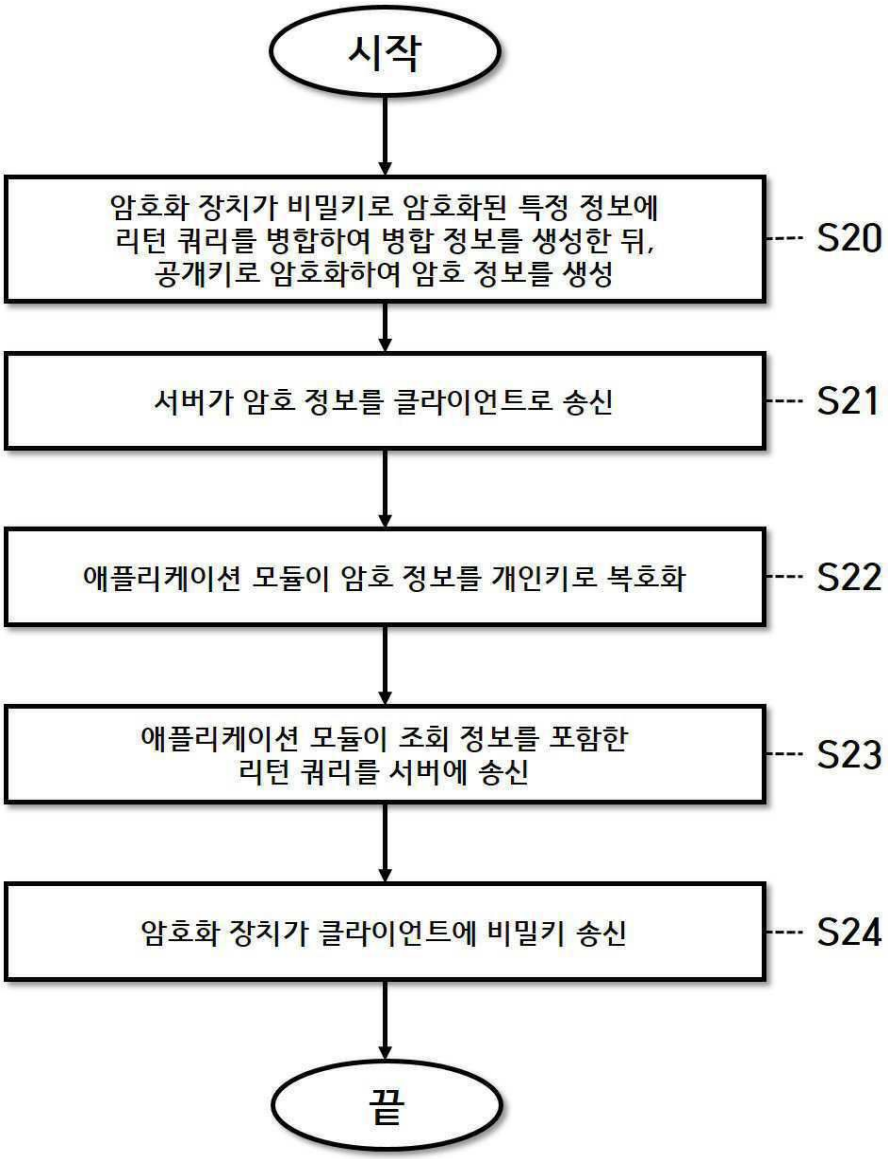
도면2


시작
암호화 장치가 특정 정보에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤, 공개키로 암호화하여 암호 정보를 생성
S10
서버가 암호 정보를 클라이언트로 송신
S11
애플리케이션 모듈이 암호 정보를 개인키로 복호화
S12
애플리케이션 모듈이 조회 정보를 포함한 리턴 쿼리를 서버에 송신
S13
끝

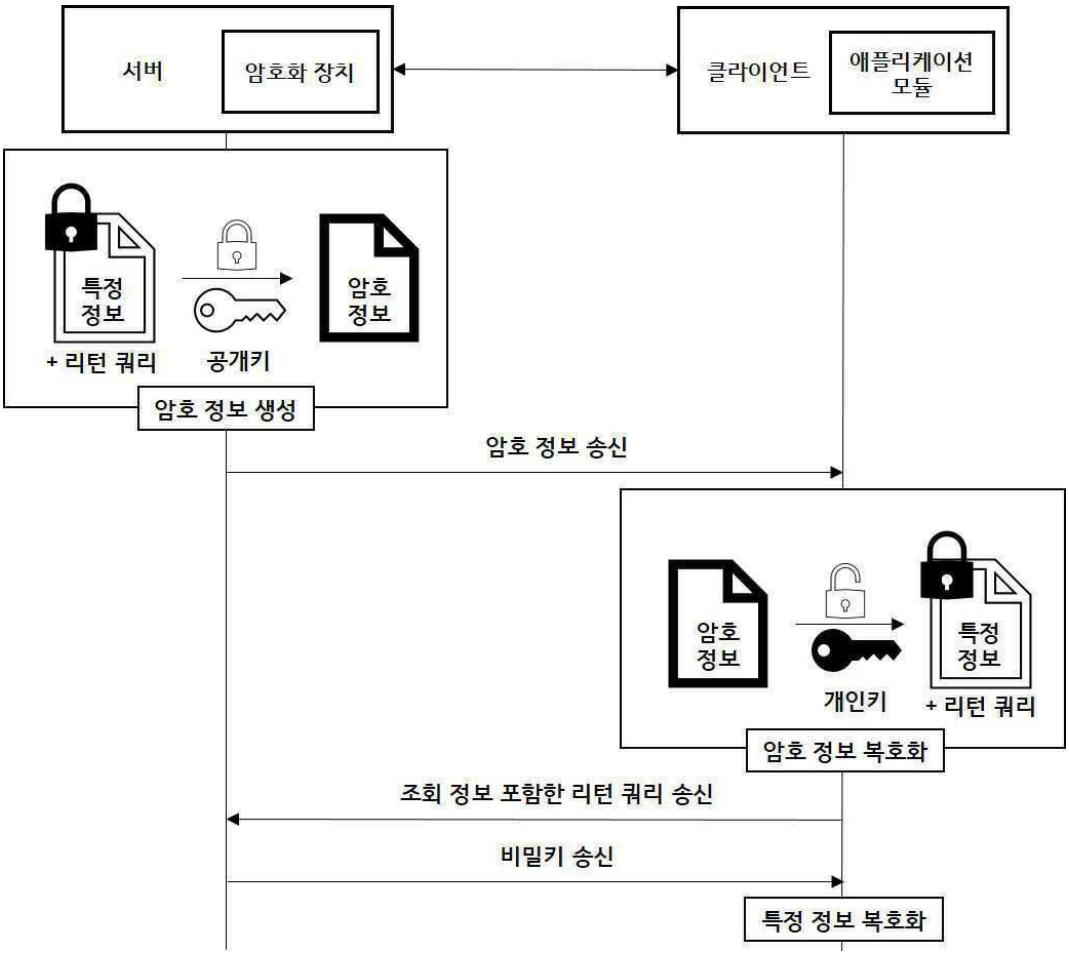
도면3


서버
암호화 장치
클라이언트
애플리케이션 모듈
특정 정보
+ 리턴 쿼리
공개키
암호 정보
암호 정보 생성
암호 정보 송신
암호 정보
개인키
특정 정보
+ 리턴 쿼리
암호 정보 복호화
조회 정보 포함한 리턴 쿼리 송신

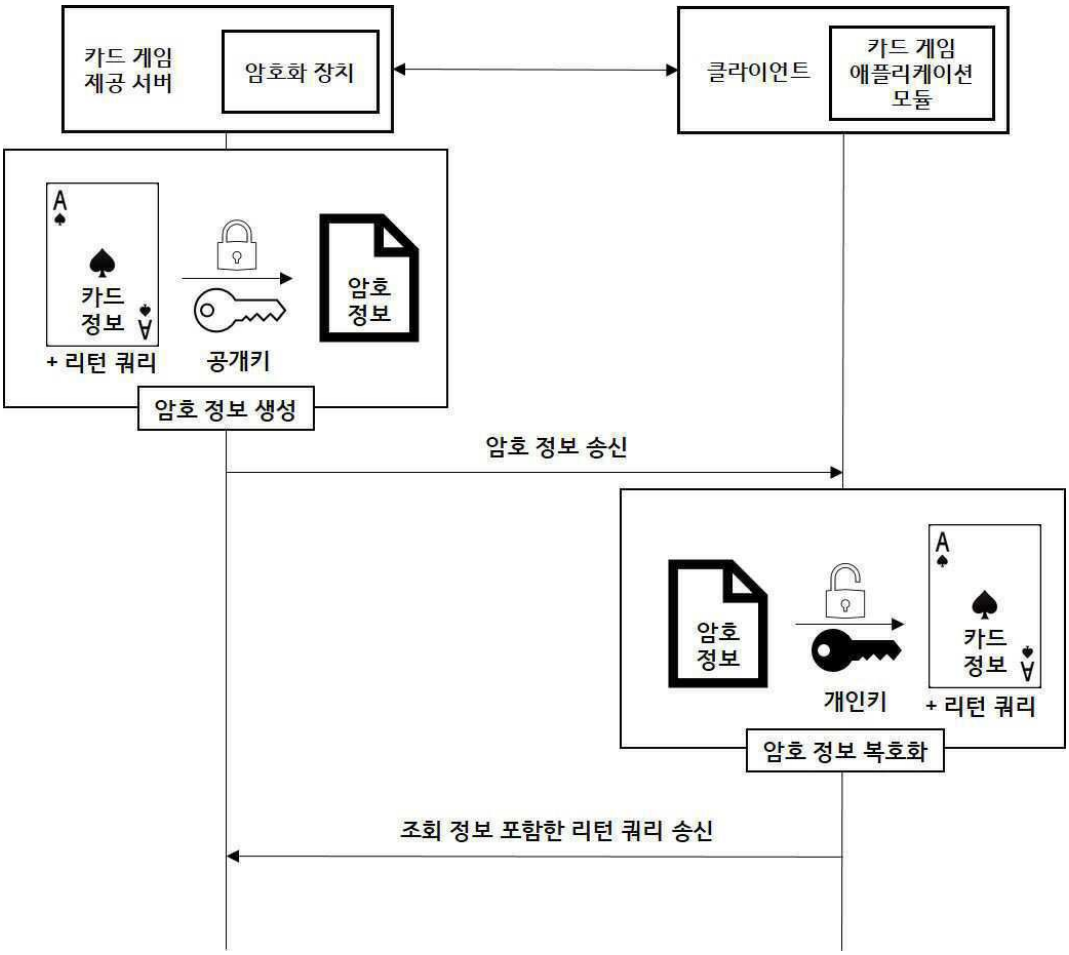
도면4


시작
암호화 장치가 비밀키로 암호화된 특정 정보에 리턴 쿼리를 병합하여 병합 정보를 생성한 뒤, 공개키로 암호화하여 암호 정보를 생성
S20
서버가 암호 정보를 클라이언트로 송신
S21
애플리케이션 모듈이 암호 정보를 개인키로 복호화
S22
애플리케이션 모듈이 조회 정보를 포함한 리턴 쿼리를 서버에 송신
S23
암호화 장치가 클라이언트에 비밀키 송신
S24
끝

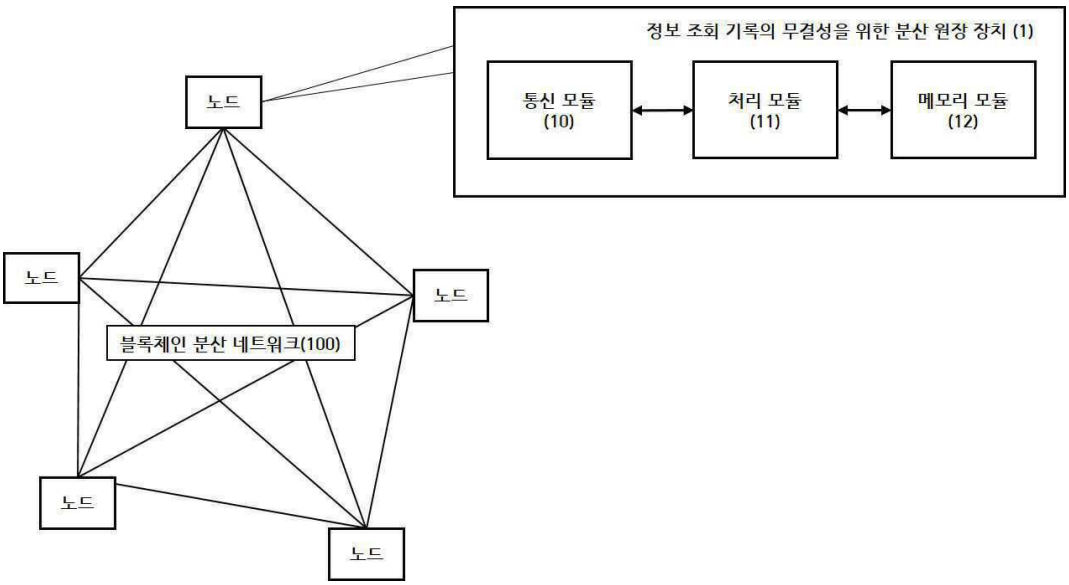
도면5


서버
암호화 장치
클라이언트
애플리케이션 모듈
특정 정보
+ 리턴 쿼리
공개키
암호 정보
암호 정보 생성
암호 정보 송신
암호 정보
개인키
특정 정보
+ 리턴 쿼리
암호 정보 복호화
조회 정보 포함한 리턴 쿼리 송신
비밀키 송신
특정 정보 복호화

도면6


카드 게임
제공 서버
암호화 장치
클라이언트
카드 게임
애플리케이션
모듈
카드
정보
+ 리턴 쿼리
공개키
암호
정보
암호 정보 생성
암호 정보 송신
암호
정보
개인키
카드
정보
+ 리턴 쿼리
암호 정보 복호화
조회 정보 포함한 리턴 쿼리 송신

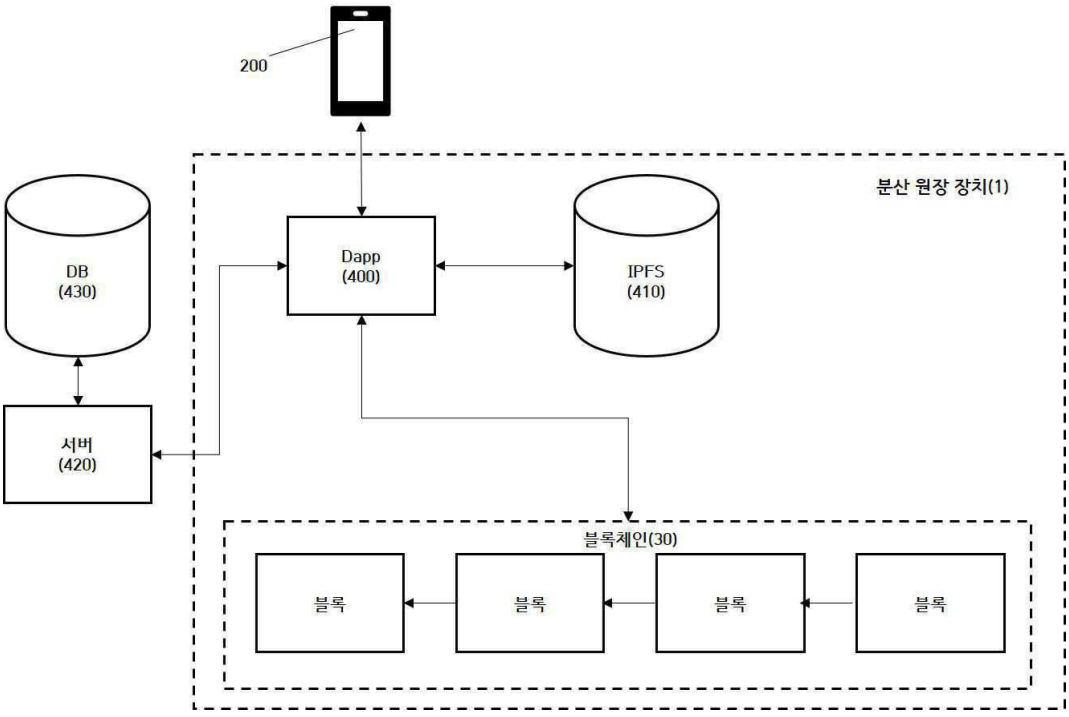
도면7


정보 조회 기록의 무결성을 위한 분산 원장 장치 (1)
통신 모듈
(10)
처리 모듈
(11)
메모리 모듈
(12)
노드
노드
노드
블록체인 분산 네트워크(100)
노드
노드

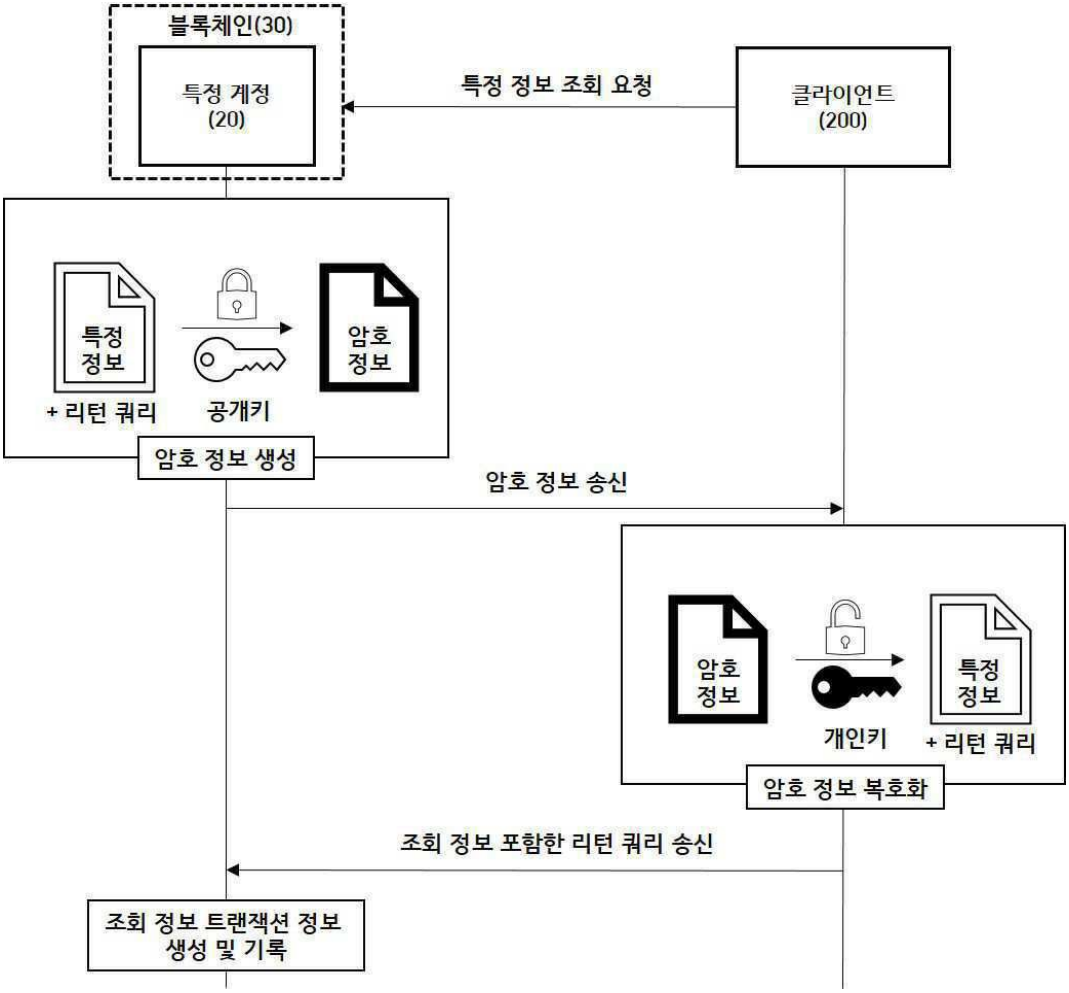
도면8


레퍼런스 클라이언트
디지털 지갑 모듈
마이닝 모듈
블록체인 데이터베이스
네트워크 라우팅 모듈
풀 노드
블록체인 데이터베이스
네트워크 라우팅 모듈
솔로 마이너 노드
마이닝 모듈
블록체인 데이터베이스
네트워크 라우팅 모듈
마이닝 노드
마이닝 모듈
게이트웨이 라우터
라이트웨이트 월렛 노드
디지털 지갑 모듈
마이닝 모듈
블록체인 데이터베이스
네트워크 라우팅 모듈

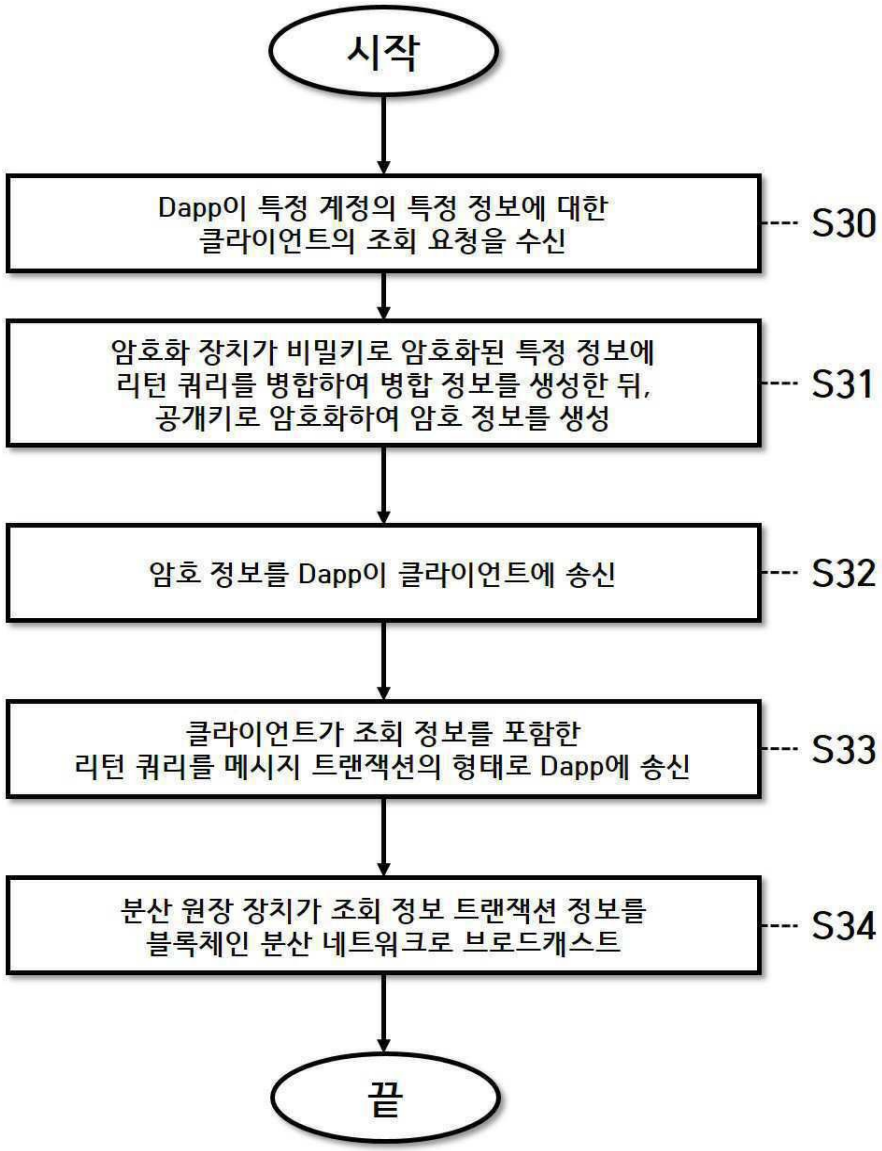
도면9


200
분산 원장 장치(1)
DB (430)
Dapp (400)
IPFS (410)
서버 (420)
블록체인(30)
블록
블록
블록
블록

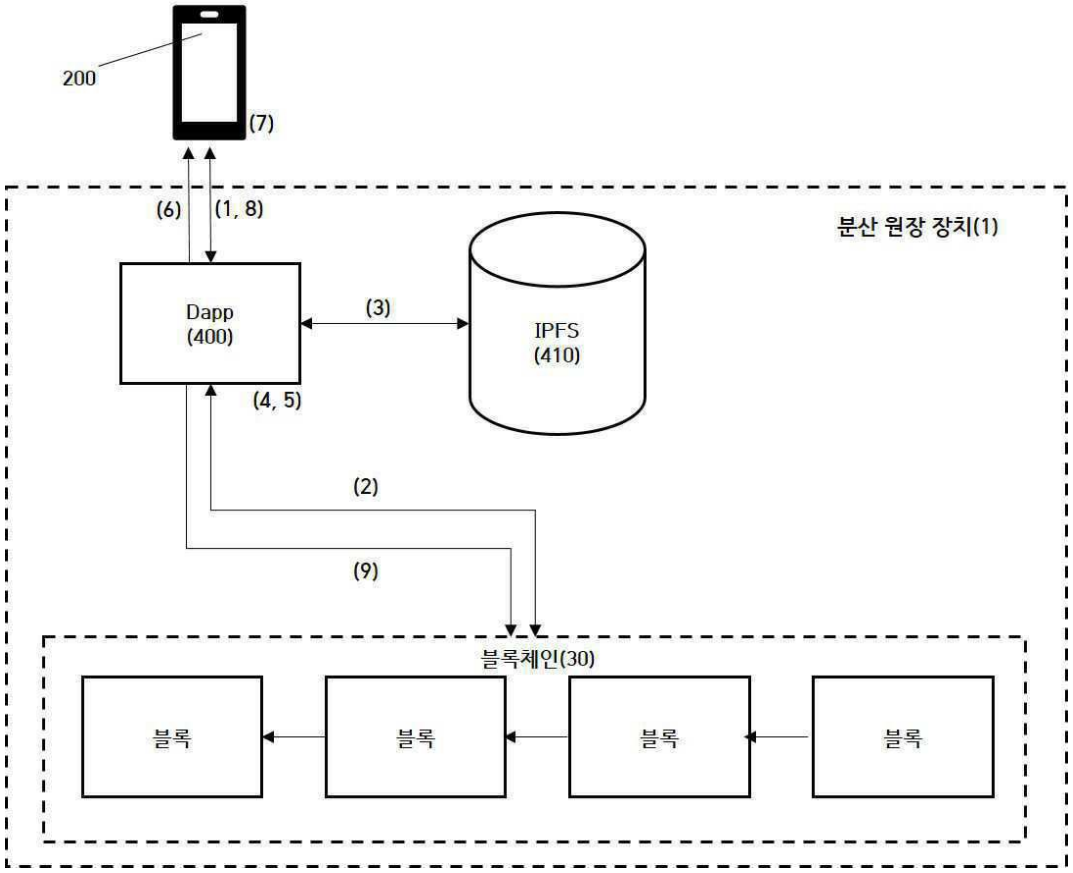
도면10


블록체인(30)
특정 계정
(20)
특정 정보 조회 요청
클라이언트
(200)
특정
정보
+ 리턴 쿼리
공개키
암호
정보
암호 정보 생성
암호 정보 송신
암호
정보
개인키
특정
정보
+ 리턴 쿼리
암호 정보 복호화
조회 정보 포함한 리턴 쿼리 송신
조회 정보 트랜잭션 정보
생성 및 기록

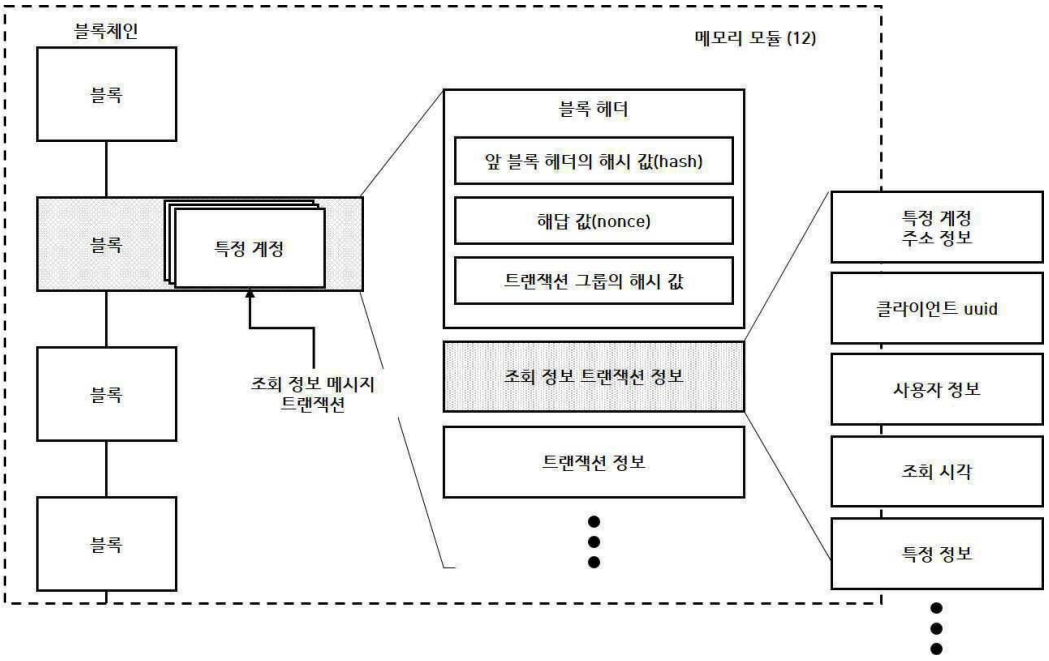
도면11


시작
Dapp이 특정 계정의 특정 정보에 대한
클라이언트의 조회 요청을 수신
S30
암호화 장치가 비밀키로 암호화된 특정 정보에
리턴 쿼리를 병합하여 병합 정보를 생성한 뒤,
공개키로 암호화하여 암호 정보를 생성
S31
암호 정보를 Dapp이 클라이언트에 송신
S32
클라이언트가 조회 정보를 포함한
리턴 쿼리를 메시지 트랜잭션의 형태로 Dapp에 송신
S33
분산 원장 장치가 조회 정보 트랜잭션 정보를
블록체인 분산 네트워크로 브로드캐스트
S34
끝

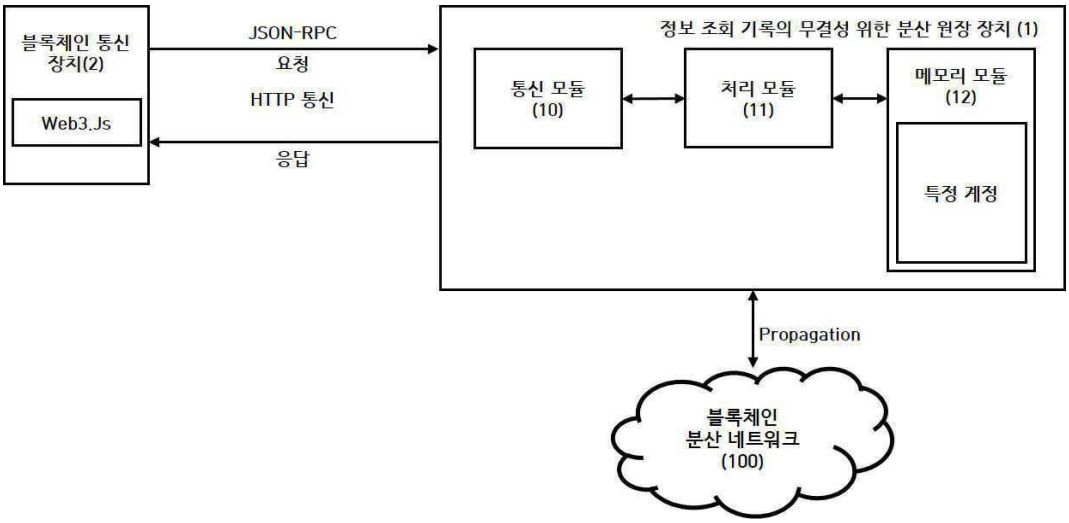
도면12


200
(7)
(6)
(1, 8)
분산 원장 장치(1)
Dapp
(400)
(3)
IPFS
(410)
(4, 5)
(2)
(9)
블록체인(30)
블록
블록
블록
블록

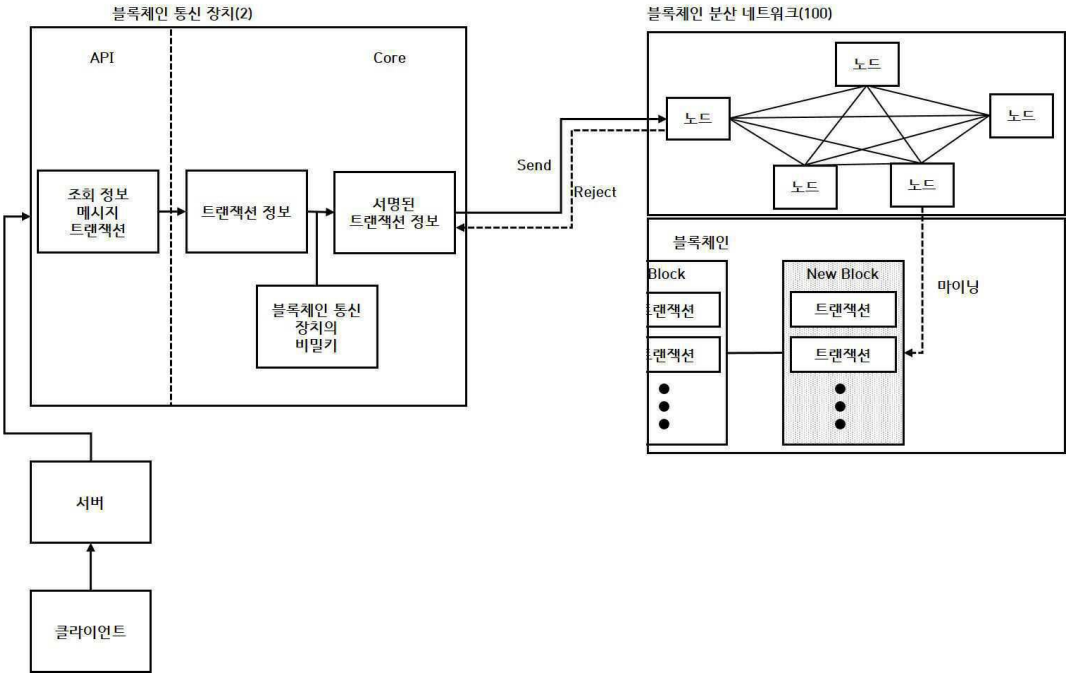
도면13


블록체인
블록
블록
특정 계정
블록
블록
조회 정보 메시지
트랜잭션
메모리 모듈 (12)
블록 헤더
앞 블록 헤더의 해시 값(hash)
해답 값(nonce)
트랜잭션 그룹의 해시 값
조회 정보 트랜잭션 정보
트랜잭션 정보
특정 계정
주소 정보
클라이언트 uuid
사용자 정보
조회 시각
특정 정보

도면14


블록체인 통신 장치(2)
Web3.Js
JSON-RPC
요청
HTTP 통신
응답
정보 조회 기록의 무결성 위한 분산 원장 장치 (1)
통신 모듈 (10)
처리 모듈 (11)
메모리 모듈 (12)
특정 계정
Propagation
블록체인 분산 네트워크 (100)

도면15


블록체인 통신 장치(2)
API
Core
조회 정보 메시지 트랜잭션
트랜잭션 정보
서명된 트랜잭션 정보
블록체인 통신 장치의 비밀키
Send
Reject
서버
클라이언트
블록체인 분산 네트워크(100)
노드
노드
노드
노드
노드
블록체인
Block
New Block
트랜잭션
트랜잭션
트랜잭션
마이닝