USECHAIN

글로벌 미러 아이덴티티

블록체인 생태계

Shengli Zhang, Huining Cao, Liang Zhang, Huazheng Cheng

기술 백서

개 요

2009 년부터 현재까지 탈중앙화의 블록체인 산업과 기술은 급속도로 발전하고 있다. 비트코인과 이더리움 시스템[1]은 블록체인 산업에 "탈중앙화 화폐"와 "스마트 컨트랙트"라는 새로운 개념을 도입했다. 블록체인 기술은 우리의 사회가 자가관리의 가능성을 보이도록 하고 있다. 하지만 현재 널리 사용되고 있는 퍼블릭 블록체인은 익명화 시스템을 기반으로 하고 있어 디지털 암호화폐는 자금 세탁, 밀수, 테러자금 조달 등 위법 행위의 도구로 이용되고 있다. 또한 실행 효율이 낮고 확장성이 떨어져 대부분 애플리케이션이 디지털 화폐 거래에 한한다. 익명 블록체인의 주체는 모두 주소이고 현실 사회의 모든 애플리케이션 주체는 자연인이다. 그러므로 블록체인의 대규모 애플리케이션을 실생활에 맞추려면 주소와 자연인 사이의 미러 아이덴티티를 구축하여야 한다.

Usechain 은 미러 아이덴티티의 퍼블릭 블록체인 개발과 ID 체인을 기반으로 한 생태 시스템 구축, 그리고 블록체인 주소와 인증된 계정의 실체를 대응시켜 블록체인의 애플리케이션을 실생활에 맞추는 것을 목표로 하고 있으며 현실 사회의 신용도와 기초시설, 상업적 애플리케이션의 결합을 실현하고자 한다. 미러 아이덴티티 프로토콜을 통해 무지식 증명을 기반으로 개인 정보와 신원 확인을 분리하여 ID 체인과 익명 퍼블릭 블록체인 간의 개인 정보 보호 수준이 동일해야 한다는 전제 조건 하에서 블록체인의 주소와 자연인의 신원이 대응되도록 한다. 미러 아이덴티티의 정보를 충분히 이용하면 샤딩 기술, 합의 알고리즘, 가상 머신 등 기존 블록체인의 알고리즘을 최적화할 수 있고 인수에 따른 투표, 주소 징벌 등 기존 블록체인이 보유하지 못한 기능을 제공할 수 있어 블록체인의 처리 성능을 향상하고 블록체인이 대규모 상업적 애플리케이션에 사용될 수 있도록 보장하여 잠재되어 있는 커다란 상업적 가치를 발굴할 수 있다.

하기 내용은 Usechain 의 기술 프레임워크, 핵심기술 원리와 기술 프로토콜을 주로 다루고 있다.

디렉토리

1. 개술

미러 아이덴티티 블록체인(이하 ID 체인이라 함) 프로젝트는 신원 확인을 진행하는 퍼블릭 블록체인을 제공하여 개인 정보를 보호하는 조건 하에서 모든 정보를 공개하고 블록체인 주소의 배후에 실생활의 자연인이 대응되는 것에 초점을 맞추고 있다. 신원 인증 분야에서 신원 인증 기관의 전문성과 인터넷 환경의 탈중앙화 특성 사이의 평행점을 찾고 메인 주소와 서브 주소(主从地址), 링서명과 신원 정보 암호화 기술로 사용자의 개인 정보를 보호하며 사용자가 권한을 부여해야만 해당 자연인의 모든 주소 정보를 공개하도록 한다.

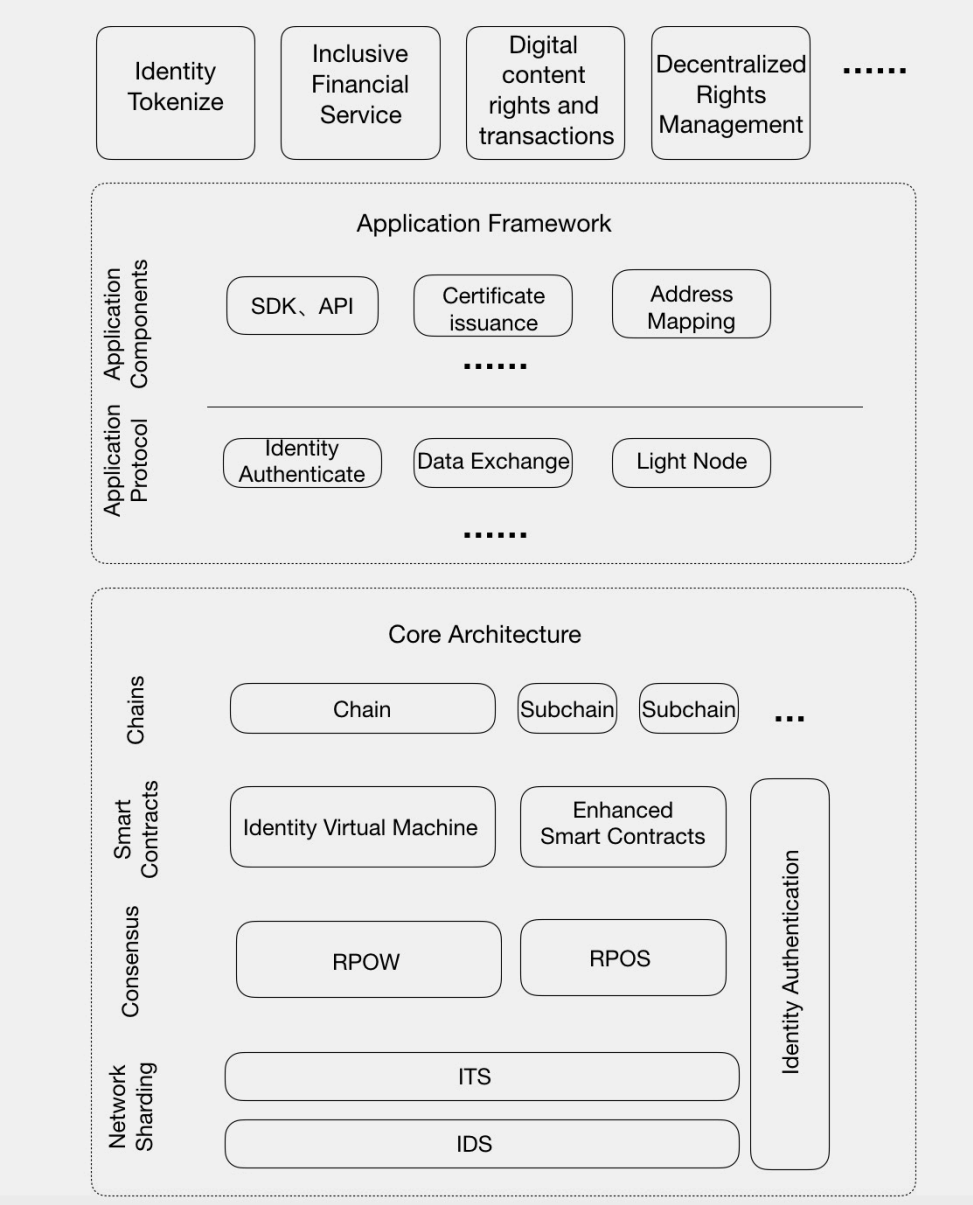


그림 1-1 Usechain 프레임워크 모형

실명제를 충분히 이용하고 개인 정보를 보호하는 유리한 조건 하에서 블록체인의 기반을 재구성하여 시스템의 성능을 대폭 향상한다. 본 프로젝트는 구체적인 상업적 환경에 입각하여 애플리케이션을 개발하고 상업 데이터 공유화를 실현하며 완전한 ID 체인 생태계를 구축할 것이다.

아키텍처 설계를 볼 때 Usechain 은 애플리케이션 서비스, 애플리케이션 프레임워크와 보톰 레이어 퍼블릭 블록체인으로 나뉜다. ID 정보를 기반으로 하는 ID 체인은 3 가지 독창적인 특성이 있다:

1) 기존 블록체인 기술을 기반으로 독창적인 미러 아이덴티티 프로토콜 기술을 개발하여 체인 주소와 인증된 사용자 사이의 다단계 대응 시스템을 구축한다. 일반 사용자를 상대로 모든 계좌 이체 주소는 인증 통과한 주소로 비교적 높은 신뢰도가 있는 동시에 사용자의 기타 개인 정보를 충분히 보호한다. 블록체인 관리위원회 노드를 상대로 모든 노드가 투표 허락한 조건 하에서 대응되는 인증된 사용자의 모든 주소를 얻을 수 있다(또한 해당 사용자의 모든 주소의 계좌 이체 내역을 획득할 수 있다). 사용자의 인증 과정은 여러 제 3 측기관이 초기화 상태에 처했을 때 이루어지며 사용자의 사회 ID 정보와 체인 주소 사이의 대응은 블록체인 결책위원회와 여러 제 3 측기관의 전원 찬성이 있어야만 완성된다(정부기관의 경로 추적에 조건을 제공).
2) 미러 아이덴티티 프로토콜을 기반으로 Usechain 은 다단계적인 기술과 설계 혁신을 가져올 것이다. 한편으로 새로운 합의 매커니즘과 샤딩 방법으로 네트워크의 확장성과 거래 속도를 높이고 거래 원가를 낮춘다. 본 프로젝트는 소프트웨어 알고리즘을 기반으로 하는 Randomized Proof of Work(RPOW)와 네트워크 샤딩(Sharding)의 혁신에 초점을 맞추고 있으며 하드웨어를 기반으로 하는 저원가 고효율의 RPOW 합의 알고리즘을 실현하는데 목표를 두고 있다. 다른 한편 신원 가상 머신(IVM-Identity Virtual Machine)을 도입한다. IVM 은 강화된 스마트 컨트랙트를 구축하는 새로운 기준이며 적은 수량의 어댑터로 정의를 할 수 있고 샌드박스화 할 수 있다. ID 체인은 중간 통신층을 구축할 수

있으며 IVM 이 중간 통신층을 기반으로 블록체인 보톰 레이어, 외부 API, 서브 체인과 데이터를 호환하고 직접 프로그래밍을 할 수 있도록 하여 이더리움 가상 머신의 부족점을 보완하고 집행 효율이 높고 외부 데이터와 호환 가능한 스마트 컨트랙트를 구축하여 스마트 컨트랙트의 애플리케이션 환경을 확대한다. IVM 은 내부에 스마트 컨트랙트의 완전성을 테스팅하는 알고리즘을 구축하여 프로그래머가 스마트 컨트랙트를 프로그래밍 하는 동시에 테스팅을 진행하여 안전한 스마트 컨트랙트 시스템을 실현한다.

3) Usechain 은 완비한 블록체인 기반의 커뮤니티 치리 시스템을 개발하였는데 사람들이 주관 문제에 관한 합의 과정을 실현할 수 있다. 또한 Usechain 은 완비한 투표 시스템을 갖추고 있어 사용자는 투표를 통해 커뮤니티의 중대결책에 참여할 수 있다. ID 체인의 주소는 신원 인증을 거친 주소여서 1 인 1 표의 공평 관리 시스템을 실현 할 수 있다. 계약에 오류가 발생하거나 해커가 코인을 훔치는 부정행위가 발생할 시 커뮤니티는 투표를 통해 해커를 추적할지 여부를 결정할 수 있다. 이 밖에 Usechain 은 일련의 징벌 조치를 규범화 하였는데 필요시에 법률집행기관과 함께 악행을 저지른 자의 법적 책임을 묻고 부정행위를 단속하고 처리할 수 있다.

ID 체인의 핵심은 신원인데 중앙화+탈중앙화를 결합한 인증 방법으로 신원 인증을 할 수 있다. 신원이 생기면 신원+데이터->신용이 된다. 마이신용(蚂蚁信用) 등 기존 신용 플랫폼과 다른점은 데이터에 한계가 없는 것이다. 블록체인 내부에 응용되는 모든 데이터는 퍼블릭 블록체인에서 공유화 할 수 있으며 완비한 사용자 페르소나의 방대한 데이터를 기반으로 하는 플랫폼을 갖출 수 있다. 퍼블릭 블록체인의 통용 화폐는 단일 Token 이며 Token 은 부동한 애플리케이션 사이에서 유통하고 유통은 데이터를 만들며 데이터+신원은 신용을 만든다. 퍼블릭 체인은 더욱 완비한 합의 매커니즘으로 소비를 활성화 하고 거래를 확인하는 것과 조작(操作)된 스마트 컨트랙트로 도구를 만들고 오프체인에서부터 온체인까지의 데이터 인터페이스를 필요로 한다.

ID 체인을 기반으로 한 더욱 철저한 DApp 을 서브 체인이라 한다. 모든 서브 체인은 완전한 ID 체인 프로토콜이 있지만 퍼블릭 블록체인과

밀접한 연관이 있으며 데이터를 호환할 수 있다. 학위를 기반으로 ID 코인을 생성하는 것은 그 중 한가지 DApp 에 속하며 사용자는 학위 인증+퍼블릭 블록체인 Token 을 통해 ID 코인을 만들 수 있다. ID 코인은 한정량으로 소장가치가 있고 판매자 결제, 판매자 포인트 코인화, 탈중앙화 무국경 P2P 등도 모두 ID 체인을 기반으로 하는 서브 체인 애플리케이션으로 될 수 있다. 또한 온라인 폴트 톨러런스 매커니즘을 도입하여 거액의 자금이 도난당할 시에 투표로 해당 계정을 동결하고 법률 기관에 신고하여 결재할 수 있다. 그리고 사법 기관의 결재에 따라 해당 계정의 정보를 공개하고 계정에 대응되는 자연인을 징벌한다.

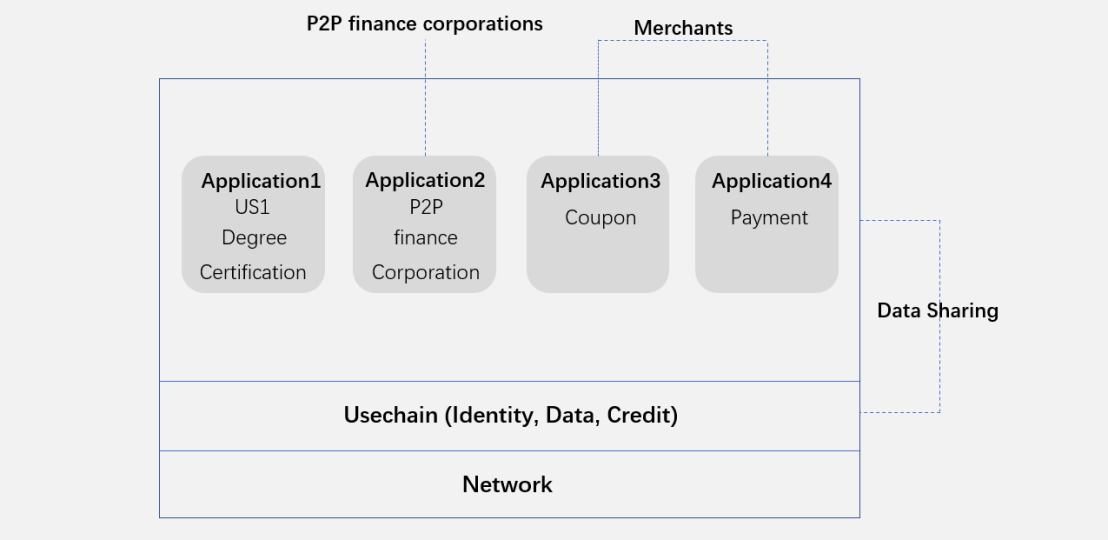


그림 1-2 신원 인증 애플리케이션 개발

ID 체인은 사용자의 실체를 기반으로 하는 신형의 퍼블릭 블록체인이며 완비한 개인 정보 보호와 신원 인증 매커니즘을 제공한다. 또한 신원에 따른 특수한 저원가 고효율의 합의 알고리즘을 제공할 수 있고 스마트 컨트랙트의 환경을 확대할 수 있다. ID 체인을 기반으로 부동한 신원의 해당 애플리케이션은 빠른 속도로 탈중앙화, 자원공유화와 자체개발의 생태 시스템을 구축할 수 있다.

2. 기술 프레임워크

2.1 생태 아키텍처

ID 체인의 생태 아키텍처는 크게 보톰 레이어 블록체인과 상층 생태 애플리케이션 이 두 가지 부분으로 나뉜다. 다른 블록체인 애플리케이션이 익명성을 강조하는데 반해 ID 체인의 주소는 인증을 거쳐야 하며 해당 계정은 유일한 실체와 대응된다[2]. 계정 실명제로 인해 ID 체인은 소셜 미디어, 개인 신용도, 상업적 홍보 등 분야에 큰 영향을 미칠 것이다.

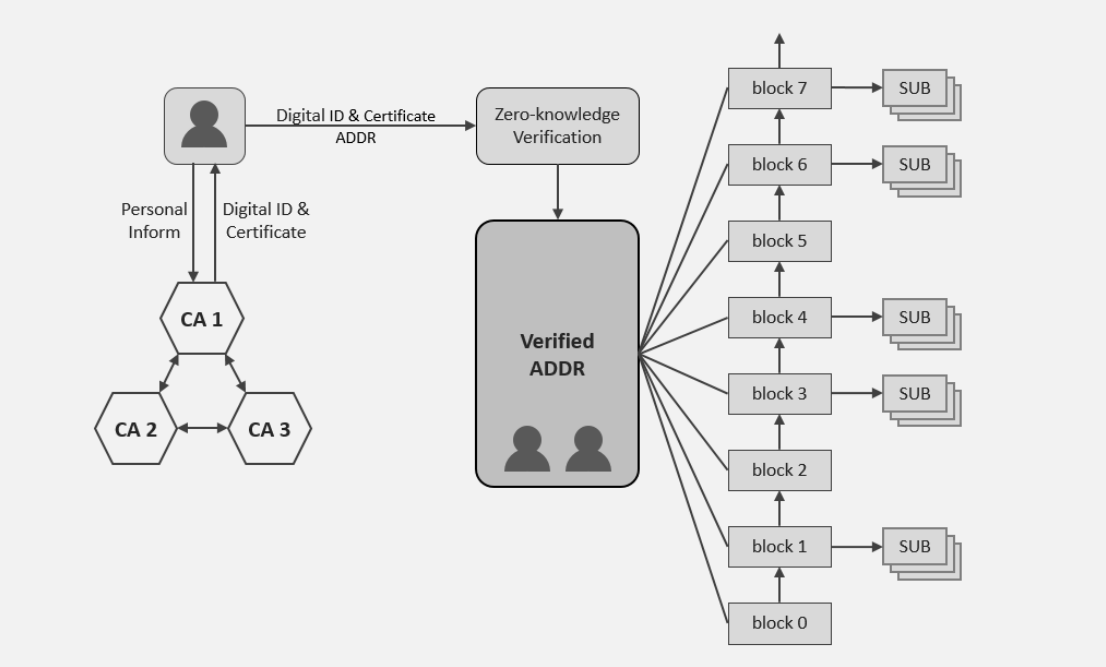


그림 2-1 신원 인증 시스템 아키텍처

보톰 레이어 블록체인(bottom-layer blockchain)의 설계는 블록체인 기술이 현존하는 채굴자원 낭비, 거래 확인 속도 느림, 인터넷 거래의 부하량 과소, 데이터 동기화 과대 등 일련의 문제들에 초점을 맞추고 있다. ID 체인은 합의 매커니즘, 네트워크 샤딩, 블록체인 가상 머신과 라이트 클라이언트 네가지 부분에서 블록체인 기술에 관해 신원 특성을 기반으로

하는 수정을 진행할 것이다. 합의 매커니즘을 선택할 시에 POW[3]와 POS 를 기반으로 개선할 것이며 RPOW(Randomized Proof of Work)와 RPOS(Randomized Proof of Stake)을 제출하여 랜덤으로 다음 블록체인 패키저(blockchain packager)를 생성한다. 이것은 컴퓨팅(computing) 경쟁을 피하고 탈중앙화를 보장하는 것을 기반으로 블록의 생성 속도를 대폭 높이는 동시에 하드웨어 물리적 증명 합의 알고리즘(the consensus algorithm of physical proof of hardware)을 사용해 합의 알고리즘의 현존 문제점을 근본적으로 해결할 수 있다.

네트워크 블록현상을 해결하고 대규모의 거래 수량을 지원하기 위해 Usechain 체인 네트워크 시스템은 서브 체인을 도입하여 부동한 애플리케이션을 처리한다. 서브 체인은 사용자의 수요에 따라 메인 체인에서 파생된 블록체인이며 애플리케이션 니즈에 따라 합의 방식, 블록 크기, 블록 제공 시간과 기타 기능 모듈을 결정할 수 있다. Usechain 메인 체인의 신원 인증 시스템과 Token 시스템을 기반으로 서브 체인의 응용은 빠른 시일내에 실현할 수 있다.

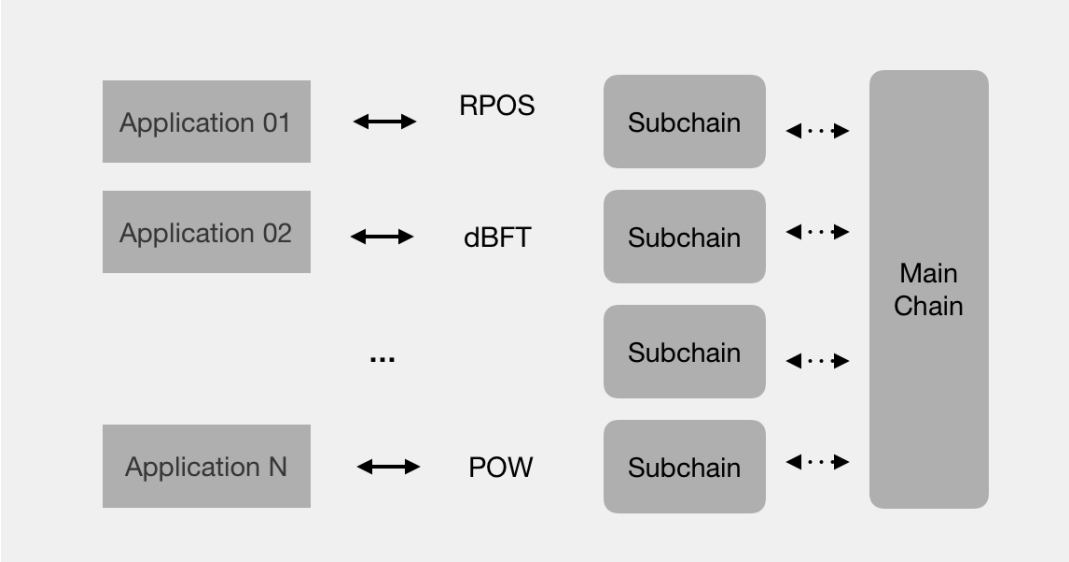


그림 2-2 서브 체인 애플리케이션

ID 체인의 높은 신뢰도를 기반으로 샤딩 기술[4]을 도입하여 모든 ID 체인 네트워크를 사용자의 계정 주소에 따라 네트워크 샤딩을 진행하고 한 개의 주소가 진행하는 거래는 반드시 소속되어 있는 샤딩 내부의 모든

노드의 확인을 거쳐야 하지만 전반 네트워크 노드가 확인할 필요가 없도록 한다. 이것은 이중 사용 문제(Double-spending)를 방비하고 거래 확인 속도를 높이며 네트워크 거래의 부하량을 높일 수 있다.

새로운 합의 알고리즘과 샤딩 기술을 기반으로 블록체인 가상 머신의 환경을 개선하여 가상 머신의 컴퓨팅 성능을 향상하고 컴퓨팅 비용을 줄이거나 없애며 ROM 모듈을 최적화하고 더 많은 종류의 언어 지원과 계약 완전성 테스팅 지원, 서브 체인의 크로스 체인 지원과 외부 API 인터페이스를 추가 한다.

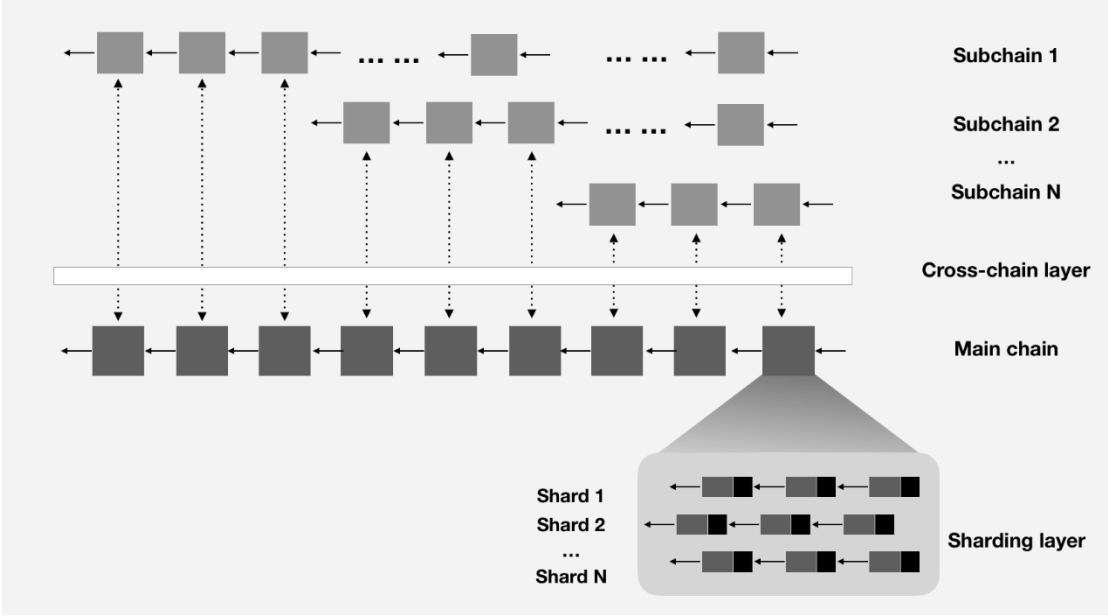


그림 2-3 서브 체인, 메인 체인과 샤딩

또한 동기화 데이터 과대의 문제를 피하기 위해 Usechain 라이트 클라이언트를 개발하였다. 이것은 머클 증명(Merkle proof)의 이론을 기반으로 하여 라이트 클라이언트의 동기화 데이터 수량을 줄이고 트랜잭션과 데이터의 정확성을 검증할 수 있도록 한다. ID 체인 네트워크 중 모든 노드는 완전한 노드가 아니어도 되며 합의 과정에 참여하지 않은 노드는 라이트 노드로 변할 수 있다. 그리고 Usechain 보톰 레이어 아이덴티티 퍼블릭 체인의 설계는 공개화 되어 있는데 이것은 다양한 기능 컴포넌트, 강화된 스마트 컨트랙트와 가상 머신 시스템을 제공할 수 있다. 개발자는 Usechain 의 보톰 레이어 퍼블릭 체인을 기반으로 부동한

DApp 을 빠른 속도로 개발할 수 있다.

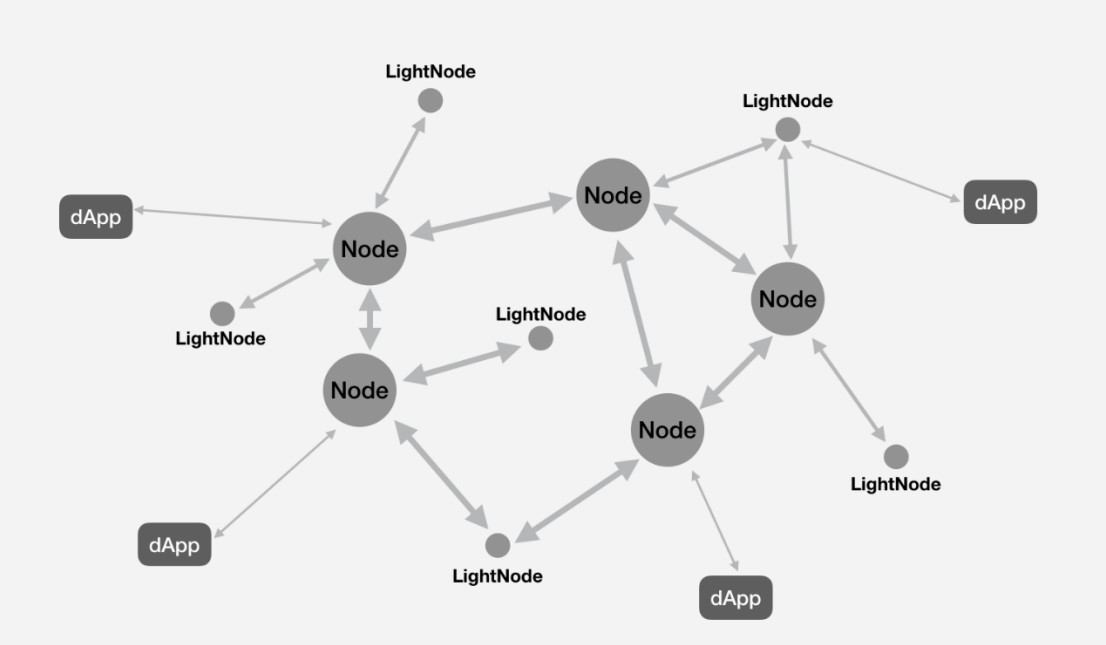


그림 2-4 Usechain 네트워크 노드

ID 체인은 "탈중앙화 애플리케이션 서브 체인"과 "스마트 컨트랙트"를 통해 서브 체인, 퍼블릭 체인, 오프 체인 요소를 도입하여 현실 세계의 비즈니스 법칙에 맞는 블록 체인 스마트 컨트랙트를 형성하며 여러 업종, 여러 경로를 지원할 수 있다. 또한 기술 프레임워크에서 볼 때 모바일 서비스를 지원할 수 있는데 그 중에는 모바일 월릿, 모바일 DApp 애플리케이션, 모바일 스마트 컨트랙트 서비스를 포함한다. ID 체인의 생태 시스템 환경에서 제 3 측 개발자(혹은 제 3 측 사용자)는 자신의 서브 체인과 기타 서브 체인을 손쉽게 연결할 수 있다. 그리고 블록 체인의 모바일 서비스를 제공할 수 있어 ID 체인의 부동한 애플리케이션 환경에서의 실현 과정을 추진할 수 있다.

2.2 소프트웨어 프로토콜 스택

ID 체인의 설계는 거래 부하량을 높여 천만급 사용자가 동시에 사용할 수 있는 것, 블록 체인의 블록 생산 속도를 높이고 거래 확인 효율을 높이는 것, 계약 배치 비용을 줄이는 것, 가상 머신을 최적화 하고 개발과 테스트 과정을 간편화 하는 것을 목표로 하고 있다. 또한 보톰 레이어의

거래 오더, 블록, 합의 가치(consensus value), 라이트 노드 인증, P2P 네트워크[5], 블록 체인 가상 머신에서부터 탑 레이어의 블록체인 애플리케이션까지 실현할 수 있으며 완전한 ID 체인 생태계를 구축할 수 있다.

설계 측면에서 볼 때 네트워크 레이어, 데이터 레이어, 합의 레이어, 격려 레이어와 스마트 컨트랙트 레이어로 나뉜다. 각 레이어마다 일정한 애플리케이션을 위해 서비스를 제공할 수 있고 부동한 애플리케이션의 수요를 만족할 수 있으며 개인 혹은 기업이 빠르고 안전하게 여러 애플리케이션 환경과 비즈니스 모드를 실현하는 것을 보장할 수 있다.

네트워크 레이어: 네트워크 레이어는 네트워크 부하량을 강화하고 네트워크 거래의 처리 속도를 높이기 위해 KaZaA 프로토콜과 P2P 프로토콜을 도입하였다. 또한 KaZaA 프로토콜의 슈퍼 노드 구조와 매치하여 INS(Identity Network Sharding, ID 체인 네트워크 샤딩) 전략과 ITS(Identity Transaction Sharding, ID 체인 트랜잭션 샤딩) 전략을 동시에 설계하였는데 INS 와 ITS 전략은 Generating shards, Directory Service Committee, Resolving Conflicts, Transaction Assignment and Processing 네가지 주요 부분으로 구성된다.

데이터 레이어: 블록의 체인식 구조를 기반으로 한 개 블록의 유효 여부는 그 전 블록이 존재하는지, 그리고 유효한지, 블록의 타임 스탬프가 유효한지, 블록의 작업증명이 유효한지, 내부 거래가 유효한지 등을 통해 확인할 수 있다. ID 체인은 이런 요소를 기반으로 각 블록마다 유효한 주소 데이터를 추가하고 머클 트리로 저장해야 한다. 모든 거래는 모두 주소의 유효성을 검증하여야 한다.

기존의 블록 인증은 블록체인의 전반 데이터를 유지하는 것을 통해 새로운 블록을 더하거나 트랜잭션을 확인한다. 하지만 이런 방법은 방대한 저장공간이 있어야만 블록체인 노드를 정상적으로 실행할 수 있다. 그에

반해 ID 체인은 라이트 클라이언트를 개발하여 머클 트리 데이터 구조의 설계를 최적화 하고 라이트 노드가 필요로 하는 데이터 공간을 대폭 줄이며 모바일 장치와 사물인터넷 장치가 블록체인에 연결될 수 있도록 가능성을 제공한다.

합의 레이어: 합의 매커니즘의 설계로부터 볼 때 새로운 합의 매커니즘인 RPOW(Randomized Proof of Work, 랜덤작업증명)와 RPOS(Randomized Proof of Stake, 랜덤지분증명)를 설계하였다. RPOW 는 작업증명 알고리즘과 ID 체인 특유의 사용자 신원 인식 매커니즘을 기반으로 랜덤 배치 알고리즘을 설계하였다. 이것은 각 블럭마다 모든 마이너에게 부동한 채굴 난이도를 선보일 것이며 네트워크의 가상 산력(算力) 배치를 조절할 것이다. 이러한 알고리즘은 한편으로 기존 POW 알고리즘의 효율을 대폭 높일 수 있고 자원 낭비를 줄일 수 있으며 다른 한편으로 마이닝 풀의 산력이 한곳으로 집중되는 추세를 늦출 수 있고 네트워크의 공정성과 보안성을 더 잘 실현할 수 있다[7]. RPOS 는 주로 DPOS(Delegated Proof of Stack, 분산지분증명) 합의 알고리즘을 기반으로 하고 있으며 각 주기내에서 위원회(committee)의 생성방식을 최적화할 수 있고 블록생산자를 선택할 시에 DPOS 의 순서별 생산 방식을 포기하고 RAS(Random Appoint Strategy, 랜덤위임전략)를 설계하여 위원회 내부의 생산권의 전달을 실현한다.

ID 체인의 보안성과 실행 효율을 높이기 위해 하드웨어 증명(Proof of Hardware) 합의 알고리즘을 제출하였다. 컴퓨팅 하드웨어(CPU 등)의 집적회로급 보안 조치를 이용하여 위조 방지와 신뢰도를 보장하는 합의 알고리즘을 실현하고 모든 노드 범위에서 랜덤으로 블록 생산 노드를 선택한다.

격려 레이어: ID 체인의 퍼블릭 블록체인 token 은 개인 신원과 긴밀히 연관되어 설계하였으며 ID 체인 기술의 발전과 함께 미래의 애플리케이션도 확장될 것으로 보이고 있고 token 의 가치도 계속 높아질 것이다. 네트워크 노드를 유지하는 마이너 노드를 상대로 일정 수량의

token 을 보상할 것이며 이를 장부 기록의 보답으로 간주한다. 비트코인의 반으로 줄어든 주기적 보상이나 이더리움의 고정된 보상 방식에 반해 ID 체인은 마이너의 수익과 네트워크의 거래 비용 수준을 감안할 것이며 이에 기초하여 독특한 보상 방식을 설계할 것이다. 또한 ID 체인은 각 노드, 각 서브 체인에서 저마다 기명(记名) token 을 발행하는 것을 지원할 것이며 부동한 token 사이의 분산화 거래를 지지하여 해당 가치를 실현할 것이다.

스마트 컨트랙트 레이어: ID 체인의 전용 가상 머신 IVM(Identity Virtual Machine, ID 체인 가상 머신)을 설계한다. 이것은 컴퓨팅 성능 개선, 계약 개발 원가 감소, ROM 배치 시스템 최적화, 개발 규범화 등을 설계 목표로 하고 있다. 또한 스마트 컨트랙트의 실행 매커니즘을 개선하고 스마트 컨트랙트의 내부 완전성 테스트 공구를 제공하여 기업 사용자가 스마트 컨트랙트를 이용할 수 있도록 한다.

애플리케이션 레이어와 온라인 폴트 톨러런스: ID 체인은 신원을 기반으로 하는 다양한 애플리케이션을 지원할 수 있으며 이것은 향후 신원 애플리케이션을 구축하는 기초 작업으로 될 것이다. 이를 기반으로 연구 개발팀에서는 온라인 폴트 톨러런스라 불리는 특별한 애플리케이션을 개발할 것이며 본 애플리케이션은 ID 체인에 충돌 혹은 포크 현상이 발생할 시에 자동으로 문제를 일으킨 계정과 해당 블록을 동결함과 동시에 기타 노드의 정상적인 거래를 보장한다. 또한 특별 위원회에서 문제를 일으킨 블록과 계정을 결재하고 서명 확인을 진행한다.

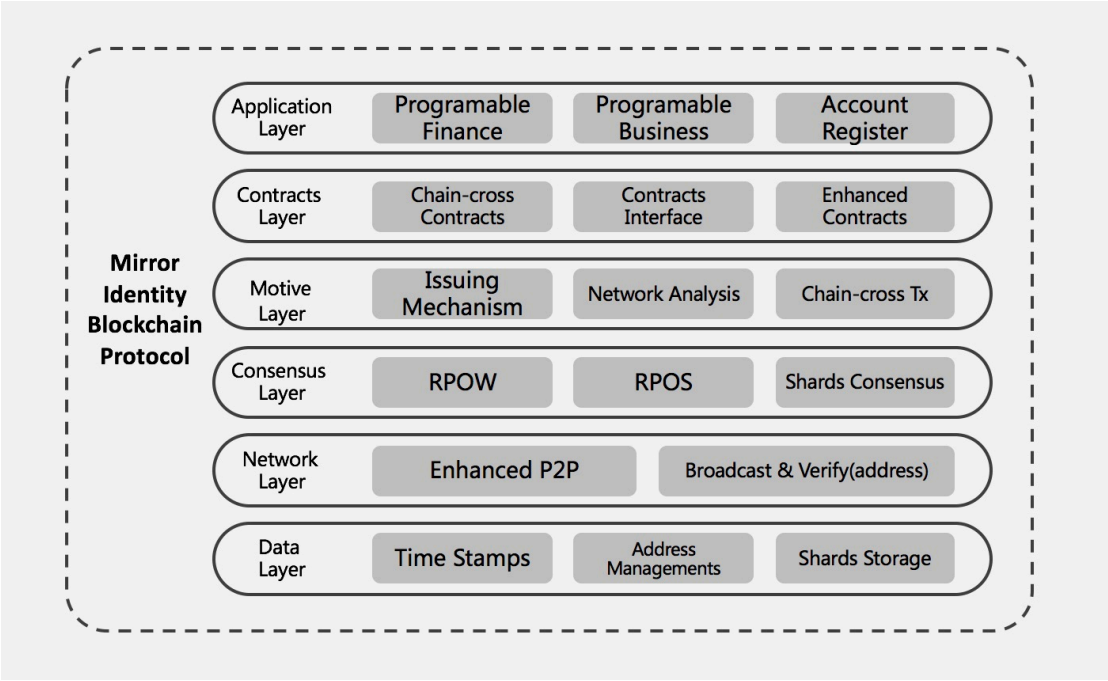


그림 2-5 신원 매핑 블록체인 프로토콜

3. 신원 인증

3.1 메인 주소와 서브 주소

메인 주소는 사용자의 메인 계정이며 사용자가 클라이언트를 등록하여 지갑을 만들 때 생성된다. 사용자는 하나의 메인 주소를 소유할 수 있으며 주소와 신원은 1 대 1 로 매치된다. 메인 주소는 타원곡선 Secp256k1 으로 퍼블릭 키와 프라이빗 키를 생성하고 해당 퍼블릭 키로 주소를 생성한다. 메인 주소와 서브 주소는 검증을 거쳐야만 거래를 진행할 수 있다. 서브 주소의 수량은 무제한이며 사용자의 수요에 따라 생성될 수 있지만 사용자의 메인 계정 퍼블릭 키를 필요로 한다. 서브 주소와 메인 주소는 별도로 저장되며 둘 사이의 의존 관계를 볼 수 없다.

3.2 서브 주소 생성 알고리즘

서브 주소는 타원곡선 Elliptic-curve Diffie – Hellman (ECDH)[8, 17] 알고리즘에 따라 생성된다. 만약 사용자의 퍼블릭/프라이빗 키가 (A, a)이면 랜덤으로 Keypair(S, s)가 생성되며 위원회의 퍼블릭 키$(B_1, B_2, \ldots, B_n)$ 로 서브

주소의 퍼블릭/프라이빗 키를 생성한다.

퍼블릭 키 생성 방법:

$$A_1 = Hash([s]B_1 + A)G + Hash([s]B_2 + A)G + \cdots + Hash([s]B_n + A)G + S$$

프라이빗 키 생성 방법:

$$a_1 = Hash([s]B_1 + A) + Hash([s]B_2 + A) + \cdots + Hash([s]B_n + A) + s$$

서브 계정:(A_1, S)과 대응되는 프라이빗 키는 Keystore 에 저장된다.

3.3 신원 검증 과정

애플리케이션의 탑 레이어는 ID 체인 주소의 실명 검증을 필요로 하며 ID 체인은 주로 여러 개의 공신력 있는 제 3 측 기관을 통해 각 메인 주소의 검증과 구축 과정을 실현한다. 구체적인 과정은 하기 참조:

1) 사용자는 클라이언트를 통해 제 3 측 기관에 CA 증서를 신청하고 제 3 측 기관에서 인증을 거친 후 사용자에게 인증 증서를 발급한다. 본 과정은 한 번만 진행된다.

2) 사용자는 무지식 증명 방법을 통해 블록체인에 인증을 받았음을 증명할 수 있고(CA 정보를 노출할 필요 없음) 사용자에게 생성된 주소는 블록체인의 스마트 컨트랙트에 기록되어 해당 사용자의 유일한 메인 주소로 된다. 온체인에는 메인 계정에 관한 신원 정보를 찾아볼 수 없을 것이며 이러한 방법으로 개인 정보의 비밀성을 확보한다.

3) 사용자는 자신의 퍼블릭 키와 프라이빗 키를 이용하여 랜덤으로 블록체인에 있는 퍼블릭 키를 선택할 수 있고 링서명 전략으로 위원회에 서브 주소의 인증을 신청할 수 있다.

4) 사용자가 거래를 진행할 시에 거래 유효성 검증을 진행해야 하는데 Usechain 은 CA 계약으로 주소를 검증하고 거래의 합법 여부를 판단하며 합법 거래일 경우 거래를 집행한다.

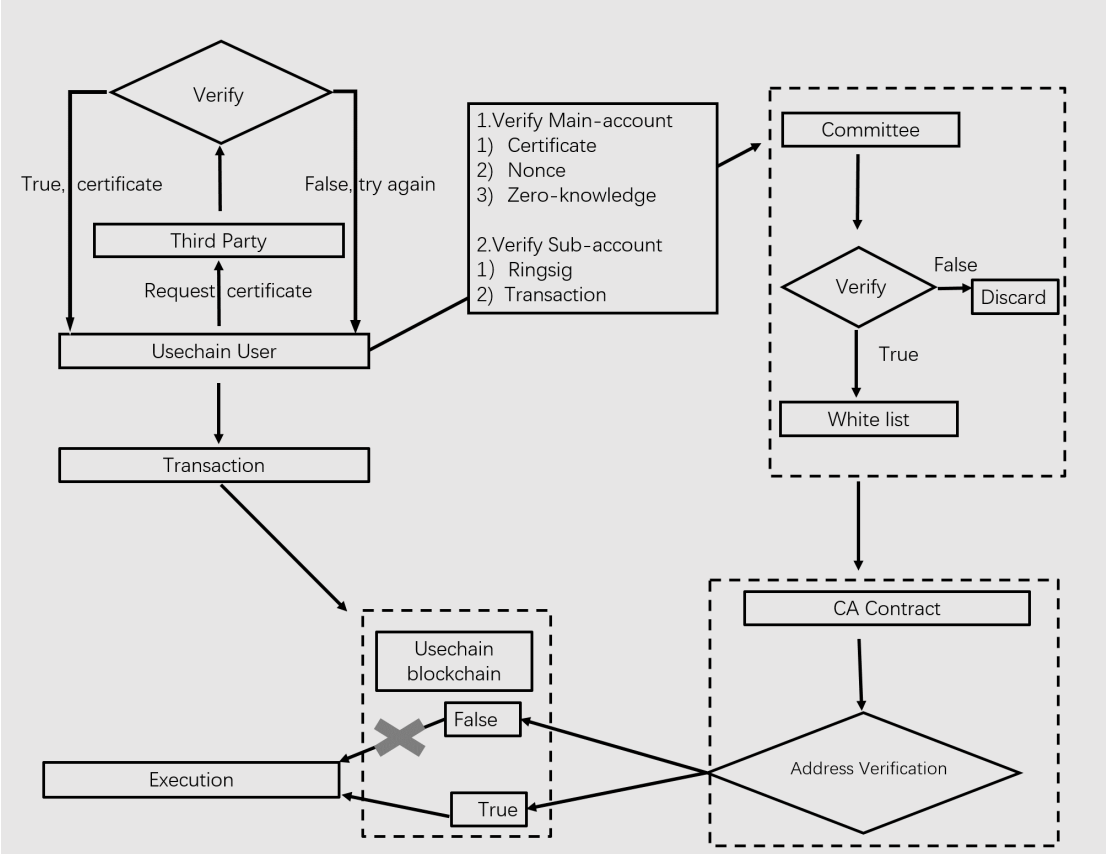


그림 3-1 ID 체인 인증 과정과 거래

5) 위원회는 계정이 위법 거래 가능성을 보이는 등 필요한 상황에서 네트워크를 스캔하여 계정 관련성을 검색한다. 위원회의 퍼블릭/프라이빗 키는 (B_1, b_1), (B_2, b_2), … , (B_n, b_n) 이며 타원곡선인 디피-헬만 키 교환 알고리즘에 따라 하기 결과를 얻을 수 있다:

$$\begin{cases} B = [b]G; S = [s]G \\ [s]B = [s][b]G = [b]S \end{cases}$$

위원회 계산:

$$A_{11} = Hash([b_1]S + A)G$$
$$A_{12} = Hash([b_2]S + A)G$$
$$\cdots\cdots$$
$$A_{1n} = Hash([b_n]S + A)G$$
$$A'_1 = A_{11} + A_{12} + \cdots + A_{1n} + S$$

$A_1 = A'_1$일 때 계정 관련성을 검색할 수 있다.

4. 합의 알고리즘

분산화 네트워크의 가장 핵심은 분산화의 일치성을 보장하는 것(전체 노드가 동일한 제안 혹은 데이터에 관해 합의를 달성하는 것)인데 합의 알고리즘이 분산화 시스템의 일치성을 보장할 수 있다. 분산화 네트워크의 수많은 노드는 불가피한 통신 지연 현상으로 인해 서버가 다운 되거나 고장 나거나 효력을 잃는 등 복잡한 상황이 발생할 수 있다. 또한 블록체인 네트워크는 상기 요소를 감안하는 동시에 일정량의 악성 노드 공격을 방어해야 하며 최대한도로 탈중앙화를 실현해야 한다. 그러므로 합의 알고리즘은 블록체인 시스템 중 가장 관건적인 일환으로 부단히 연구하고 최적화 할 의미가 있다. 하지만 분산화 네트워크에는 완벽한 합의 알고리즘이 없다는 점을 알아두어야 하는데 극히 높은 거래 부하량과 빠른 거래 확인 속도, 완벽한 탈중앙화는 동시 실현이 불가능한 상황이다. Usechain 네트워크는 단일한 한 갈래의 메인 체인으로 모든 애플리케이션을 실행하지 않을 것이며 네트워크 샤딩, 사이드 체인 등 기술로 메인 체인과 서브 체인을 나누어 부동한 애플리케이션을 처리하도록 한다. 또한 서브 체인은 구체적인 애플리케이션의 환경 수요에 따라 가장 적합한 합의 알고리즘을 선택한다.

4.1 RPOW

4.1.1 RPOW

비트코인 시스템은 작업 증명 매커니즘(Proof of Work)을 실행하고 있다. POW 는 전체 시스템 중 산력 경쟁 매커니즘을 통해 컴퓨팅을 먼저 완성한 노드가 기록 작업을 진행하도록 한다. POW 는 현존 합의 알고리즘 중 유일하게 대량 사용자의 장기간 검증을 받은 합의 알고리즘이다.

사토시 나카모토가 POW를 설계한 초심은 모든 비트코인 노드가 전체 시스템의 결책 매커니즘에 참여할 수 있도록 하는 것이었지만 GPU 채굴부터 FPGA 까지, 나아가 ASIC 채굴까지를 볼 때 산력을 집중한

마이닝 풀은 이미 최대한의 민주와 탈중앙화를 완전히 벗어나고 있다. 많은 마이너들은 비트코인의 생태 시스템을 잘 알지도 못하는 상황에서 비트코인의 발전추세를 컨트롤 하고 있다. 이 밖에도 작업 증명은 대량의 전력을 소모하고 있지만 이런 전력 소모는 전력을 낭비하는 것이나 다름없으며 그 어떠한 사회적 산품도 창조할 수 없다.

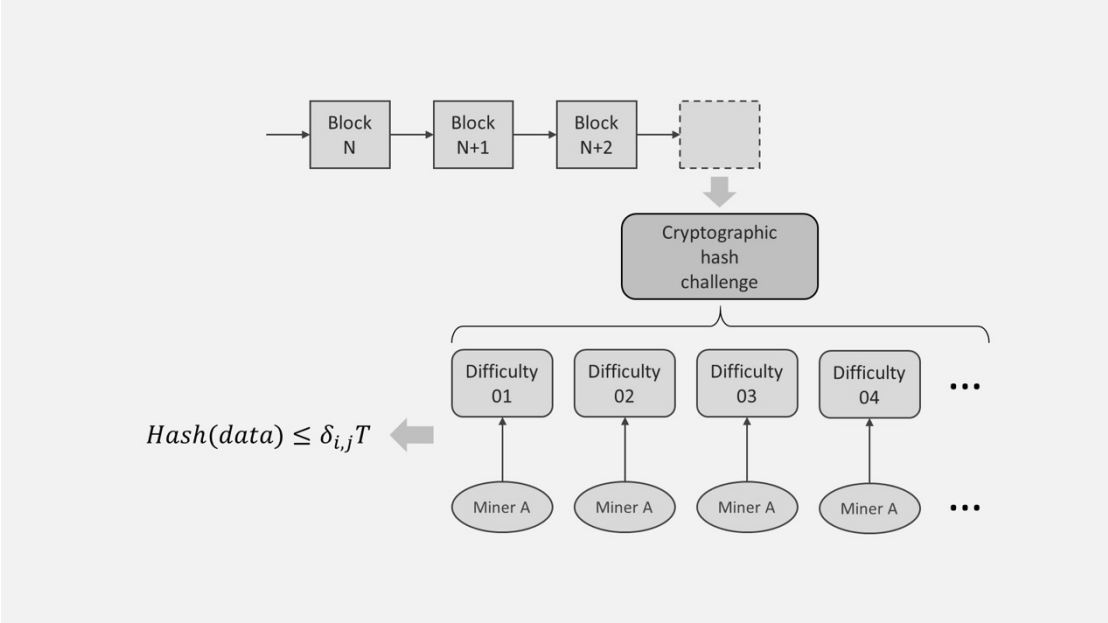


그림 4-1 RPOW 합의 알고리즘

POW 의 양호한 특성을 살리고 새로 발생하는 여러가지 문제점을 줄이기 위해 본 프로젝트는 RPOW 알고리즘을 제출한다. 마이너 i 는 제 j 번째 블록을 포장할 시에 랜덤 난이도 인하 요소 $0 < \delta_{i,j} < 1$ 를 얻을 수 있으며 마이너가 계산해야 하는 해시값은 하기 조건을 만족:

$$Hash(data) \leq \delta_{i,j}T$$

그중 T 는 전체 체인의 기초 난이도값이다. 상기 조건에서 네트워크 중 각 노드의 난이도는 전부 부동하며 각 블록마다 랜덤으로 변할 수 있다. $\delta_{i,j}$ 의 랜덤성, 식별성과 불변성을 실현하기 위해 $\delta_{i,j}$ 는 채굴 당시의 메인 주소, 해당 블록체인 데이터와 사전 저장 정보에 따라 계산해 낼 수 있다. 마이너의 구체적인 채굴 과정은 하기 참조:

1) 마이너 A 는 전량(全量) 블록 장부 데이터를 동기화 하고 채굴 계정 정보를 도입한다. 채굴 계정의 주소는 T 이다;

2) 새로운 블록을 접수한 후 블록의 유효여부를 검증한다;

3) txpool 의 거래 데이터를 포장하고 포장 완료한 거래의 유효여부를 검증한다;

4) A 는 자신이 처한 채굴 난이도를 계산하며 기초 난이도는 지난번 블록 생산 시간과 블록 생산 수량에 따라 조정된다. 다른 한편 마이너는 이전 블록의 블록 데이터와 채굴 주소에 전자서명을 진행하며 서명 문자열 S 를 얻는다;

5) 서명 S 와 주소 T 의 매치 정도를 계산하고 매치 정도에 따라 난이도를 조정한다. 일부 노드는 난이도가 대폭 낮추어질 것이며 블록 생산 확율을 높일 수 있다;

6) 블록 데이터를 조정(랜덤숫자 nonce, 타임 스탬프, 거래 등)하고 해시 계산을 진행하여 블록이 난이도 요구를 만족할 수 있도록 한다;

7) 새로운 블록을 브로드캐스팅 한다. 기타 노드는 블록을 접수한 후 난이도의 유효성을 포함한 블록의 유효여부를 검증한다.

RPOW 알고리즘은 마이닝 풀의 산력 집중 정도를 낮추고 전력 낭비를 줄일 수 있으며 전체 시스템의 블록 생산 속도를 백 배 이상 높일 수 있다.

4.1.2 하드웨어를 기반으로 하는 RPOW

본 프로젝트는 소프트웨어 알고리즘을 기반으로 하는 RPOW 를 과도기 합의 알고리즘으로 한다. ID 체인은 궁극적으로 하드웨어를 기반으로 하는 RPOW 를 실현할 것이며 전용 하드웨어 장치로 합의 매커니즘과 신원 인증 매커니즘을 생성한다. 이런 방법은 에너지 소모를 필요로 하지 않는 조건 하에서 전체 블록체인 시스템의 안전성과 효율성을 보장한다.

4.2 RPOS

4.2.1 DPOS 설명

분산지분증명(Delegated Proof of Stake)은 전국인민대표대회와 유사한 매커니즘이다. DPOS 의 블록체인에서 온체인의 모든 코인 소지자는 투표 방식으로 위임자를 선거할 수 있고 후보들 중 득표수가 상위 21 명인 노드는 위임자로 선출된다(현재 EOS 위임자의 수량은 항시 21 명이고 Bitshares 는 101 명이다. 이론적으로 홀수노드는 모두 가능하다). 노드 투표의 가중과 노드의 코인 소유량은 정비례를 이룬다. 선출된 위임자는 위임 주기 동안 블록을 생산할 수 있고 일반 노드는 투표를 통해 악성 노드 혹은 일탈 노드를 변경할 수 있다. DPOS 는 모든 코인 소지 노드에게 투표권을 부여하며 이런 방법으로 네트워크의 운영원가를 줄일 수 있다.

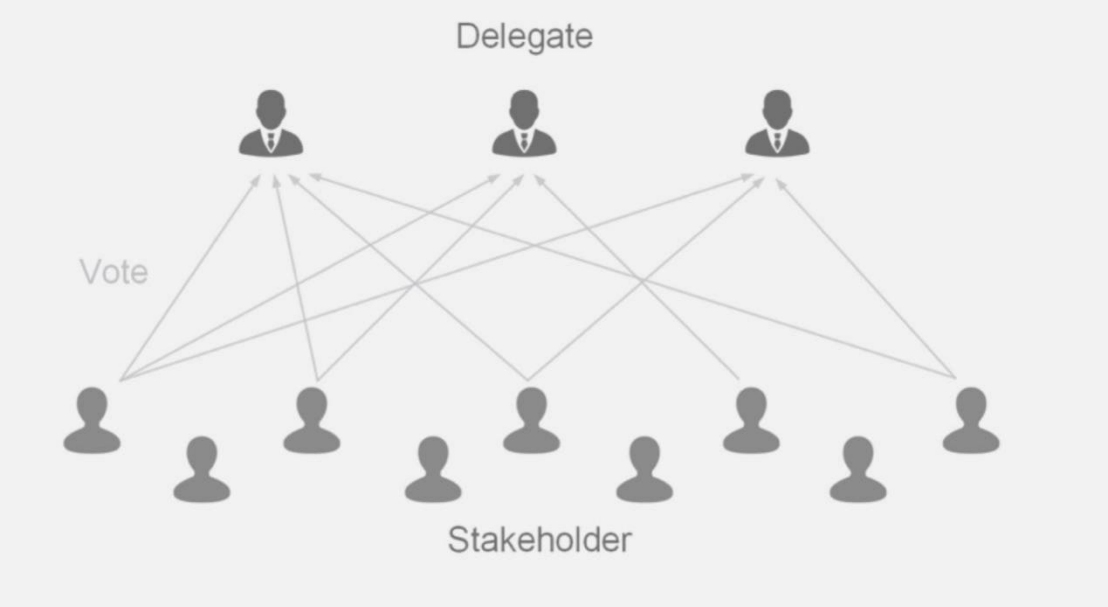


그림 4-2 위임자 선거

4.2.2 DPOS 설계

DPOS 는 Usechain 의 서브 체인에 사용되고 있다. 또한 DPOS 는 Usechain 의 블록체인 네트워크 중 중요한 일환으로서 코인 소지 노드와 위임자 노드가 공동으로 블록체인의 운영을 수호한다.

UST 를 소지한 노드는 투표권이 있으며 한차례의 투표 주기동안 투표권을 한 번만 행사한다. 투표는 거래 발송을 통해 실현된다. 투표 가중과 노드가 소지한 UST 는 정비례를 이룬다. Usechain 의 투표는 두 가지(지지표와 반대표)로 나뉘고 Usechain 은 한 개의 데이터 구조에서

온체인의 모든 투표 정보에 관해 저장 통계를 진행하는 과정을 수호하며 검색 인터페이스를 제공한다. 이밖에도 Usechain 은 정기적으로 효력을 상실한 투표를 삭제하여 공간을 절약한다.

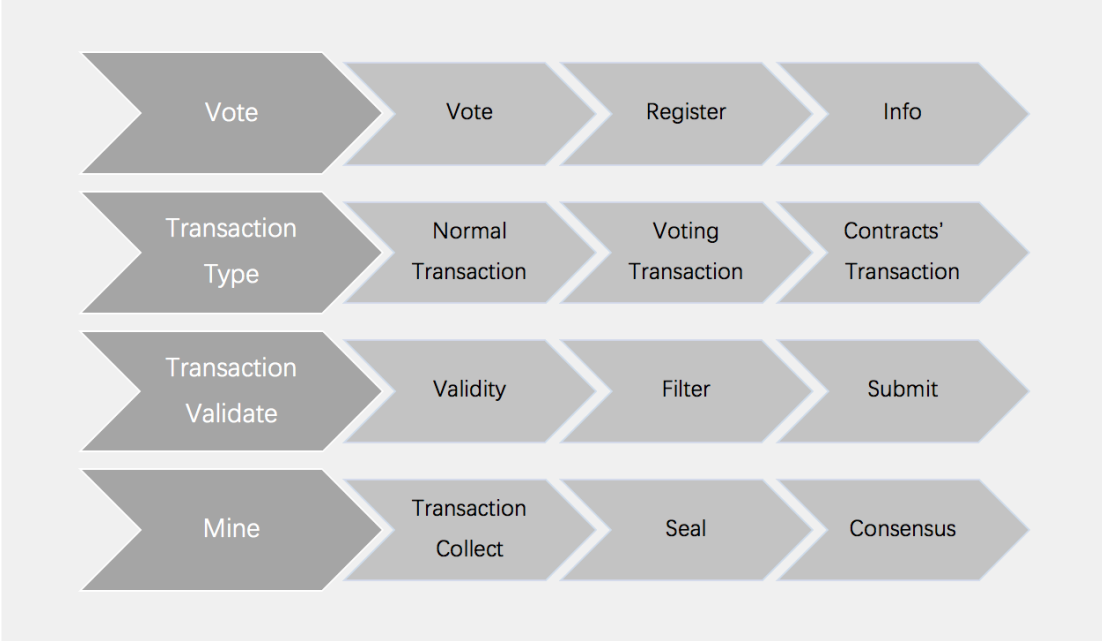


그림 4-3 DPOS 거래 과정 처리

POW 에 반해 DPOS 의 마이너 관계는 경쟁이 아닌 합작이므로 DPOS 는 블록포크에 대해 더 많은 처리를 할 필요가 거의 없다. 일부 노드가 고장나거나 악성 노드인 경우가 발생해도 기타 노드는 여전히 최대로 긴 한 갈래의 체인을 유지할 수 있다. 고장이 빈번하거나 악성 행위가 확연히 드러나는 노드는 다음 라운드의 위임자 선거에서 아웃시킬 가능성이 높으며 일반 노드는 투표 선거를 통해 새로운 위임자를 선출할 수 있다.

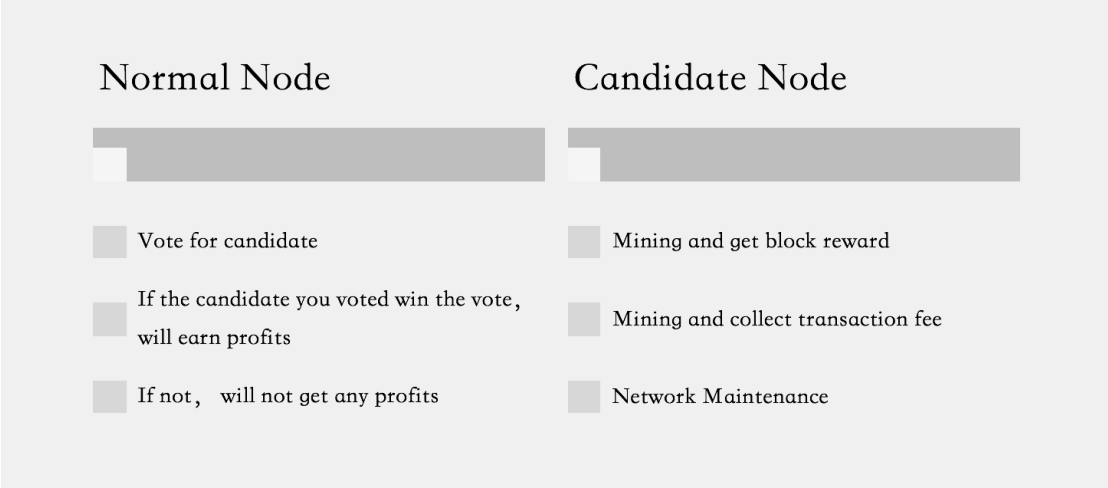


그림 4-4 코인 소지 노드와 위임자

DPOS 는 거래 처리 속도를 대폭 높였으며 위임자에게만 채굴권을 부여한 설계는 DPOS 가 POW 와 같이 산력 경쟁으로 인해 발생하는 자원의 극심한 소모를 피할 수 있도록 한다. 하지만 DPOS 는 아직 개선할 부분이 많은데 DPOS 블록체인에서 위임자 노드는 블록 포장을 마친 후 2/3 이상 위임자의 확인을 거쳐야만 해당 블록이 불가역적 가능성이 있다는 것을 증명할 수 있다. 블록을 확인할 때 본 프로젝트는 PBFT(Practical Byzantine Fault Tolerance) 합의 알고리즘을 도입하여 확인 속도를 빠르게 할 수 있다. 다시 말해 BFT-DPOS(PBFT 에 기초한 DPOS 합의)인데 기본원리는: 만약 현재 51 명의 위임자 노드가 있고 각 노드마다 연속으로 5 개 블록을 생산하며 블록 사이 간격이 1 초일 때(블록 생산 빈도를 높이는 동시에 위임자의 교대 주기에 변화가 없음을 보장해야 한다. 다시 말해 똑같은 블록 생산 간격 조건 하에 동일한 위임자가 더 많은 블록을 생성하도록 한다) 위임자 노드는 블록을 생성한 후 해당 블록을 즉시 기타 50 명의 위임자에게 발송하며 35 명 이상(2/3 이상)의 위임자의 확인을 받으면 해당 블록은 불가역 블록으로 판단된다. 이런 방법은 거래 속도를 대폭 높일 것이다. 또한 네트워크 지연 문제가 블록 생산에 미치는 영향을 없애기 위해 블록 생산 전에 51 명의 위임자가 서로의 네트워크 지연 현상을 감안하여 최적화한 블록 생산 순서를 계산해 내도록 한다.

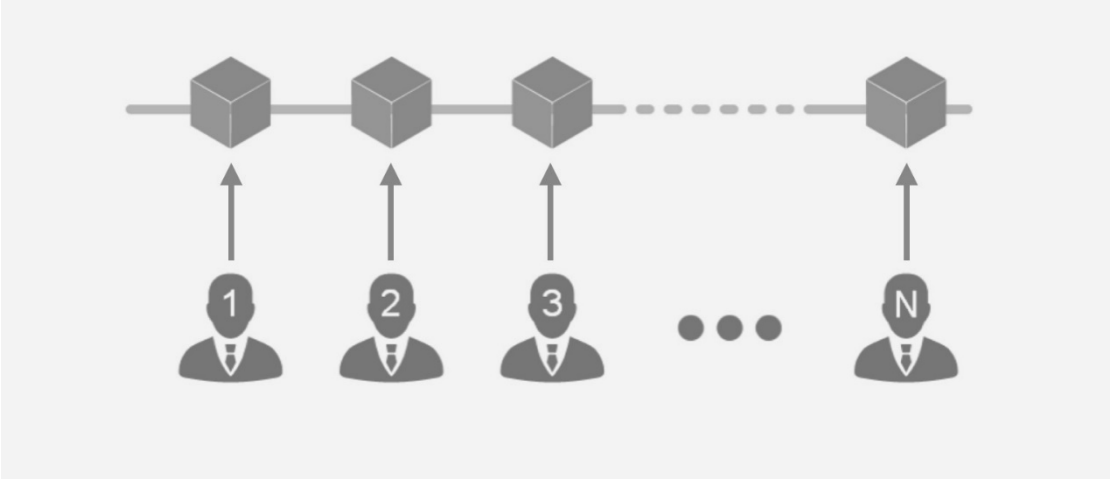


그림 4-5 위임자 블록 생산

4.2.3 RPOS

DPOS 는 공격을 저항할 때 위임자 대표의 사람수가 적어 한 명의 공격자가 교대로 위임하는 블록 생산 대표에게 차례로 서비스 거절 공격을 진행할 수 있다. 사실상 각 대표의 표지는 IP 주소가 아닌 퍼블릭 키여서 이런 특정 공격은 어느정도 줄일 수는 있지만 완전히 피할 수는 없다. 블록 생산 대표에는 예견할 수 있는 순서가 있으며 공격자는 여전히 생산자의 IP 주소를 통해 공격을 가할 수 있다.

이런 문제점을 피하기 위해 DPOS 알고리즘을 기반으로 RPOS 알고리즘을 제출하였다. RPOS 알고리즘은 다음 생산자를 선출할 때 순차적이 아닌 랜덤 선출의 원칙을 지키고 있다. 다음 생산자의 신분은 B 의 프라이빗 키가 앞 블록에 서명을 진행하는 방법을 통해 서명이 해당 블록에 부착된 후 서명값에 따라 다음 생산자의 ID 를 계산해 낸다. 이런 방법은 A 가 블록을 생산한 후 다음 생산자 B 가 다음 생산자의 ID 를 계산해 낼 수 있게 한다. 그리고 block01 에 서명을 하였기 때문에 생산자 B 는 다음 생산자의 ID 를 통제할 수 없고 해당 ID 는 앞 블록의 해시값에 따라 확정된다. RPOS 알고리즘의 주요 과정은 하기 참조:

1) 각 투표주기마다 신원 인증을 통과한 메인 계정은 원하는 위임자에게 투표할 수 있으며 1 인 1 표여야 한다. 서브 주소는 투표를 진행할 수 없다;

2) 투표 주기가 끝나면 시스템에서 자동으로 득표수를 통계하고 상위 N 명의 후보는 위임자로 선출되며 차기 위임자 주기의 블록 생산을 책임진다;

3) 위임자의 블록 생산 주기 동안 위임자는 블록을 포장할 때 프라이빗 키로 앞 블록과 위임자 블록 생산 메인 계정 주소에 서명을 진행한다. 만약 위임자의 퍼블릭/프라이빗 키가 (sk, pk) 이면 랜덤숫자 $k \in [1, l-1]$ 를 선택하며 계산 공식은 하기 참조:

$$r = H(A||pk||m)\ mod\ n$$

$$s = k - r \bullet sk\ mod\ n$$

그중 H 는 해시 함수이고 m 은 서명 정보이다. 다시 말해 앞 블록의 해시값에 해당 위임자의 주소 정보를 더하는 것이다. (r, s)는 서명값이다;

4) 공식으로 얻은 서명값 (r, s)를 해당 블록에 기입한다;

5) 서명값 (r, s) 으로 한 개 블록의 블록 생산자를 계산해 낸다. 계산 공식은 하기 참조:

$$ID = H(r, s)\ mod\ n$$

그중 H 는 해시 함수이고 mod 는 나머지 계산이며 n 은 위임자 수량이다;

6) ID가 부합한 위임자가 블록을 생산한다;

7) 블록 생산권을 획득하지 못한 위임자는 블록을 생산할 수 없으며 블록이 생성된다 할지라도 기타 노드의 검증을 통과할 수 없다.

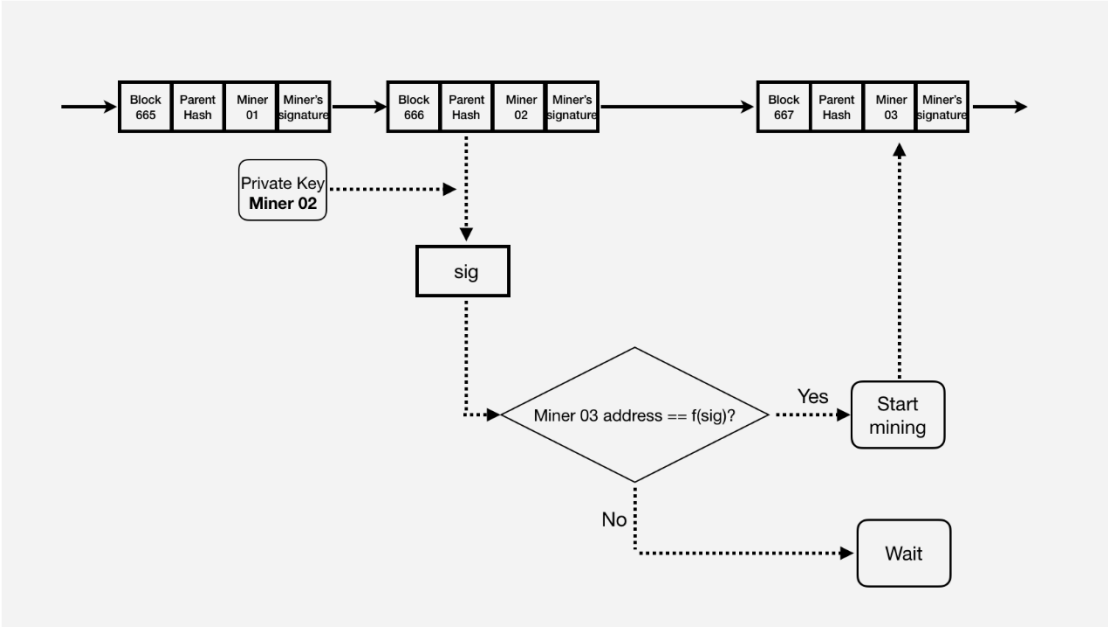


그림 4-6 RPOS 기본 작업과정

만약 A, B, C 가 모두 공격자이고 서로 결탁한 상태이면 A 가 블록을 포장할 시에 블록 내용을 부단히 조정하여 B 가 서명한 데이터가 특정된 생산자를 선택하거나 심지어는 공격자 C 를 지정하게 하여 생산 블록의 권력이 공격자 A, B, C 내부에서만 전전하게 함으로써 기타 생산자가 생산권을 획득할 수 없게 할 수 있다. 이러한 문제점을 해결하기 위해 생산자가 생산권을 획득하는 평균 횟수가 이론상 가능한 횟수를 훨씬 초과할 때 생산권 획득 대표로 선임되는 확율과 보상을 줄이는 방법, 나아가 징벌을 내리거나 대표위원회 자격을 박탈하는 등 방법을 적용한다. ID 체인의 모든 거래, 모든 마이너는 대응되는 신원이 있어 주소가 악행을 저지르는 원가를 대폭 인상하였고 네트워크가 악성 노드를 추적하고 식별하는 과정을 더욱 용이하게 하였다.

정상적인 네트워크 환경 아래 RPOS 가 선출한 대표는 100% 온라인 상태이며 모든 정보는 2s 안에 전체 대표에게 발송될 것이다. 다시 말해 한 차례 거래가 1.5s 뒤에 블록체인에 기입되는 것과 동시에 모든 블록 생산 노드에서 해당 거래에 관해 알게 됨을 의미한다.

만약 네트워크 전체가 체증되거나 일부 노드가 다운되었을 때 블록체인은 포크 현상이 나타날 수도 있다. 그러면 거래의 유효성을 보증하기 위해 한 개 수량의 블록의 확인을 기다려야 한다. POW 알고리즘의 경우 블록 길이가 가져다 주는 산력 장벽을 전적으로 의지하고 있어 거래 불가역 결과를 기다리는 주기가 길어질 것으로 예상된다.

RPOS+TaPos 의 전체 시스템을 볼 때 한 차례 거래는 최근 블록의 해시값(5 개 이내의 블록을 리콜)을 포함하고 있으며 이런 방법으로 포크 블록체인에 대량의 거래 기록이 나타나는 것을 피할 수 있는 동시에 사용자가 포크

블록체인에 있는지 여부를 시스템이 판단할 수 있다. 만약 한 개 노드가 연속 2 회 블록을 잃어버리는 현상이 발생하면 해당 노드는 포크 블록체인에 처할 가능성이 95%에 달한다. 그리고 연속 3 회 블록을 잃어버릴 시에 포크 블록체인에 처할 가능성이 99%이다.

4.3 고속채굴허가의 토크나이제이션(tokenization)

현재 채굴은 폭리를 가져다 주는 산업으로 거듭나고 있다. 2017 년 비트랜드의 이윤은 25 억 USD 에 달한다. 하지만 마이너의 폭리는 전체 생태 시스템에 불리한 영향을 미친다. 한편으로 소수의 고속채굴자는 대량의 비트코인을 소지하고 있어 비트코인의 집중 분포 현상이 나타나게 한다. 다른 한편으로 채굴 원가는 자원 낭비를 초래함과 동시에 거래 원가로 인해 실제 애플리케이션이 뿌리를 내리기 어렵다. 본 프로젝트는 고속채굴허가 token 인 USMK 를 도입하였다. USMK 의 원가는 2 의 K 승(UST+USK)이다. USMK 를 소유하고 있는 실명 계정은 채굴 난이도를 K 개의 0 으로 낮출 수 있고 마이너는 수시로 USM1 을 구입할 수 있다. 하지만 USMK 를 구입하는 과정 중 K 가 1 보다 큰 전제 조건은 USM(K-1)을 이미 20 매 소유한 상태여야 한다. 이와 마찬가지로 저레벨 USMK1 을 소유한 사용자는 USMK2 로 레벨업 할 수 있으며 K2 가 K1 보다 크다. 그 전제 조건은 USM(K2-1)을 20 매 소유한 상태여야 하고 가격 차이를 메꾸어야 한다. USMK 는 체인 내부 거래에도 활용될 수 있지만 USMK 를 사용하는 계정이 USMK 를 소유한지 24 시간 이상 되어야지 사용 가능하다.

랜덤으로 마이너를 선택하며 앞 블록의 해시값에 NUSMK+1 을 제한 나머지 숫자에 대응되는 마이너는 정상적인 난이도보다 K 개 0 이 줄어든 난이도로 블록체인의 온체인을 획득할 수 있다. 만약 나머지가 0 이면 회사에서 퍼블릭 계정을 대신하여 채굴하며 채굴 과정 중 얻은 보상은 퍼블릭 계정으로 돌린다. NUSMK 의 수량은 USMK 와 대응되어야 한다. 또한 그 어떠한 계정도 연속 두 개의 블록을 채굴할 수 없다.

5. 샤딩과 서브 체인

현재 블록체인의 특성은 모든 노드가 동시에 거래를 전파하고 스마트 컨트랙트의 집행 결과를 검증하는 동시에 각 노드의 stateDB 일치성을 보증하는 것이다. 스마트 컨트랙트의 이용 빈도가 높아지고 거래 수량이 많아짐에 따라 거래 확인에 필요한 시간도 길어진다. 그리고 점점 많아지는 거래 수량으로 인해 네트워크는 뚜렷한 체증현상이 나타난다. 만약 전체 네트워크의 거래 처리 속도가 일부 슈퍼 노드의 속도에만 의존하고 있으면 네트워크 산력이 극소수의 슈퍼 노드의 통제를 받게 된다. 이런 전제 하에 Usechain 은 부동한 네트워크에서 거래를 처리하는 방법으로 거래 처리와 확인 속도를 대폭 높일 것이다.

5.1 샤딩

ID 체인을 기반으로 각 노드마다 신원 인증의 특성을 지니고 있으며 Usechain 은 샤딩 매커니즘을 설계하여 거래 처리 속도와 확장성을 높인다. 각 샤딩은 대응되는 거래 처리 노드가 있으며 샤딩 사이에서도 동시에 거래를 처리할 수 있고 스마트 컨트랙트의 집행 검증을 진행할 수 있다. 각 샤딩은 부동한 샤딩 ID 가 있으며 고정으로 선출한 노드에서 수집 거래를 진행하고 일정 시간 동안의 거래를 한 개의 블록으로 포장하는 동시에 서명을 진행하고 위층에 넘긴다. 그리고 위층의 합의 노드가 부동한 샤딩의 블록을 블록체인에 포장하여 넣는다. 현존하는 기타 퍼블릭 블록체인과는 달리 ID 체인은 사용자의 인증 매커니즘과 폴트 톨러런스 매커니즘으로 네트워크의 안전성을 보증할 수 있다. 샤딩의 거래 처리 속도를 감안할 때 샤딩 내부의 노드 수량은 너무 많으면 아니 된다. 이런 방법으로 미러 아이덴티티 블록체인은 네트워크를 더 많은 샤딩으로 나눌 수 있고 네트워크의 전반 거래 처리 성능을 제고할 수 있다.

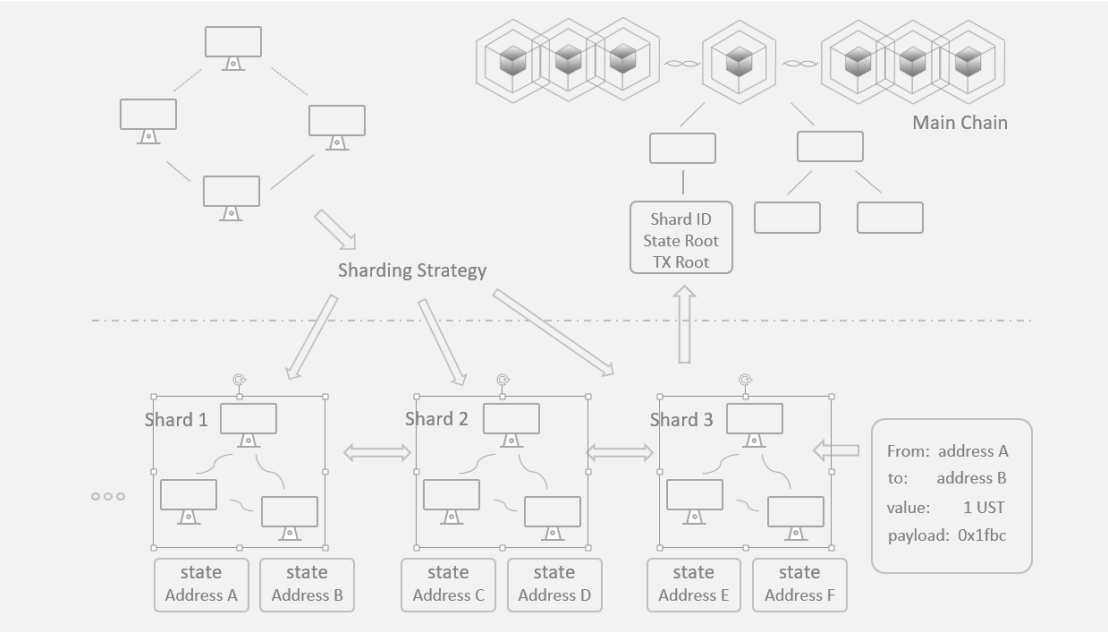


그림 5-1 샤딩 시스템

샤딩은 두 층으로 구성된다. 위층 합의 노드는 한 갈래의 메인 체인을 실행하고 메인 합의 노드는 샤딩 합의 노드에서 수집한 샤딩 거래를 접수하며 메인 체인은 합의 노드의 거래도 동시에 처리한다. 샤딩 합의 노드는 샤딩 내부의 거래를 단독으로 처리하며 샤딩 수량이 많아짐과 동시에 더 많은 거래를 처리할 수 있다.

5.1.1 메인 체인 합의 매커니즘

메인 체인은 RPOW 의 합의 알고리즘에 따라 실행되고 있는데 랜덤으로 노드를 생성하여 블록을 포장하는 방법으로 마이너 풀이 네트워크를 통제하는 현상이 나타나는 것을 방지한다. 또한 일정 시간을 간격으로 RPOW 합의 매커니즘을 통해 위층 네트워크 노드를 선출해 내며 위층 합의 노드는 샤딩 노드가 수집한 거래를 검증하고 서명을 마친 후 블록체인에 넣는다.

위층 네트워크의 특징은 하기 참조:

1) 슈퍼 노드 혹은 마이너 풀이 나타나는 것을 방지하기 위해 위층 네트워크 노드는 RPOW 합의를 통해 랜덤으로 선출된다;
2) 위층 네트워크의 블록 포장 노드는 전부 랜덤으로 생성된다;

3) RPOW 는 랜덤으로 실행되고 있지만 부동한 노드가 결탁하는 확율을 낮추기 위해 샤딩 노드는 정기적으로 업데이트 된다.

5.1.2 샤딩 합의 매커니즘

샤딩 노드 내부에는 자체 합의 매커니즘이 있으며 샤딩 내부의 합의 노드는 RPOS 를 기반으로 투표하여 선거한다. 노드는 투표 전에 메인 체인의 계약에 등록하여야 하고 일정 금액의 보증금을 지불하여야 한다. 그리고 샤딩 노드를 선출해낸 후 해당 샤딩 내부의 거래는 합의 노드를 거쳐 확인되며 블록으로 포장을 마친 뒤 포장을 진행한 노드가 서명을 하고 위층 네트워크 노드에 넘기며 위층 네트워크 노드가 블록의 합법여부를 확인 뒤 블록체인을 업데이트 한다.

샤딩의 특징:

1) 위층 네트워크 노드에서 일정 시간의 $T_{checkpoint}$ 를 합의 후 새로운 네트워크 샤딩을 선택한다;

2) 각 샤딩 내부의 노드는 자체 샤딩 내부의 거래만 처리하고 부동한 샤딩 사이는 통신을 진행할 수 있다;

3) 샤딩의 가입은 합의 과정을 단축할 수 있고 거래에 관해 더 빨리 합의를 달성할 수 있다.

5.1.3 샤딩 거래 처리

메인 체인은 사용자의 메인 주소, 서브 주소의 거래를 지원할 수 있고 RPOW 합의를 통해 거래를 포장한다.

샤딩 내부의 거래는 단독으로 권한을 부여 받은 노드에서 거래를 수집하고 위층 메인 체인에 넘겨 포장한다. 그러므로 샤딩의 수량을 늘려 거래 처리 속도를 높일 수 있다. 사용자의 부동한 샤딩 내부에서의 주소는 사용자 메인 주소 퍼블릭 키, 위원회 퍼블릭 키와 샤딩 ID 등을 통해 생성되며 사용자의 부동한 샤딩 내부에서의 주소를 확인하는 유일한 근거로 된다.

사용자가 부동한 샤딩 사이에서 거래를 진행할 시에 본인이 샤딩 내부에서의 주소는 관리할 필요가 없으며 메인 주소만으로 크로스 샤딩의

거래 과정에 참여하고 스마트 컨트랙트를 집행할 수 있다. 크로스 샤딩의 거래는 일단 한 개의 샤딩 내부에서 처리되며 해당 거래가 메인 체인의 확인을 받고 기타 샤딩의 거래 주소에서 메인 체인의 확인 통보를 받은 뒤에야 롤 아웃 작업을 진행할 수 있다. 샤딩 수량을 늘리면 Usechain 의 거래 처리량을 증가할 수 있고 노드는 샤딩 매커니즘을 통해 샤딩을 연결 구성한다. 샤딩의 보톰 레이어는 P2P 네트워크를 통해 연결되며 거래는 일정 법칙에 따라 동일한 샤딩 내부에서 전파된다. 이런 방법은 크로스 샤딩의 빈번한 확인 과정을 피할 수 있다. 거래의 발기인을 일정 시간 뒤 부동한 샤딩에 배치하는 것은 악성 노드가 시종일관 동일한 샤딩에 영향을 미치는 현상을 방지할 수 있다.

5.2 서브 체인과 크로스 체인 거래

크로스 체인 기술은 여러 블록체인을 연결하는 다리로 볼 수 있으며 주로 여러 블록체인 사이의 원자적 트랜잭션(Atomic Transaction), 자산전환, 블록체인 내부 정보 공유 혹은 Oracle 문제의 해결 등에 이용될 수 있다. 크로스 체인은 일종의 복잡한 과정이며 인터체인 노드에 대한 단독적 검증능력이 있어야 하고 탈중앙화의 입력을 필요로 하며 체인 밖 세계의 정보를 획득하고 검증하는 능력이 있어야 한다. 블록체인 세계에서 각 체인마다 모두 독자적인 장부를 이루고 있으며 그들 사이에는 아무런 연관이 없다. 본질적으로 가치는 부동한 장부 사이를 전전할 수 없지만 특정 사용자가 한 갈래 블록체인에 저장한 가치는 다른 한 갈래 체인의 가치로 변할 수 있으며 이런 과정을 크로스 체인 거래라 한다. 다시 말해 크로스 체인이란 부동한 소지자 사이의 교환을 의미한다.

본 프로젝트는 메인 체인과 샤딩을 설계할 때 크로스 체인 거래도 감안한 상태이다. 서브 체인은 RPOS, RPOW 등 합의 알고리즘을 실행하고 있고 서브 체인을 실행하는 주체는 본 프로젝트의 협력 파트너이며 서브 체인은 단독 실행이 가능하다. 현재 실행 가능한 크로스 체인 거래의 방법으로는 스마트 컨트랙트를 이용하여 메인 체인에 앵커링(anchoring)한 코인과 서브 체인의 코인을 통해 크로스 체인의 계좌 이체를 실현하는 것이다. 또한 샤딩 기술은 크로스 체인 계좌 이체를 확인하는 속도를 높일

수 있다.

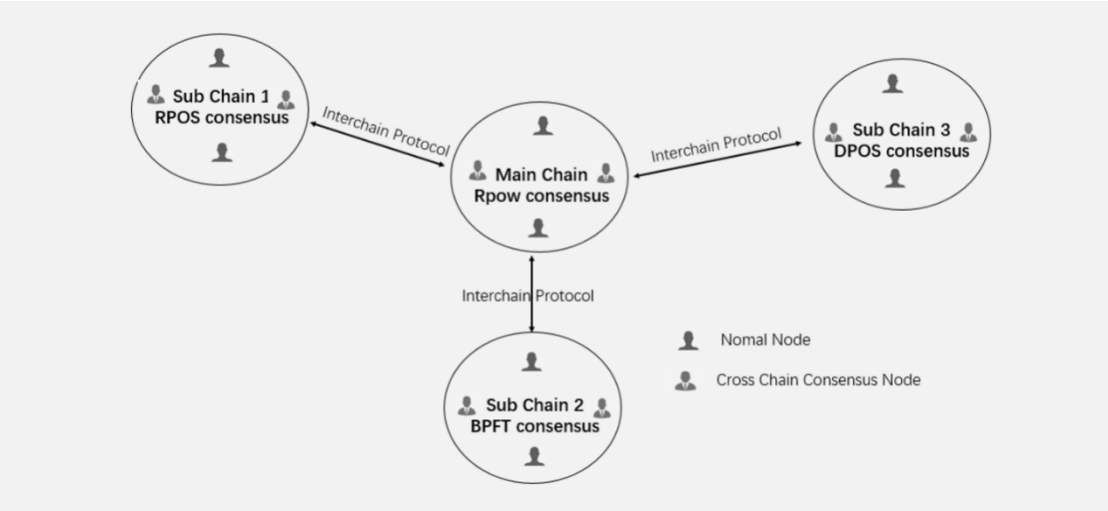


그림 5-2 크로스 체인 프로토콜

6. P2P 네트워크

현재 퍼블릭 블록체인의 주요문제는 확장성이다. 사용자 수량이 증가되고 거래량이 커지는 반면 네트워크의 전반 거래 속도에는 변화가 없다. 다시 말해 현재 블록체인 네트워크 중의 거래, 스마트 컨트랙트의 저장과 사용 빈도가 높아지는데 비해 거래의 확인 속도에는 현저한 변화가 없다. 현재 비트코인의 거래 확인 속도는 7 Tx/s 이고 이더리움의 거래 확인 속도는 25 Tx/s 인데 전반 네트워크의 통일된 합의 매커니즘이 거래 속도를 제한하고 있어 한 차례 거래를 확인하는 속도가 지나치게 늦는 현상을 초래하게 된다. 이에 대응하는 해결방도로는 비트코인이 블록크기를 확대하여 더 많은 거래와 서브 체인 기술의 애플리케이션을 포함하는 것인데 이런 방법은 메인 체인의 거래 성능이 지나치게 낮은 문제를 근본적으로 해결하지 못하고 있다.

6.1 계층적 P2P 네트워크

KaZaA 프로토콜과 이전 네트워크 샤딩의 수요를 기반으로 대응되는 보톰 레이어 P2P 네트워크 구조를 구축한다(그림 6-1 참조). 네트워크 노드는 슈퍼 노드 SN 과 일반 노드 ON 으로 나뉘는데 SN 은 대역폭이 넓고 처리 능력이 강하며 저장 용량이 큰 특징이 있고 NAT 제한을 받지

않으며 장시간 온라인 상태를 유지하는 서비스 능력이 있다. 모든 ON 은 네트워크에 접속할 때 한 개의 부모 SN 과 반영구적인 TCP 연결을 유지하며 일반 노드는 자신이 공유한 온체인 데이터 해시를 업로드 한다. 부동한 SN 사이는 분산화 P2P 프로토콜을 실행하고 있으며 이런 방법으로 장기적인 TCP 연결성, 강인성과 빠른 데이터 분석능력을 유지할 수 있고 슈퍼 노드 커버리지 네트워크를 구축할 수 있다.

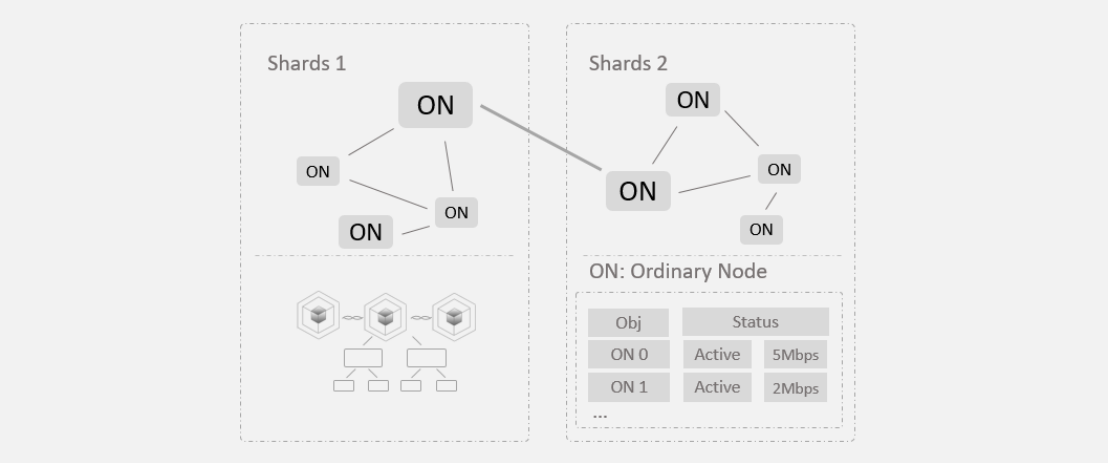


그림 6-1 P2P 네트워크 노드

6.2 네트워크 샤딩

네트워크 샤딩은 노드를 관리하는 일종의 매커니즘이며 일정한 샤딩 법칙에 따라 자동으로 노드를 부동한 네트워크 샤딩으로 구분한다. 그리고 각 샤딩 네트워크 내부에 존재하는 노드는 고정된 노드인데 모든 네트워크 샤딩이 병행으로 거래를 처리할 수 있어 전반 네트워크 처리량이 직선적성장을 보일 것으로 예상된다.

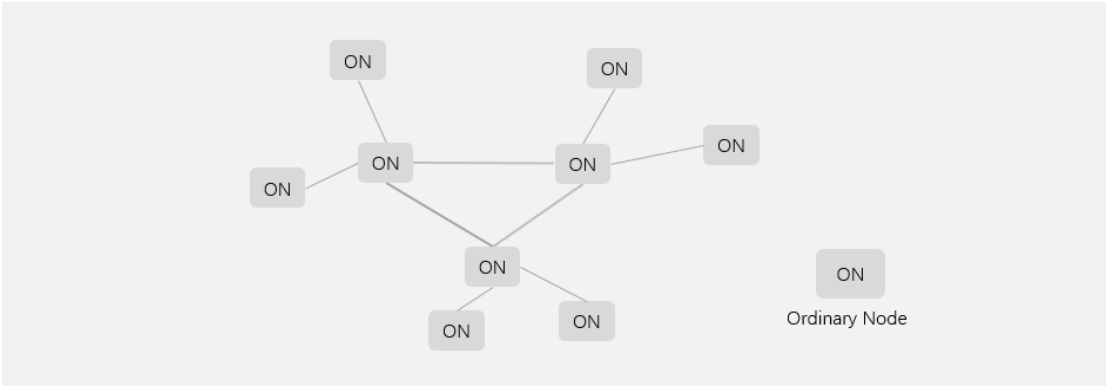


그림 6-2 네트워크 샤딩

네트워크 샤딩 내부의 노드는 대역폭의 변화에 따라 가감된다. 일정한 네트워크 상태에 처한 노드는 자발적으로 네트워크 샤딩을 구성하며 한 개의 네트워크 샤딩 중의 노드가 일정량에 도달했을 때 기타 노드는 자발적으로 다른 네트워크 샤딩으로 전이한다.

7. 스마트 컨트랙트와 가상 머신 IVM

Usechain 은 탈중앙화의 애플리케이션 플랫폼으로 Usechain 밖에도 여러가지 다른 유형의 ID 체인 애플리케이션을 지원하고 사용자와 회사가 자체의 애플리케이션 시스템을 구축하는 것을 허락한다. 금융거래, 신용증명 혹은 더 복잡한 애플리케이션 프로그램도 Usechain 에서 자동으로 믿음직하게 실현할 수 있는데 Usechain 은 학위인증시스템, 보험공유, 소액대출, 투자공유 등 분야에서 더 많은 혁신을 가져올 것으로 보인다.

본 프로젝트는 퍼블릭 체인의 용도를 감안하여 강화된 ERC20 인터페이스를 설계할 것이며 인터페이스에 신원 정보의 인출을 내포하고 있어 정규적이고 통일된 신원 token 의 설계와 개발을 실현할 수 있다.

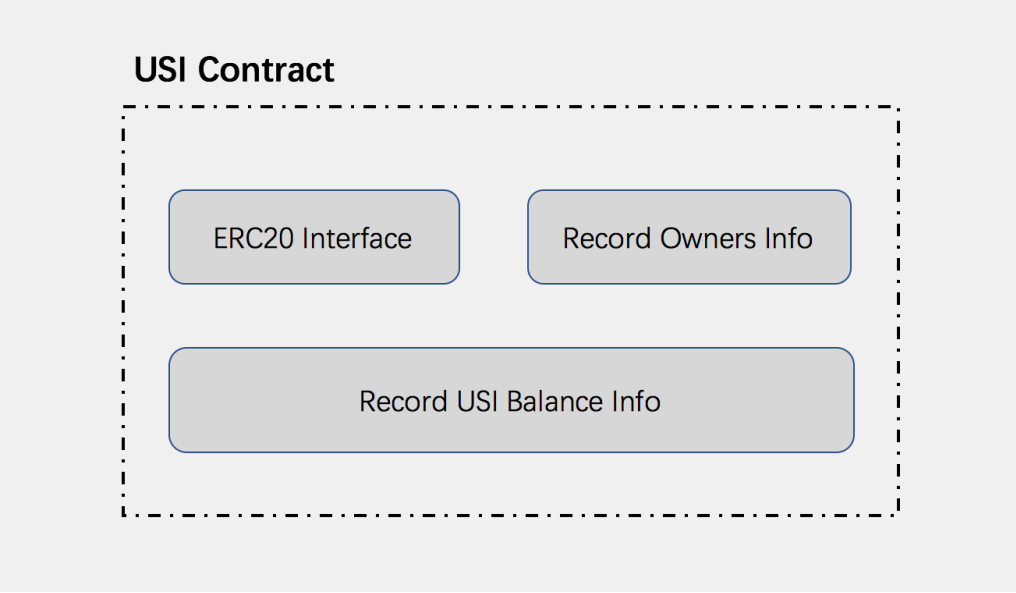


그림 7-1 ERC20 인터페이스 개선

7.1 컨트랙트

컨트랙트는 탈중앙화한 이더리움 네트워크 중의 모든 노드가 공동으로 배치하고 실행하여 컨트랙트의 정확성, 안정성과 불가역성을 보증한다. 컨트랙트의 내용은 코드에서 컨트랙트를 지정하는 법칙에 따라 EVM 에서 집행한다. 또한 컨트랙트를 실행할 때 제출자는 소량의 Gas 비용을 지불하여 이더리움 네트워크의 정상적인 운영을 격려해야 한다.

7.1.1 애플리케이션 레이어의 확장

Usechain 은 다양한 애플리케이션 레이어 프로토콜과 컴포넌트를 제공하여 여러가지 부동한 수요의 ID 체인 애플리케이션을 지원할 수 있다. 애플리케이션 개발자는 보톰 레이어의 데이터 인터랙티브를 확인할 필요 없이 빠른 속도로 탈중앙화 애플리케이션을 구축할 수 있다. Usechain 애플리케이션 프레임워크는 높은 확장성이 있으며 사용자는 부동한 환경 수요에 따라 부단히 확장할 수 있다.

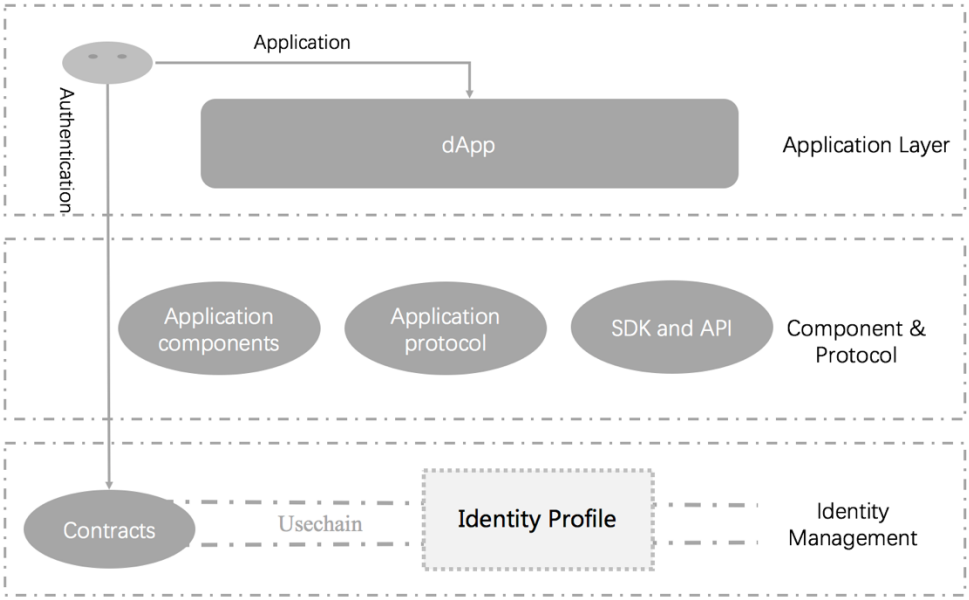


그림 7-2 Usechain 애플리케이션 프레임워크

1) Usechain 은 탈중앙화한 미러 아이덴티티 시스템과 스마트 컨트랙트 등 방법으로 분산화 신뢰를 실현하였다;

2) 프로토콜을 통해 컴포넌트와 API 는 위층 애플리케이션에 인터페이스를 제공하여 애플리케이션이 Usechain 의 기능을 손쉽게 사용할 수 있도록 한다;

3) Usechain 은 부동한 환경에 처한 DApp 를 주목하여 신원 인증 서비스를 제공하고 보톰 레이어로부터 신뢰 문제를 해결한다.

7.1.2 컨트랙트 스피드 업(코드 레이어)

스마트 컨트랙트는 이더리움에서 가장 흔하고 최고로 강력한 기능이며

외부 계정의 peer-to-peer 기본 계좌 이체 외에도 기타 계좌 이체 방식과 논리적 실무(예를 들어 다중 서명, 인출 지연, 기관 위탁 등)에 스마트 컨트랙트 기능을 사용할 수 있다. 다시 말해 스마트 컨트랙트가 이더리움에서 나타나는 빈도는 매우 높으며 스마트 컨트랙트의 코드 내역은 해당 컨트랙트의 집행 효율을 직접적으로 결정할 수 있고 나아가 이더리움의 실제 처리 능력에 영향을 미칠 수 있다.

7.1.3 Solidity 의 단점 및 개선

이더리움의 스마트 컨트랙트는 Bytecode 로 구성된다. Stack 을 기반으로 보톰 레이어 작업을 진행하는 Bytecode 는 복잡한 컨트랙트를 프로그래밍 하는데 적용될 수 없어 이더리움은 간단한 고급 프로그래밍 언어인 Solidity 를 도입하였다. Solidity 는 JS 와 흡사한 객체 지향 언어이며 배우기 쉽고 계승을 지원하며 라이브러리 호출 등이 가능해 사용자는 solidity 로 간편하고 빠르게 컨트랙트를 구축할 수 있다.

Solidity 가 현재 출시한 최신 버전은 0.4.21 인데 정식 버전처럼 안정적인 버전이 아니다. 이더리움 스마트 컨트랙트의 규범화 문제와 확장 기능은 아직 해결을 보지 못한 상태이며 Solidity 는 다른 고급 프로그래밍 언어에 비해 여전히 많은 문제점들이 존재한다. 예를 들면:

1) Solidity 는 Calldata, Callcode 의 원본 내용을 직접 추출할 수 없다;

2) 동적크기(dynamic length) 유형의 Bytes 와 고정크기(fixed length) 유형의 Bytes21, Uint256 은 서로 직접적인 전환이 불가능하며 Bytes 가 256bit 의 메모리 공간을 배치하였다 할지라도 Solidity 는 바이트를 하나씩 기입해야 하며 이러한 방법은 대량의 Gas 를 허비한다.

상기 문제점들은 solidity 에 Assembly 를 사용하여 개선하거나 해결하여 gas 를 절약할 수 있고 코드의 집행 효율을 높일 수 있다.

7.1.4 Assembly 의 사용 및 리스크

고급 프로그래밍 언어인 Solidity 의 실제 작업은 편집을 마친 바이트 코드에서 집행하며 다른 고급 프로그래밍 언어와 같이 사용자가 직접 바이트 코드를 라이닝하여 프로그래밍 할 수 있다.

사용자는 Solidity 에서 Assembly 코드 블록으로 바이트 코드를 라이닝할 수 있고 가독성을 높이기 위해 Assembly 의 대다수 바이트 코드는 함수 형식으로 편집할 수 있는데 이런 방법으로 바이트 코드를 직접 편집하는 어려움을 피할 수 있다.

위에서 언급한 문제는 Assembly 에 라이닝한 어셈블러 코드로 해결할 수 있다.

1) 명령어(예를 들어 Calldataload, Calldatacopy 등)를 직접 사용할 수 있어 TX 의 원본 데이터를 직접 방문할 수 있다.

2) 공간을 배치한 동적크기배열(arrey)(예를 들어 Bytes, String)은 해당 데이터 세그먼트의 위치를 추적하는 방법으로 직접 대입할 수 있다. 예를 들면:

```
1. bytes32 b32 = 0x12345678;
2. bytes memory b = new bytes(32);
3. assembly {
4.         mstore(add(b, 32), b32)
5.         re := add(b, 32)
6. }
```

32 를 더해야 하는 이유는 동적크기배열이 Memory 에서의 첫번째 slot 이 길이이기 때문이다. 동적배열을 방문하고 데이터를 읽으며 대응되는 고정크기 유형 혹은 일반 유형에 저장한다:

```
1. assembly {
2. b32 := mload(add(b, 32))
3. }
```

이밖에도 동적크기 유형은 Solidity 에서 직접 전환할 수 있고 고정크기 유형도 같은 이치이다(크기가 다르면 잘리거나 제로화 할 가능성이 있다)

Assembly 를 이용하면 Solidity 의 일부 제한을 받지 않을 수 있으며 대량의 Gas 를 소모하여 바이트 순서대로 목표를 채울 필요가 없어 Gas 를 절약함과 동시에 컨트랙트 코드의 효율을 높일 수 있다.

Solidity 와 같은 고급 프로그래밍 언어의 초심은 개발자에게 편의를 도모하여 애플리케이션의 개발 속도를 단축하는데 그치지 않고 다른 고급 프로그래밍 언어와 가상머신과 같이 안전을 가장 중요한 요소로 간주하고 있다. 만약 개발자 마음대로 바이트 코드나 기계어 코드를 사용하여 프로그래밍을 할 시에 메모리 부족, 인덱스 혼동, 데이터 혼란 등 엄중하면서도 발견되기 어려운 문제들이 나타날 수 있다. 그러므로 안전과 성능 사이의 평행점을 찾기 위해 일정한 조건 하에서만 광범위한 인증을 받은 Assembly 코드 블록을 사용하여야 하며 일방적으로 성능을 높이기 위해 안전성을 낮추는 현상이 나타나는 것을 방지해야 한다.

7.2 가상 머신

현재 블록체인 분야에서 비교적 널리 사용되고 있는 것은 이더리움 가상 머신(Ethereum Virtual Machine, EVM)이다. EVM 은 이미 현존하고 있는 대다수 스마트 컨트랙트에 사용되고 있으며 여하한 알고리즘 난이도의 코드에도 사용될 수 있다. 블록체인 데이터 베이스는 블록체인 네트워크에 연결된 수많은 노드의 보호와 관리를 받고 있으며 각 노드마다 EVM 을 실행하여 스마트 컨트랙트 코드를 집행하고 있다. 탈중앙화 일치성은 전반 네트워크가 매우 높은 폴트 톨로런스 성능을 소유할 수 있도록 보장한다. EVM 에서 Call()함수를 예로 들면 먼저 계좌 이체 함수 Transfer(), 송금 계정 caller, 인출 계정 addr 를 호출한다; Contract 상대를 새로 만들고 멤버 변수인 caller, self(addr), value 와 gas 를 초기화 한다; Contract 상대의 Code, CodeHash, CodeAddr 멤버 변수를 대입한다; run()함수를 호출하고 해당 컨트랙트의 명령을 집행한다. 마지막으로 Call()함수가 돌아온다. 전반 과정은 지금 사용되고 있는 이더리움 블록 구조와 잘 맞다.

해당 코드는 하기 참조:

```
1.  func (evm *EVM) Call(caller ContractRef, addr common.Address, input
[]byte, gas uint64, value *big.Int) (ret []byte, leftGas *big.Int, error){
2.      var snapshot = evm.StateDB.Snapshot()
```

```
4.      contract.SetCallCode(&addr, evm.StateDB.GetCodeHash(addr),
evm.StateDB.GetCode(addr))
5.      ret, err = run(evm, snapshot, contract, input)
6.      return ret, contract.Gas, err }
```

EVM 은 간편성, 확정성, 안전성을 목표로 하고 있으며 블록체인 시스템을 위해 설계한 가상 머신이다. 하지만 EVM 은 현재의 주류 기술, 설계 범례와 맞지 않으며 설계부터 실현까지 모두 일정한 단점이 존재한다. 현재 EVM 의 스마트 컨트랙트 집행 효율은 매우 낮으며 암호 작성술을 실현하려면 solidity 가 아닌 go, C/C++ 혹은 더 효율적인 다른 언어를 선택해야 한다. 또한 메모리 분배 시스템을 볼 때 스마트 컨트랙트 중의 함수가 분배된 메모리를 사용 할 때 전적으로 프로그래머의 확인에 의존해야 하며 캐시메모리를 재사용할 시에 완비한 검측 과정이 없으면 컨트랙트는 잠재된 bug 가 발생할 수 있다. 그리고 EVM 의 스마트 컨트랙트는 시운전과 테스트가 어렵다. EVM 의 유일한 에러 상황은 gas 부족이며 컨트랙트 실행 과정 중 시운전 일지가 없고 외부 코드를 호출할 수도 없으며 표준 라이브러리가 부족하다. 현재 대부분의 컨트랙트 개발자들은 오픈 소스 소프트웨어에서 코드를 복사 후 붙여넣는 방법 외에 다른 방법이 없다. 스마트 컨트랙트의 배치와 실행은 이더리움 네트워크에서 대량의 gas 를 소모하고 있으며 그로 인해 좋은 코드를 써내는 것이 점점 어려워지고 비싸진다.

EVM 은 블록체인 네트워크에서 처음으로 출현한 성숙되고 널리 사용된 가상 머신이며 이 분야의 선두자라고 해도 과언이 아니다. 하지만 현존하는 일부 문제점을 감안하여 Usechain 가상 머신은 ID 체인 특유의 장점으로 이런 문제점들을 최적화하고 보완할 것이며 블록체인 가상 머신의 기능을 더 강대하고 실용적이며 안전하게 개선할 것이다.

본 프로젝트는 초기에 이더리움 가상 머신 EVM 을 기반으로 하는 환경을 그대로 사용할 것이며 이더리움 스마트 컨트랙트의 편집과 집행

과정이 간편하여 초급 애플리케이션의 개발에 적용될 수 있다. 그리고 후기에는 ID 체인 가상 머신 Identity Virtual Machine(IVM)을 구축할 것이다. IVM 은 강화된 스마트 컨트랙트를 구축하는 새로운 기준이며 적은 수량의 어댑터로 정의를 할 수 있고 샌드박스화 할 수 있다. ID 체인은 중간 통신층을 구축할 수 있으며 IVM 이 중간 통신층을 기반으로 블록체인 보톰 레이어와 데이터를 호환하게 한다. 이런 방법은 이더리움 가상 머신의 부족점을 보완하고 집행 효율이 높고 외부 데이터와 호환 가능한 스마트 컨트랙트를 구축하여 스마트 컨트랙트의 애플리케이션 환경을 확대한다. 스마트 컨트랙트의 완전성을 보증하기 위해 본 프로젝트는 IVM 의 컴파일러를 다시 설계할 것이며 내부에 스마트 컨트랙트의 완전성을 테스팅하는 알고리즘을 구축하여 컨트랙트에 잠재된 코드의 약점을 제시하도록 한다.

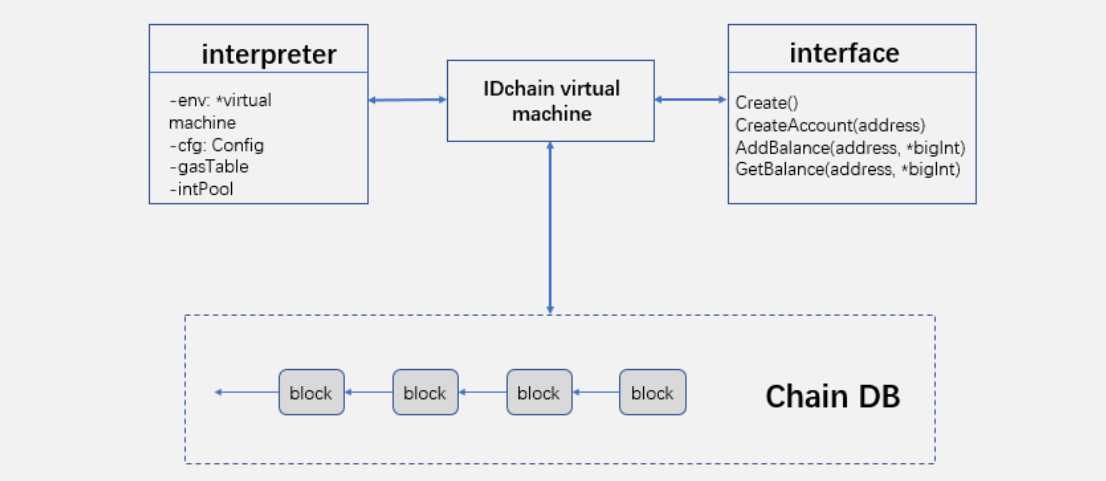


그림 7-3 ID 체인 가상 머신

향후 WASM 을 Usechain 가상 머신의 개발 기준으로 할 것이다. WASM 은 고성능의 Web 애플리케이션 프로그램을 구축하는 신흥 Web 기준이며 적은 수량의 어댑터로 정의를 할 수 있고 샌드박스화 할 수 있다. WASM 의 장점은 업계의 광범위한 지지를 받을 수 있어 익숙한 언어로 스마트 컨트랙트를 개발할 수 있는 것이다. 예를 들어 Go 언어 혹은 C 언어 등이다.

8. 라이트 노드 프로토콜

블록체인 네트워크의 블록 데이터는 방대하지만 일반 사용자를 놓고 볼 때 자신과 연관이 있는 거래 정보 혹은 기타 데이터 정보에만 관심을 가지게 된다. 그러므로 라이트 클라이언트는 모든 블록 데이터를 동기화 할 필요가 없다. 하지만 그와 동시에 데이터의 신뢰성을 보장해야 한다. 이러한 모순을 해결하기 위해 본 프로젝트는 경량급의 머클 증명을 도입하여 시스템이 모든 노드의 마이너를 전적으로 의지하지 않음과 동시에 모든 노드의 마이너가 동기화를 진행할 때의 비용을 최대한 줄일 수 있다.

비트코인 시스템에서 라이트 클라이언트는 각 블록의 블록 해더만 다운로드 하면 되는데 블록마다 앞 블록 해더의 해시값, 타임 스탬프, 채굴 난이도, 랜덤숫자와 블록 거래의 해시머클루트를 포함하고 그 크기는 80 바이트 밖에 되지 않는다. 비트코인 라이트 클라이언트는 블록 해더의 정보만으로 블록이 내포한 거래를 증명할 수 있지만 당시의 상태를 증명할 수는 없다(디지털 자산의 잔액 등). 그러므로 노드의 잔액 정보를 증명하기 위해 전반 노드의 블록체인에서 획득한 정보를 끊임없이 리콜하여야 한다.

Usechain 의 일부 애플리케이션 환경은 짧은 시간 안에 주소의 잔액과 신원 상태 증명을 알아낼 수 있는 것을 필요로 하고 있다. 이런 문제를 해결하기 위해 본 프로젝트는 이더리움에서 이미 실현한 방법을 참고하여 Merkle Patricia Tree(MPT)의 데이터 구조로 블록체인의 거래, 영수증과 상태에 관해 각기 별도로 저장한다. 그러므로 각 블록 해더마다 한 개가 아닌 세 개의 머클루트데이터를 내포하고 있다.

경량급 노드는 블록의 해더 데이터를 저장하고 P2P 네트워크를 통해 네트워크의 모든 노드에게 데이터 신청을 발송한다(Bloom filter). 또한 소량의 검증이 필요한 거래 내역을 저장하는데 이런 거래 내역은 지갑 안의 프라이빗 키가 대응하는 모든 거래이며 이런 방법으로 블록체인

데이터 저장을 대폭 감소할 것이다. 또한 Usechain 은 암호학 원리를 결합하여 데이터 저장 구조를 최적화 하고 데이터의 효율성과 안전성을 제고할 것이다.

라이트 노드 데이터의 검색과 검증:

1) 만약 라이트 노드가 한 계정의 상태(랜덤숫자, 잔액)를 획득하고 싶다면 결과가 검색될 때까지 State Root 에서 재귀적으로 트랜잭션 트리를 다운로드 할 수 있다;

2) 만약 라이트 노드에서 한 차례 거래가 확인된 적 있는지를 검증하려면 인접한 P2P 네트워크에서 해당 거래가 위치한 블록을 검색할 수 있고 State Root 의 트랜잭션 트리를 다운로드하여 계산해 낸 해당 거래의 해시와 트랜잭션 트리와 대응되는 해시로 Merkle Patricia Tree Root 를 얻을 수 있다. 그리고 블록 해더의 값과 비교하여 값이 동일하면 해당 거래가 확인된 적 있음을 검증할 수 있고 확인을 진행한 블록 숫자를 얻을 수 있다.

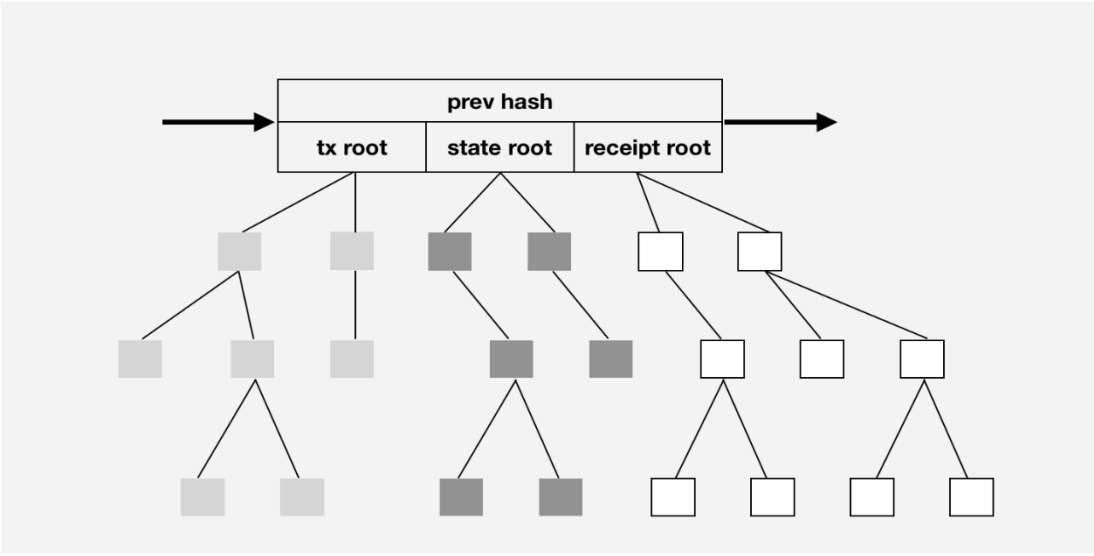


그림 8-1 Merkle Patricia Status Tree

9. 일반기능

9.1 투표 시스템

Usechain 은 투표 시스템을 구축하여 전체 커뮤니티가 투표를 통해 어떤 노드를 블록 생산 위원회로 선출할지, 어떤 특성을 실행해야 할지, 그리고 어떤 순서로 실행할지에 관해 합의를 달성할 수 있다. 모든 사용자는 자신의 수요에 따라 투표를 제안할 수 있다. 투표 제안자는 투표 내용과 투표 기한(블록 수량과 연관 있음)을 확정해야 하며 투표를 통해 모든 문제를 해결할 수 있다(예를 들면 새로운 아이콘 선택 문제). 새로운 특징의 추가는 주주들의 투표를 통해 허락 여부를 확인할 수 있고 특정 코인(특히 도둑이나 해커의 코인)을 파괴(동결)할지 여부도 결정할 수 있다. 또한 민주투표로 악성 공격을 가하는 노드를 정지시킬지 여부도 결정할 수 있는데 다시 말해 커뮤니티는 투표로 일부 사용자 혹은 노드에 관한 투표를 진행할 지 여부를 결정할 수 있다.

투표는 소지한 USI/USN 의 수량에 따라 계산하는데 신원 코인을 더 많이 소지하고 있는 사용자는 투표 시스템에서 더 큰 투표 능력을 가지게 된다(예를 들면 중앙화 거래소 혹은 마이너 풀 소유자). 이런 상황을 방지하기 위해 투표 능력에 상한 제한을 설정한다. 이런 설정은 익명의 블록체인에서 실현 불가능 하지만 Usechain 는 실명제의 특성이 있어 각 신원의 투표 상한을 제한할 수 있다.

Usechain 의 투표 시스템은 탈중앙화 화폐의 중요한 구성 부분 중 하나이며 지도자가 없고 집권 실체가 없이 모든 결정을 민주투표로 해결한다. 그리고 글로벌 문제를 해결하는 외에도 주주들은 투표 시스템으로 자산 거래 기능을 실현할 수 있어 주주를 도와 합의를 달성할 수 있다.

9.2 악성 주소의 발견과 징벌 매커니즘

Usechain 은 신원 퍼블릭 체인이며 온체인의 모든 주소는 신원 검증을 거친 주소이고 사용자는 무지식 증명 방법으로 블록체인에 제 3 측 신원 인증기관의 인증을 받았음을 증명할 수 있다. 온체인에서는 계정의 신원 정보를 확인할 수 없으며 일반적으로 사용자의 신원은 추적 불가능하다. 하지만 온체인에 악성행위(예를 들면 코인을 훔치거나 마이너의 악성 포크 행위 등)가 발생할 시에는 전문적인 심사 위원회에서 악성 주소를 추적하는데 관한 투표를 제안할 것이며 전반 네트워크 중 절반 이상의 사용자가 투표하여 허락한 뒤 해당 주소의 자산을 동결할 수 있다. 또한 심사 위원회는 악성 주소의 신원 검증 증명을 언록(unlock)하여 제 3 측 신원 검증 기관의 협조를 요청하고 악성 사용자의 진실한 신원 정보를 검색하여 해당 법률로 징벌을 내릴 것을 신청할 수 있다.

총 괄

가장 비전이 좋은 블록체인 생태 시스템인 Usechain 은 퍼블릭 블록체인과 프라이빗 블록체인의 장점을 결합하여 기존 블록체인 시스템의 단점을 보완하고 ID 애플리케이션 생태 시스템을 구축하는 것을 목표로 하고 있다. Usechain 은 기본 플랫폼의 부단한 개발과 제품개발, 상업화 프로젝트의 발전 및 교체를 통해 블록체인 ID 경제를 형성하고 업계 효율을 높여 사회의 협업 발전을 촉진할 것이다.

부록

1. 타원곡선 암호학

타원곡선 암호학은 미국 학자 Neil Koblitz 와 Victor Miller 가 1985 년에 각기 독자적으로 창립한 것이다[2]. RSA 와 ElGamal 암호기술은 사용길이가 1024 비트인 계수여야만 기본 안전등급에 도달할 수 있다. 하지만 타원곡선(ECC) 암호기술의 암호 공격 방법은 사용길이가 160 비트만 만족하면 같은 안전등급에 도달할 수 있다.

G 가 한 개의 유한체를 대표할 때 G 위에 타원곡선 E 를 정의하면 사실상 곡선 E 는 한 개 점집합을 대표한다. 그리고 E 와 무한 원점 O 로 이루어진 집합을 방정식이 결정한 타원곡선으로 볼 수 있다.

그러면:

$$E/G: \{(x,y)|y^2 + a_1xy + a_3y = x^3 + a_2x^2 + a_4x + a_6, a_1, a_2, a_3, a_4, a_6, x, y \in G\} \cup \{O\}$$

타원곡선 E 에서 덧셈 계산을 정의한다. $P(x_1, y_1), Q(x_2, y_2)$ 은 타원곡선 E 위에 있는 두 개의 점이며 이 두 점에 관한 덧셈 계산은 $P + Q$ 이고 R 은 (x_3, y_3) 이다. 여기서 R 은 지나가는 점 P 와 Q 의 직선과 E 곡선이 만나는 점이 X 축에 대하여 대칭되는 타원곡선 위의 점을 가리킨다.

P, Q, $-R$ 점이 공선이므로 공선 방정식은:

$$y = kx + b$$

그중 만약 $P \neq Q$(P, Q 두 점은 겹치지 않음)이면 직선 경사도는:

$$k = (y_1 - y_2)/(x_1 - x_2)$$

만약 $P = Q$,(P, Q 두 점은 겹침)이면 직선은 타원곡선의 접선이 되며 직선 경사도는:

$$k = (3x^2 + 2a_2x + a_4 - a_1y)/(2y + a_1x + a_3)$$

그리고:

$$(kx + b)^2 + a_1x(kx + b) + a_3(kx + b) = x^3 + a_2x^2 + a_4x + a_6$$

3 차항계수가 1 일 때; $-x_1x_2x_3$ 는 상수항계수이며 $x_1x_2 + x_1x_3 + x_2x_3$ 은 1 차항계수이고 $-(x_1 + x_2 + x_3)$ 는 2 차항계수이다

그러므로:

$$\begin{cases} x_3 = k^2 + ka_1 + a_2 + x_1 + x_2 \\ y_3 = y_1 - k(x_1 - x_3) \end{cases}$$

2. Secp256K1

Secp256k1 은 F_p유한체를 기반으로 한 타원곡선이며[1] 특별한 구조로 인해 최적화를 거친 후 기타 곡선보다 30%의 성능을 제고할 수 있고 아래와 같은 두 가지 장점이 있다:

1) 대역폭과 저장 자원을 적게 차지하고 키의 길이가 짧다;

2) 모든 사용자가 같은 방법으로 유한체계산을 진행할 수 있다.

비트코인과 이더리움은 secp256k1 기준으로 정의한 一갈래의 특별한 타원곡선을 사용하고 있다. 본 기준은 미국표준기술연구소(NIST)에서 설계한 것이다. secp256k1 곡선은 하기 함수로 정의하며 하나의 타원곡선을 그려낼 수 있다:

$$E: y^2 = x^3 + ax + b \ over \ F_p$$

그중 F_p은 소체이고 p는 소수이며 G는 기점이다. Secp256K1 의 파라미터 설정은 하기 참조:

p= FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFF FFFFFFFE FFFFFC2F

a = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000

b = 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000007

압축 형태의 기점 G는:

G = 02 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798

미압축 형태는:

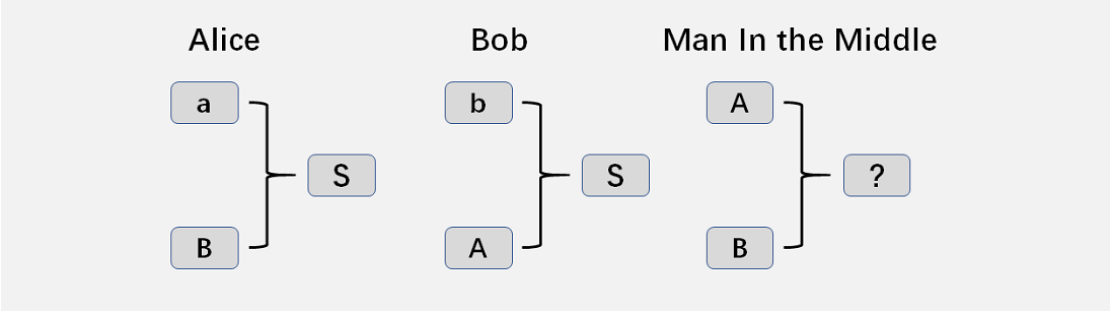
G= 04 79BE667E F9DCBBAC 55A06295 CE870B07 029BFCDB 2DCE28D9 59F2815B 16F81798 483ADA77 26A3C465 5DA4FBFC 0E1108A8 FD17B448 A6855419 9C47D08F FB10D4B8

3. 타원곡선 디피-헬만 키 교환 (Elliptic Curve Diffie–Hellman key Exchange , ECDH)

ECDH 는 익명으로 실행되는 Key-agreement protocol 이다. 이 프로토콜 아래 쌍방은 디피-헬만 키 교환 알고리즘을 통해 타원곡선으로 구축한 퍼블릭 키와 프라이빗 키에 암호를 걸고 안전성이 떨어진 통로에서 안전한 암호자료공유를 실현할 수 있다. 이것은 디피-헬만 키 교환의 변종이며 타원곡선 암호화를 통해 안전성을 강화할 수 있다.

만약 Alice 와 Bob 가 각기 퍼블릭/프라이빗 키(a, A), (b, B)를 생성하면 $A = [a]G$, $B = [b]G$, G는 동일한 타원곡선의 기점이다. Alice 와 Bob 가 각기 $S = [a]B$, $S = [b]A$를 계산하면:

$$S = [a]B = [a]([b]G) = [b]([a]G) = [b]A$$



Alice 와 Bob 는 쉽게 자신이 공유한 secret S를 계산해 낼 수 있지만 다른 사용자는 계산할 수 없다.

4. 링서명 알고리즘

링서명은 Rivest, shamir 와 Tauman 세 명의 암호학자가 2001 년 처음으로 제출한 알고리즘이며 간소화된 그룹 서명의 일종이다[15, 16, 19].

서명자는 먼저 임시 퍼블릭 키 집합을 선택하는데 집합에는 서명자 자신도 포함해야 한다. 서명자는 자신의 프라이빗 키와 서명 그룹 안의 기타 멤버의 퍼블릭 키로 다른 사용자의 협조 없이도 단독적인 서명을 생성할 수 있다. 그룹 안의 다른 사용자는 서명자가 동일한 키로 두 번째 서명을 생성할 때까지 서명자의 신원과 퍼블릭 키를 보아낼 수 없다. 또한 퍼블릭 키 그룹 안의 다른 멤버는 자신이 해당 그룹에 포함되어 있다는

사실을 모른다. 링서명은 네가지 부분으로 나눌 수 있다: GEN, SIG, VER, LNK:

1) GEN: 공용 파라미터(公共参数)를 채집하고 랜덤으로 $n-1$개의 퍼블릭 키를 선택해 동일사용자의 퍼블릭 키 P와 함께 퍼블릭 키 집합을 이룬다.

$$\{P_i | = 1,2,\dots,n\}$$

사용자의 퍼블릭/프라이빗 키(P,x)를 사용, $x \in [1, l-1]$, l은 점P의 계(阶)이며 퍼블릭 키 미러 I를 생성한다.

2) SIG: 필요한 서명 정보 m와 퍼블릭 집합에 관해:

$$\{P_i | = 1,2,\dots,n\}$$

링서명 $ringsig$을 계산해 낸다.

3) VER: 정보 m, 퍼블릭 키 집합 S와 서명 $ringsig$을 기반으로 서명의 합법성을 검증하고 True 혹은 False 를 출력한다.

4) LNK: 집합 $J = \{I_i\}$으로 서명이 사용된 적 있는지를 검증한다.

구체적인 과정은 하기 참조:

1) GEN: 서명자는 자신의 퍼블릭 키 x로 퍼블릭 키 $P = xG$와 퍼블릭 키 미러 $I = xH_p(P)$를 계산해 낸다. H_p는 해시 함수이고 무작위로 타원곡선 위의 점 한 개를 출력함과 동시에 블록체인에서 랜덤으로 퍼블릭 키 집합구성 $\{P_i | = 1,2,\dots,n\}$을 선택한다. $P_s = P$, s는 서명자 퍼블릭 키의 위치이다.

2) SIG: 서명자는 랜덤숫자를 선택한다.

$$\{q_i | = 0,1,\dots,n, q_i \in [1, l]\}$$

$$\{w_i | = 0,1,\dots,n, i \neq s, w_i \in [1, l]\}$$

아래와 같이 변화된다:

$$L_i = \begin{cases} q_i G, & i = s \\ q_i G + w_i P_i, & i \neq s \end{cases}$$

$$R_i = \begin{cases} q_i H_p(P_i), & i = s \\ q_i H_p(P_i) + w_i I_i, & i \neq s \end{cases}$$

다음 단계의 계산은:

$$c = H(m, L_1, \dots, L_n, R_1, \dots, R_n)$$

마지막으로 서명자 계산:

$$c_i = \begin{cases} w_i, & i \neq s \\ c - \sum_{k=0}^{n} c_i \ mod\ l, & i \neq s \end{cases}$$

$$r_i = \begin{cases} q_i, & i \neq s \\ q_s - c_s x \ mod\ l, & i \neq s \end{cases}$$

최종 서명은:

$$ringsig = (I, c_1, \dots, c_n, r_1, \dots, r_n)$$

3) VER: 검증자가 서명을 검증할 시에 정보 m, 공용 파라미터(公共参数)와 $S = \{P_i | = 1,2, \dots, n\}$, $ringsig$을 기반으로 계산하면:

$$\begin{cases} L'_i = q_i G + c_i P_i \\ R'_i = r_i H_p(P_i) + c_i I_i \end{cases}$$

그리고 하기 등식이 성립되는 지를 검증:

$$\sum_{k=0}^{n} c_i = H(m, L'_1, \dots, L'_n, R'_1, \dots, R'_n)$$

만약 성립되면 LNK 를 집행. 성립되지 아니 하면 검증자는 서명을 거절.

4) LNK: 블록체인에 출현했던 모든 I에게 집합 J를 구축한다. 만약 I가 집합에 포함될 때 해당 퍼블릭 키가 이미 사용되고 있음을 의미하며 서명 거래가 불법이라고 판단한다. 만약 집합에 포함되지 않을 때 서명 거래는 합법적이라 판단하고 I는 집합 J에 가입한다.

참고 문헌

[1] Wood G. Ethereum: A secure decentralised generalised transaction ledger[J]. Ethereum project yellow paper, 2014, 151: 1-32.

[2] Cooper D. Internet X. 509 public key infrastructure certificate and certificate revocation list (CRL) profile[J]. 2008.

[3] Vukolić M. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication[C]//International Workshop on Open Problems in Network Security. Springer, Cham, 2015: 112-125.

[4] Luu L, Narayanan V, Zheng C, et al. A secure sharding protocol for open blockchains[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016: 17-30.

[5] Schlosser M, Condie T, Kamvar S. Simulating a file-sharing p2p network[C]//1st Workshop on Semantics in Grid and P2P Networks. Stanford InfoLab, 2003.

[6] Li H, Lu R, Zhou L, et al. An efficient merkle-tree-based authentication scheme for smart grid[J]. IEEE Systems Journal, 2014, 8(2): 655-663.

[7] Vukolić M. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication[C]//International Workshop on Open Problems in Network Security. Springer, Cham, 2015: 112-125.

[8] Joux A. A one round protocol for tripartite Diffie–Hellman[C]//International algorithmic number theory symposium. Springer, Berlin, Heidelberg, 2000: 385-393.

[9] Rackoff C, Simon D R. Non-interactive zero-knowledge proof of knowledge and chosen ciphertext attack[C]//Annual International Cryptology Conference. Springer, Berlin, Heidelberg, 1991: 433-444.

[10] Gervais A, Karame G O, Wüst K, et al. On the security and performance of proof of work blockchains[C]//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. ACM, 2016: 3-16.

[11] Goldwasser S, Micali S, Rivest R L. A digital signature scheme secure against adaptive chosen-message attacks[J]. SIAM Journal on Computing, 1988, 17(2):

281-308.

[12] Modified Merkle Patricia Trie Specification. https://github.com/ethereum/wiki/wiki/Patricia-Tree.

[13] Johnson D, Menezes A, Vanstone S. The elliptic curve digital signature algorithm (ECDSA)[J]. International journal of information security, 2001, 1(1): 36-63.

[14] Koblitz N. Elliptic curve cryptosystems[J]. Mathematics of computation, 1987, 48(177): 203-209.

[15] Zhang F, Kim K. ID-based blind signature and ring signature from pairings[C]//International Conference on the Theory and Application of Cryptology and Information Security. Springer, Berlin, Heidelberg, 2002: 533-547.

[16] Liu J K, Wei V K, Wong D S. A separable threshold ring signature scheme[C]//International Conference on Information Security and Cryptology. Springer, Berlin, Heidelberg, 2003: 12-26.

[17] Certicom Research, Standards for efficient cryptography, SEC 1: Elliptic Curve Cryptography, Version 2.0, May 21, 2009.

[18] Rahat Afreen, S.C. Mehrotra , A Review on Elliptic Curve Cryptography for Embedded Systems : International Journal of Computer Science & Information Technology (IJCSIT), Vol 3, No 3, June 2011

[19] How to leak a secret, Ron Rivest, Adi Shamir, and Yael Tauman, ASIACRYPT 2001. Volume 2248 of Lecture Notes in Computer Science, pages 552–565.