SASEUL: Universal Computer

Jungwoo Lee

jjal@artifriends.com

https://artifriends.com

Abstract.

나카모토 사토시는 네트워크에 존재하는 데이터베이스 중 반드시 하나만을 선택할 수 있는 방법을 제시하였고, 이를 비트코인으로 증명했다. 그리고, 이는 이후 블록체인이라는 기술로 명명되었다.

이어서 비탈릭 부테린은 데이터가 절차로서 동작하는 스마트 컨트랙트의 개념을 블록체인에 최초로 도입하여 블록체인 데이터 그 자체가 곧 프로그램으로 동작할 수 있는 방법을 제시했고, 이를 이더리움으로 증명했다.

여러 대의 컴퓨터가 하나의 데이터, 코드, 정책을 공유하고 같은 의도로 동작하도록 할 수 있게 되었지만, 인류의 보편적 기술로 도입되기에는 성능적으로 해결해야할 이슈가 세 가지가 있다.

첫째로 신규 데이터가 업로드 되는 주기가 너무 길다는 것이며, 둘째로는 모두가 같은 데이터를 가지고 있어야만 하여 전체 데이터의 용량이 커지면 저장하기 어렵다는 것, 마지막으로 네트워크 참여 인센티브가 소수자에게 집중될 수밖에 없다는 점이다.

이 글에서는 상기 세 가지 문제를 해결하는 방법에 대해 다룬다.

1. Introduction

현대의 전산 시스템은 데이터베이스 관리자가 데이터를 언제든지 조작할 수 있다는 문제점이 있다. 나카모토 사토시는 이 문제의 해결을 위해 "신뢰할 수 있는 제 3 자" 없이 Peer-to-Peer 로만 동작하는 Electronic Cash System 인 Bitcoin 을 제안했다.

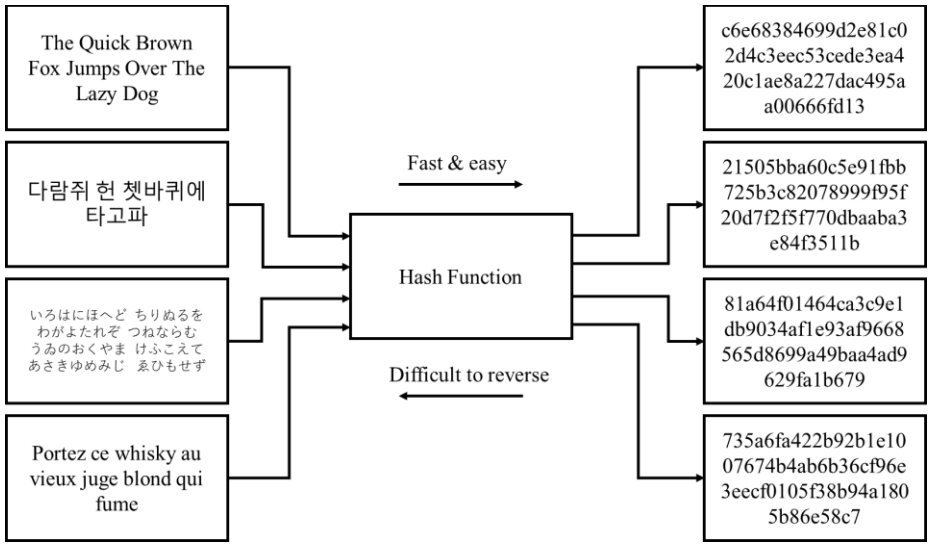
일반적으로 신뢰할 수 있는 제 3 자는 서비스 이용자들의 데이터베이스를 하나로 통일시켜주는 역할을 하며, 만약 신뢰할 수 있는 제 3 자가 없어 데이터베이스를 하나로 통일할 수 없다면 double-spending 문제가

발생할 수 있다. 나카모토 사토시는 이에 대한 해결책으로 “The network timestamps transactions by hashing them into an ongoing chain of hash-based proof-of-work, forming a record that cannot be changed without redoing the proof-of-work.” 라고 표현되는 블록체인과 작업증명의 개념을 제안했으며, 블록체인의 원리는 다음과 같다

1.1. Hash function

해시함수란 입력 값을 역변환할 수 없는 고정 길이의 값으로 변환해주는 비가역적 함수를 말한다. 결과값이 고정 길이이기 때문에 비둘기집의 원리에 의해 같은 해시 결과값을 가지는 원본 입력값이 두 개 이상 존재한다.

해시 결과값이 어떤 원본 입력값에서 연산되었다는 보증을 해주지는 않지만 높은 확률로 해당 원본 입력값으로부터 연산한 결과값이라는 단서를 제공한다. 또한 해시 결과값만 가지고 원본 입력값을 유추하기는 어렵지만 원본 입력값을 알게 되면 해시 결과값이 해당 원본 입력값에서 연산되었음을 검증할 수 있다.

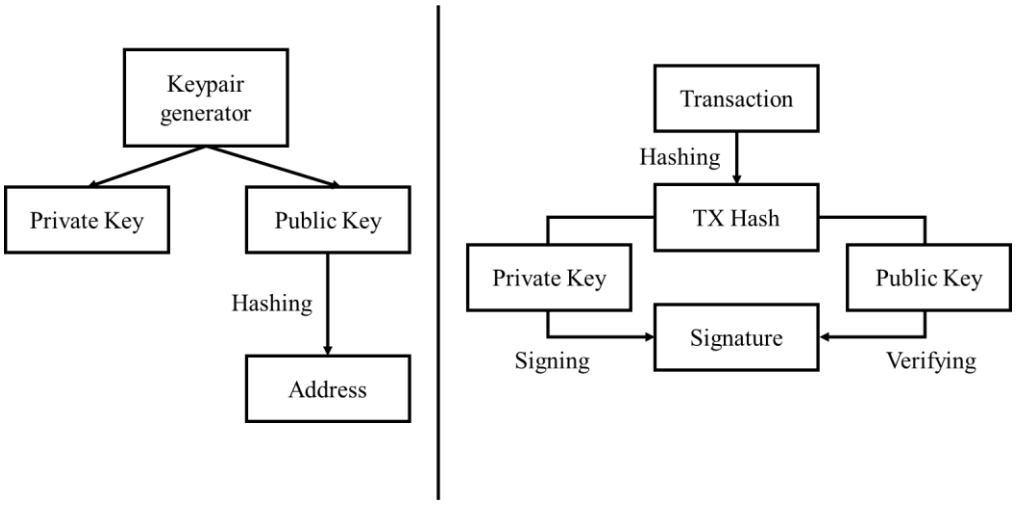


블록체인에서 해시함수는 데이터의 무결성을 검증하는 데에 이용된다. 블록체인 네트워크에서 모든 시스템은 같은 데이터를 보유하고 있어야 하며, 같은 데이터에 같은 트랜잭션을 가했을 때 같은 결과값이 도출되어야 한다. 또한 이 결과값은 다시 데이터베이스에 같은 값으로 저장되어야 하며, 이 일련의 절차가 잘 수행되었는지 확인할 수 있어야 한다. 이 때, 매 번 전체 데이터를 검증하는 것은 비효율적이므로 각

절차의 결과를 해시 결과값으로 변환하여 해시 결과값이 같은 지 비교하는 것으로 데이터의 무결성을 검증한다.

1.2. Public-key cryptography

Public-key cryptography 는 Asymmetric cryptography 라고도 불린다. 일반적으로 사용자는 키 생성 함수를 이용하여 공개 키와 비밀 키의 키 쌍을 생성할 수 있으며, 공개 키를 가지고 비밀 키를 유추할 수 없어야 한다. 이렇게 생성된 키 쌍은 알고리즘에 따라 1) 공개 키로 데이터를 암호화 하고, 비밀 키로 데이터를 복호화할 수 있는 체계를 제공하거나 2) 비밀 키로 데이터의 디지털 서명을 생성하고 공개 키로 디지털 서명의 유효성을 검사하는 체계를 제공한다.

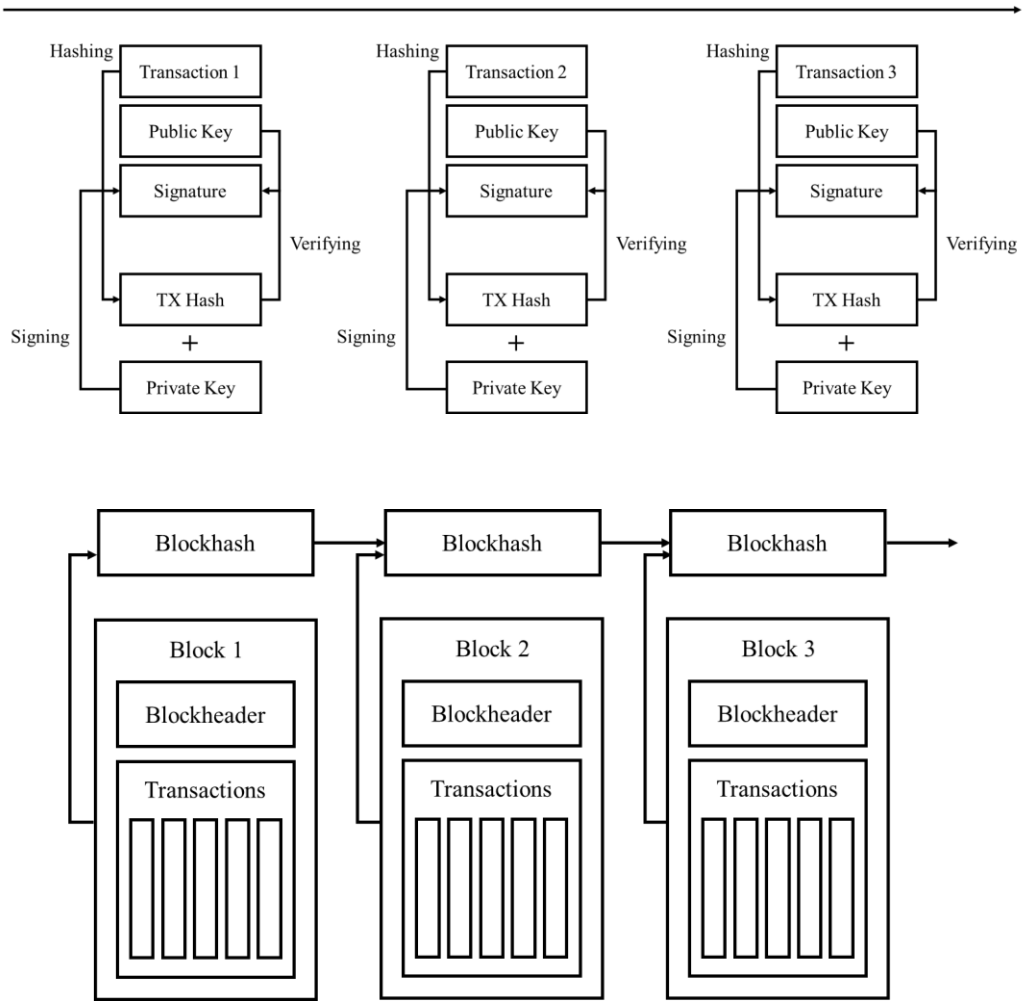


블록체인에서 Public-key cryptography 는 디지털 서명 체계를 이용하여 트랜잭션의 보유자를 식별하는 데에 이용된다. 일반적으로 공개 키를 해시화하여 “Address”로 불리는 식별자를 생성하고, Address 에 잔고 등의 Status 데이터를 귀속시킨다. 블록체인 네트워크의 이용자들은 디지털 서명을 제공하는 것으로 이용자 개개인이에 귀속된 Status 데이터를 변경하겠다는 의사표시인 트랜잭션을 등록할 수 있으며, 이 일련의 절차를 이용하여 송금, 판매, 구매 등의 개념을 블록체인 네트워크 상에 구현할 수 있다.

1.3. Blockchain

데이터를 묶어 블록을 형성하고, 해시 함수를 이용하여 블록이 이전 블록과 연결성을 가지도록 구현한 데이터 저장 체계를 블록체인이라고 하며, Public-key cryptography 등을 활용하여 여러 이용자가 하나의 블록체인 데이터를 공유하는 형태를 블록체인 네트워크라고 한다.

블록체인 네트워크에서 이용자들은 개개인이 지닌 키 쌍으로 서명된 트랜잭션을 블록에 포함시키고 데이터를 동기화하는 것으로 네트워크를 이용할 수 있다. 각 블록은 이전 블록과 연결성을 가지고 있기 때문에 특정 이용자가 블록체인의 중간 데이터를 수정하면 블록 데이터의 무결성이 파괴되어 올바른 블록체인 데이터로 인정받지 못한다.



각 블록은 이전 블록해시와 블록 데이터로 연산된 블록해시를 가지게 되며, 블록체인 네트워크의 이용자들은 모든 데이터를 비교하지 않아도 블록해시 값이 동일한 지 확인하는 것으로 데이터의 동기화가 잘 이루어졌음을 파악할 수 있다. 중간 블록 데이터를 변조하려면 모든 유효성 검사를 통과하면서 해시 충돌을 일으킬 수 있는 값을 찾아야 하며, 이는 현재 컴퓨팅 파워로는 거의 불가능한 일이다.

일반적으로 블록 데이터를 변조하는 공격은 중간부터 현재까지의 모든 블록 데이터를 새로 생성하여 덮어씌우는 것이 더 경제적다. 따라서 블록체인 네트워크는 단시간에 블록 데이터를 생성할 수 없도록

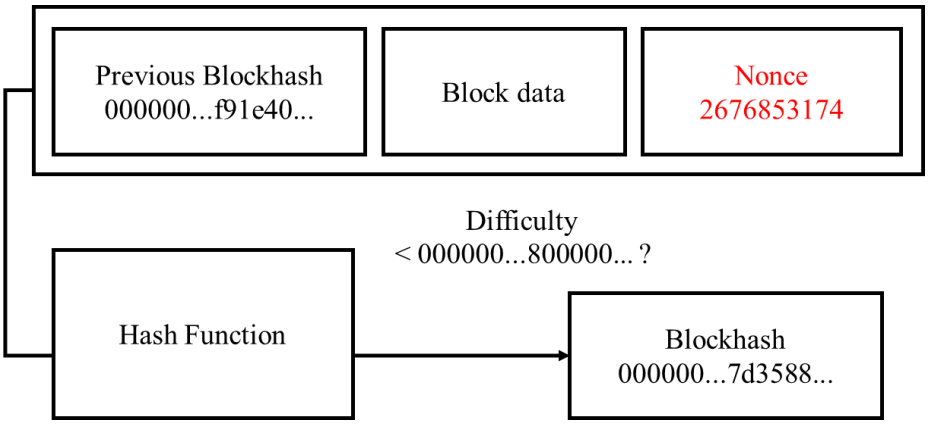
하는 장치를 가지고 있으며 대표적인 방법으로 Proof of Work 가 있다.

2. Proof of Work

만약 블록 데이터를 생성하기 위해 일정량의 컴퓨팅 파워를 사용해야 한다면, 데이터를 변조하려는 공격자는 블록 데이터 생성에 투입된 컴퓨팅 파워만큼을 매몰비용으로 투입해야만 블록체인 데이터를 변조할 수 있게 된다.

앞서 해시함수의 해시 결과값을 가지고 원본 입력값을 찾는 것이 어려움을 밝혔다. 블록을 생성하기 위한 조건으로 특정 해시 결과값이 나오는 원본 입력값을 찾아야하는 규칙이 있고 해시함수의 유효한 공격 방법이 없다면, 네트워크 참여자들은 무작위 대입을 통해 원본 입력값을 찾는 수밖에 없다.

네트워크 참여자들이 컴퓨팅 파워를 사용하여 새 블록 데이터를 위한 해시 결과값의 원본 입력값을 찾는 행위를 채굴이라고 부르며, 더 많은 컴퓨터가 채굴에 참여할수록 블록체인 네트워크의 데이터가 위변조될 가능성이 점점 줄어든다. 참여자들이 자발적으로 컴퓨팅 파워를 소모할 이유가 없기 때문에 충분한 인센티브를 제공하는 것이 일반적이다.



특정 해시 결과값이 나오는 정확한 원본 입력값을 찾는 것은 거의 불가능하기 때문에, 일정한 값보다 작은 해시 결과값이 나오도록 하는 원본 입력값을 찾으면 블록 데이터를 연결할 수 있도록 한다. 16 진수 해시에서 0 으로 시작하는 해시를 찾을 확률은 1/16 이며, 00 으로 시작하는 해시를 찾을 확률은 1/256, 000 ~ 008 로 시작하는 해시를 찾을 확률은 1/512 이다.

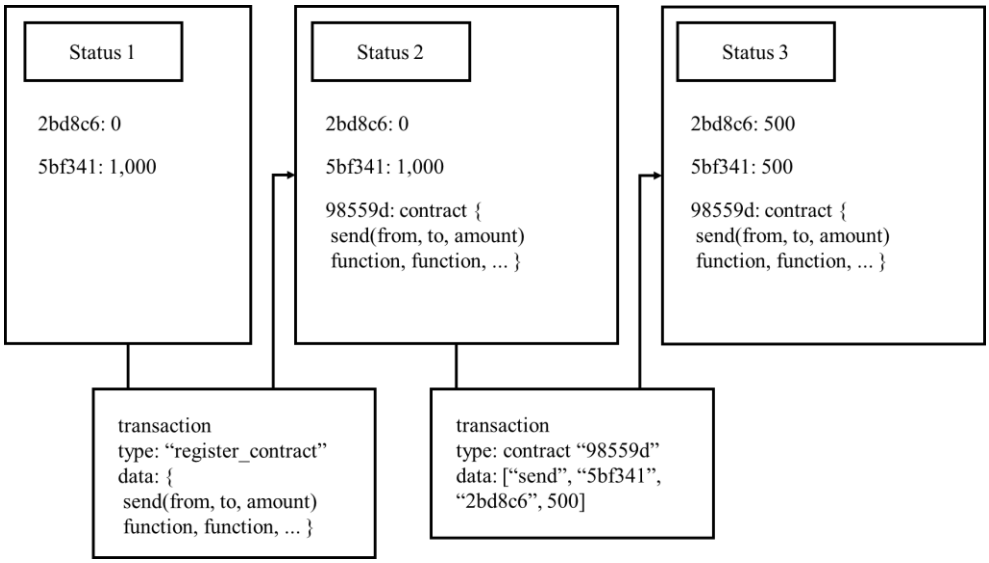
평균적으로 블록을 10 분에 1 개씩 찾도록 설계할 때, 이전 블록이 5 분만에 찾아졌다면 다음 블록은 해시를 찾을 확률이 1/2 배가 되도록 난이도를 설정하고, 이전 블록이 20 분만에 찾아졌다면 다음 블록은 해시를

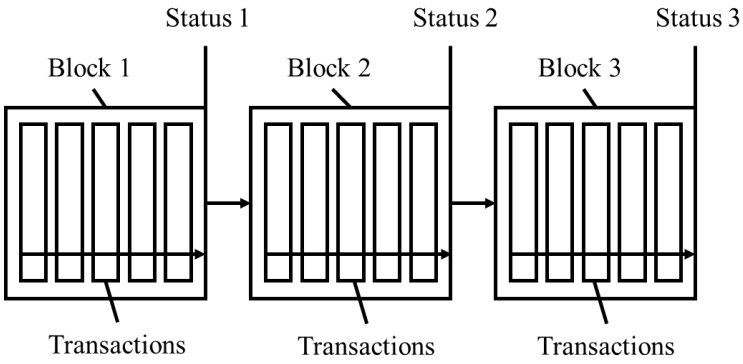
찾을 확률이 2 배가 되도록 난이도를 설정할 수 있다. 빈번하게 난이도를 변경하는 것은 네트워크의 안정성을 저해하므로 일반적으로 1 일에서 14 일 정도의 기간을 두고 주기적으로 난이도가 변경되도록 설계한다.

3. Smart Contract

나카모토 사토시는 블록체인과 작업증명의 결합으로 신뢰할 수 있는 제 3 자 없는 electronic cash system 을 온전히 구현할 수 있음을 입증하였다. 이에 비탈릭 부테린은 블록체인 기술을 화폐가 아닌 다른 분야에도 응용해보고자 하는 시도로, 블록 데이터 내에 실행 가능한 코드를 담는 "Smart Contract"의 개념을 제시하였다.

네트워크 이용자는 코드를 관리하는 코드를 호출하여 새로운 코드를 등록하거나 수정할 수 있으며, 이용자는 등록된 코드를 호출하여 자신 또는 타인이 배포한 새로운 코드를 이용할 수 있다. 이를 위해 블록체인에 저장되는 데이터의 종류를 Transaction 과 Status 로 분리하고, Transaction 을 순차적으로 연산하면 최종적으로 블록체인 네트워크가 가지고 있는 Status 를 얻을 수 있다.





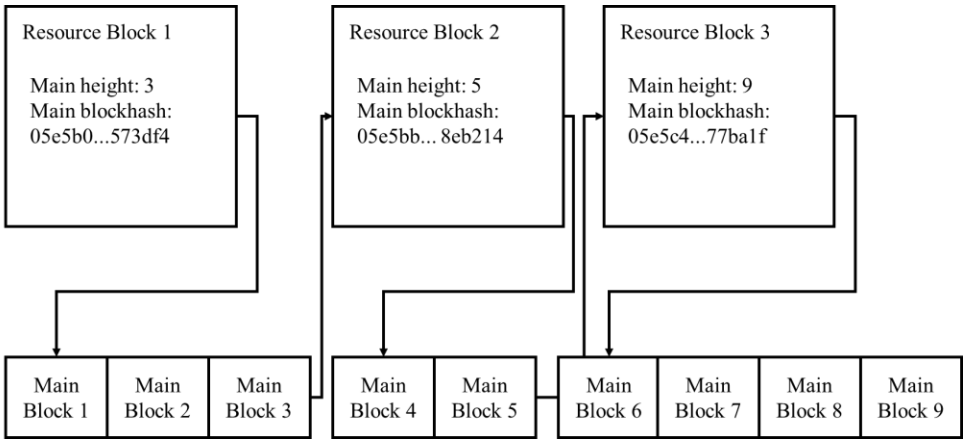
시스템은 Genesis 및 Register, Modify, Delete 등의 코드를 관리하는 컨트랙트, Virtual Machine 을 제공하고, 다른 일반적인 서비스 개념은 컨트랙트로 디자인하여 블록 데이터에 포함시킨다. Virtual Machine 은 코드 등록자와 이용자가 접근 및 조작할 수 있는 Status 데이터의 영역을 지정하여 잘못된 코드가 실행되지 않도록 하고, 코드 실행시간의 한계를 지정하여 코드가 무한히 실행되지 않도록 한다.

4. Validators

작업증명 기반의 블록체인 시스템은 트랜잭션 등록 완결성을 빠른 시간 내에 보장해줄 수 없다는 단점이 있다. 블록체인 연구자들은 이 단점을 탈피하기 위해 Validator 를 일정한 방식을 통해 지정하여 해결하고자 하였다. 그러나 온전히 Validator 에만 의존하는 방식은 블록 데이터 생성에 매몰비용이 없기 때문에 이중 서명을 통해 중간부터 현재까지의 모든 블록 데이터를 새로 생성하여 덮어씌울 수 있다는 문제가 있다.

4.1. Dual Chain

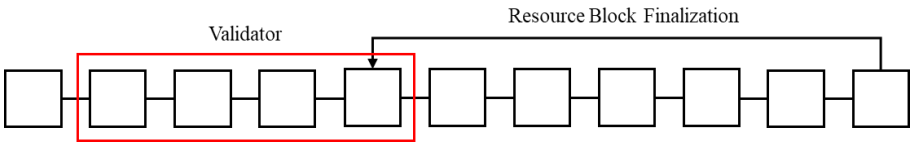
온전히 Validator 에만 의존하는 방식은 위험하기 때문에 작업증명의 방식과 결합하여 사용하여야 하며, 이 체계를 Dual Chain 이라고 한다. Dual Chain 은 작업 증명의 방식으로 Validator 를 선출하는 Resource Chain 과 Validator 에 의한 검증 방식으로 일반 트랜잭션을 처리하는 Main Chain 으로 구성된다.



채굴에 성공한 이용자는 약간의 인센티브를 얻고 다음 Validator 로 선출되며, 일반 트랜잭션을 온전히 처리하고 Main Chain 을 생성해야 더 큰 인센티브를 얻을 수 있게 된다. 네트워크 참여자들은 가장 긴 Resource Chain 을 기준으로 Resource Chain 의 동기화를 진행하며, Resource Chain 에 기록된 Main Chain 의 블록해시를 기준으로 Main Chain 의 동기화를 진행한다. Resource Chain 에 기록되지 않은 Main Chain 의 블록 데이터는 별도의 분기가 관측되지 않는 한 잠정적으로 완결되었다고 간주하며, 따라서 악의적인 이용자가 실시간으로 관측되지 않는 한 빠른 트랜잭션 등록 완결성을 보장할 수 있다.

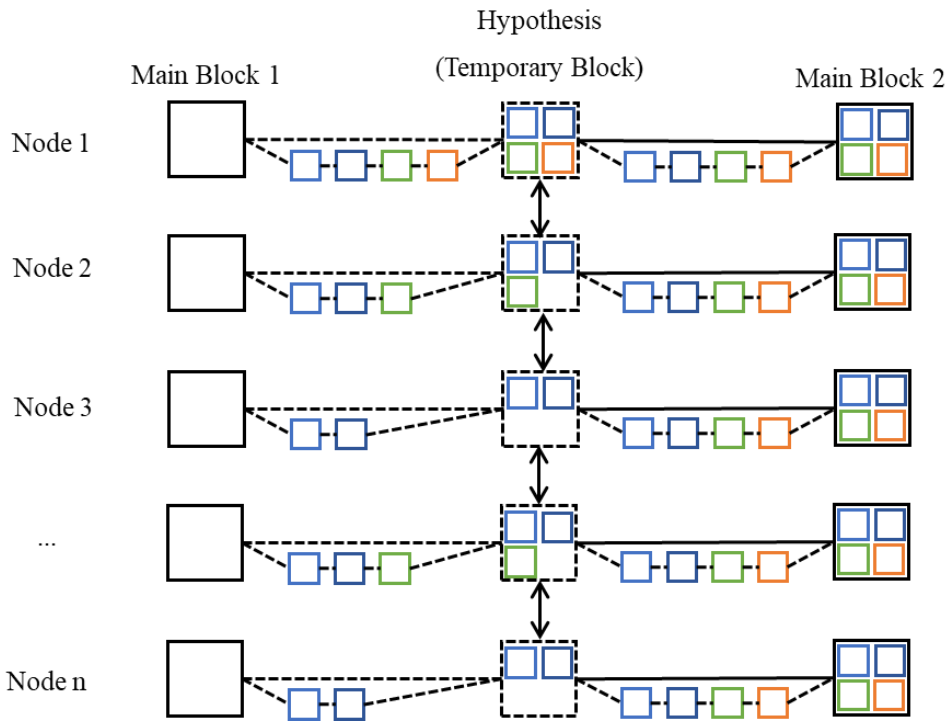
4.2. Hypothesis Acceptance Procedure

Dual Chain 의 도입으로 탈중앙화를 유지하며 트랜잭션의 고속처리를 보장할 수 있게 되었다. 그렇지만 채굴에 성공하여 Validator 로 선정된 참여자가 일시적으로 네트워크를 멈추거나 방해할 수 있다는 위험성을 가지고 있다.



이러한 상황을 방지하기 위하여, 채굴에 성공한 참여자는 다음 채굴자가 등장하더라도 바로 Validator 에서 제외되지 않고 일정 시간동안 Validator 의 역할을 수행하도록 하여 다수의 Validator 가 유지되도록 한다. 또한 선정된 다수의 Validator 는 Round-robin 방식이 아닌, 여러 Validator 가 동시에 블록 생성에 참여할 수 있는 알고리즘인 Hypothesis Acceptance Procedure 를 사용한다. 이 절차는 다음과 같다.

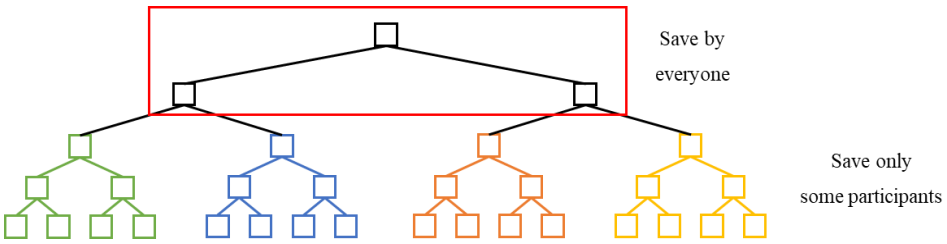
1) Validator 들은 생성할 블록의 헤더 정보를 자신의 서명을 담아 가설로 생성한다.
2) 모든 네트워크 참여자는 가설 데이터를 서로 동기화한다.
3) 서로 다른 가설 데이터가 존재할 경우, 일정한 규칙에 따라 Best 가설을 선택한다.
4) Best 가설이 아닌 가설을 제시한 Validator 는 Best 가설을 지지하는 서명을 담아 전파한다.
5) 동기화 결과 Validator 중 66% 이상이 동일한 가설을 지지하고 있다면, 블록을 확정된다.



Hypothesis Acceptance Procedure 방식의 가장 큰 장점은 가설의 서명만으로 블록을 확정하기 때문에 Validator 의 위치(IP)가 특정되지 않는다는 점이다. 이는 블록체인 네트워크가 DDoS 공격에 대한 강력한 저항을 가질 수 있게 도와준다.

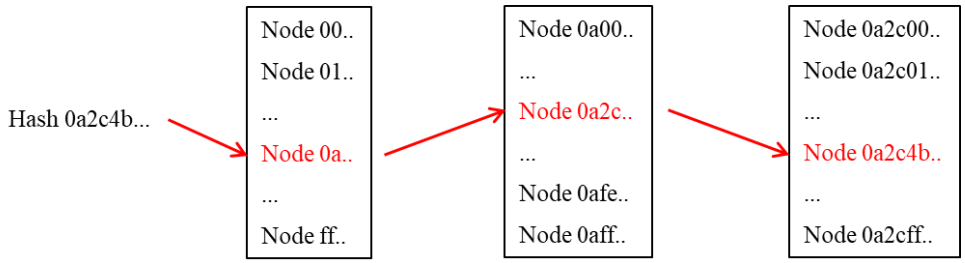
5. Partial Data Storage

트랜잭션을 고속으로 처리하게 되면 전체 데이터 용량의 증가 속도 또한 급격하게 상승하는 문제가 발생한다. 전통적인 블록체인은 무어의 법칙에 의해 하드웨어 성능이 증가할 것이기 때문에 큰 문제가 되지 않지만, SASEUL 과 같은 초고속 블록체인 네트워크에서는 전체 데이터의 용량이 단일 컴퓨터가 저장할 수 있는 한계를 넘어서게 된다.



네트워크 참여자들은 블록체인 데이터를 공통적으로 보관할 데이터와 개별적으로 보관할 데이터로 분리하여 자신의 Peer Address 에 따라 저장할 데이터만 저장하고, 나머지 데이터는 삭제한다. 일반적으로 Peer Address 는 랜덤하게 결정되므로 네트워크 참여자가 많아지면 균일하게 분산될 것이다. 또한 블록체인 네트워크가 온전하게 유지되길 원하는 참여자는 여러 대의 노드를 구축하여 전체 데이터를 보관하게 된다.

앞서 서술한 것과 같이 블록체인 데이터는 트랜잭션 데이터와 Status 데이터의 두 종류로 나뉜다. 트랜잭션 데이터는 블록 데이터에 포함되므로, 블록 데이터를 Merkle 트리로 구성하여 공통 보관 부분과 개별 보관 부분을 분류한다. Status 데이터는 Key-Value 형태의 Hash table 형태로 Merkle 트리를 구성하여 공통 보관 부분과 개별 보관 부분을 분류한다.



만약 데이터가 너무 많은 부분으로 쪼개졌다면, 데이터를 보관하는 피어를 한 번에 바로 찾을 수 없다. 이 경우, IP 주소를 찾는 원리와 비슷한 방식으로 데이터를 보관하고 있는 피어를 찾는다.

6. Universal Computing

6.1. Proof of Observation

Proof of Work 에서 Nonce 를 찾아내는 행위는 컴퓨팅 파워를 소모했다는 증명으로 볼 수 있다. 사슬에서는 데이터를 분할 저장함에 따라, 채굴에 참여하는 네트워크 참여자뿐 아니라 데이터를 많이 보관하는 네트워크 참여자 또한 중요한 참여자가 되었다. 따라서, 충분한 스토리지 용량을 제공하고 있는 참여자에게

인센티브를 제공해야 한다.

Validator 가 트랜잭션을 처리할 때, 전체 데이터를 보유하지 못한 상태라면 부분 데이터를 가진 네트워크 참여자의 데이터를 조회하여야 한다. 검증인이 Main Chain 을 생성하는 과정에서 주변 노드를 통해 상태와 블록 데이터의 유효성을 검사할 때, 조회 당하는 측에서는 검증인의 서명을 받게 되며, 이를 영수증이라고 한다. 영수증이 데이터를 유효하게 주었는지 아닌지를 검증할 수는 없지만 검증인이 데이터를 조회했다는 증거가 되며, 올바른 데이터를 보관하고 있는 피어는 영수증을 많이 모을 수 있게 된다.

영수증은 Resource Chain 에 기록되는 순간 인센티브로 전환되며, Resource Chain 의 블록을 생성한 다음 검증인은 과거 영수증을 채굴하는 동시에 영수증에서 전환되는 인센티브의 일부를 나눠받게 된다. 일반적으로 스토리지의 가격이 CPU 나 GPU 보다는 월등히 저렴하기 때문에 스토리지 용량을 늘리는 것이 더 효율적인 채굴 수단이 된다.

6.2. Multi-chaining

서로 다른 프로그램 또는 블록체인 네트워크의 데이터 무결성을 유지하며 실행하는 방법은 크게 어렵지 않다. 아래는 서로 다른 블록체인 간 Swap 을 구현하는 절차의 예시이다.

1) (Chain A) X 는 50A 를 Deposit 한다. Deposit 된 금액은 Y 가 특정 해시의 원본값을 입력하면 Approve 트랜잭션으로 가져갈 수 있지만, Y 는 해시의 결과값만 알 뿐 원본값을 알지 못한다. 이 Status 는 10 분 후부터 취소할 수 있다.
2) (Chain B) Y 는 100B 를 Deposit 한다. Deposit 된 금액은 X 가 1)에서 생성된 해시의 원본값을 입력하면 Approve 트랜잭션으로 가져갈 수 있다. Y 는 Chain A 에서의 해시 결과값을 알고 있기 때문에 "서로 다른 체인이지만 Chain A 의 원본값을 조건으로 하는 컨트랙트를 Chain B 에 설정할 수 있다." 이 Status 는 5 분 후부터 취소할 수 있다.
3) (Chain B) X 는 해시의 원본값을 입력하고 2)의 거래를 완결시킨다.
4) (Chain A) Y 는 2)의 거래가 완결되었으므로 1)에서 생성된 해시의 원본값을 알 수 있고, 1)의 거래를 완결시킬 수 있다.

영수증 시스템과 멀티체이닝 기법을 결합하여 데이터의 조회 뿐 아니라 프로그램의 실행에 대한 증명 또한 만들어낼 수 있으며, 블록에 컨트랙트 코드가 아닌 별도의 실행가능한 바이너리 코드를 담을 수 있다. 즉, 설계에 따라 모든 네트워크 참여자를 분산 컴퓨팅의 주체로 활용할 수 있으며 이를 Universal Computing 이라 한다.

7. Random Number

블록체인 네트워크에서는 모든 참여자가 동일한 결과를 연산해야하기 때문에 통상적인 난수 체계를 사용할 수 없다. 사슬 기반의 블록체인 네트워크에서는 컨트랙트 내에 "난수" 변수를 제공하고 있으며, 이 변수는 블록이 생성되는 순간 블록해시에 따라 값이 결정된다. 사슬 기반의 네트워크에서 Main Chain 의 블록해시 값은 예측이 거의 불가능하기 때문에 블록해시가 가지는 최대 연산한계 내에서 완전한 난수를 제공할 수 있다.

8. Conclusion

본 글에서 블록체인 네트워크가 빠른 속도로 블록을 처리하는 방법과 무한히 늘어나는 블록체인 데이터를 분산 저장하고 분산 처리할 수 있는 방법, 그리고 블록체인 네트워크를 유지하며 발생하는 인센티브를 간접 채굴의 방식을 통해 분산시킬 수 있는 방법을 제시하였다. 그러나 사슬이 제안한 방법들은 Universal Computer 구현의 시작점일 뿐이며 향후 더 연구할 여지가 남아 있다.

Reference

[1] Satoshi Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System" https://bitcoin.org/bitcoin.pdf

[2] Vitalik Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform"

https://ethereum.org/669c9e2e2027310b6b3cdce6e1c52962/Ethereum_Whitepaper_-_Buterin_2014.pdf

[3] Jae Kwon, "Tendermint: Consensus without Mining" https://tendermint.com/static/docs/tendermint.pdf