기술 백서

하모니 팀

버전 2.0

1. 서론

2008년에 비트코인 백서가 발간된 이래로 블록체인의 개념은 전 세계로 전파되었습니다. 분산화 된 화폐와 애플리케이션은 잘 알려진 개념이 되고 있기는 하지만 설계 상의 제약은 비트코인의 핵심적인 목표의 실현을 어렵게 하여 왔습니다. 최초의 비트코인 블록체인은 사람들이 은행이나 결제기관 같은 중간매개인이 없이 가치를 전송할 수 있도록 하여 주는 개인 대 개인의 지급결제 시스템으로 [13] 설계되었습니다. 그러나 비트코인이 인기를 얻게 됨에 따라, 초당 7건까지의 거래(TPS)를 처리할 수 있는 제한된 처리속도 때문에 그 성능 병목현상이 명백하게 드러났으며 지급결제 시스템으로서 그 비용이 엄청나게 높아지는 결과를 가져왔습니다.

2014년에 부테린과 그 외의 사람들은 (Buterin et. al.) [27] 이더리움이라고 불리는 새로운 블록체인 기반구조를 제안하였으며 이것은 개발자들이 "스마트 계약"을 사용함으로써 다양한 종류의 블록체인 애플리케이션을 개발할 수 있도록 하여 주었습니다. 그러나, 이더리움은 확장성 문제를 해결하지 못하였으며 초당 15건까지의 거래(TPS)를 처리할 수 있는 그 성능으로서는 게임이나 분산화 된 거래소와 같이 고속 처리성능을 필요로 하는 애플리케이션들을 지원할 수 없었습니다.

이더리움과 비트코인의 성능 상의 제약 때문에, 많은 수의 블록체인 프로젝트들이 거래의 처리속도를 개선하기 위한 다양한 솔루션들을 [3, 4, 5, 6, 7, 8, 9, 10, 24, 25] 제안하였습니다. 다양한 블록체인들은 [3, 4, 5, 6, 24, 25] 작업증명(PoW) 합의를 지분증명(PoS) 합의로 대체할 것을 제안하였습니다. EOS와 같은 다른 블록체인들은 위임 지분증명(DPoS)을 사용하며 여기에서 블록의 제안자는 체인 상의 알고리즘 프로세스에 의해서가 아니고 투표에 의하여 결정됩니다. IOTA와 같은 프로젝트들은 블록들의 체인으로 구현되는 자료구조를 방향성 비순환그래프(Directed Acyclic Graph 또는 DAG) 자료구조로 대체하였으며 이것은 거래의 순차적 처리의 제약을 극복합니다.

그러나 제안된 그러한 솔루션들은 보안 및 분산화와 같은 다른 중요한 측면들을 희생하지 않고는 유의적인 성능개선을 가져올 수 없습니다. 보안과 분산화를 모두 보호하는 확장성 솔루션은 샤딩(sharding)이며 이것은 다중의 검증자 그룹을 (즉, 샤드 – shards) 생성하고 그들이 거래들을 병행하여 처리하도록 합니다. 따라서, 전체적인 거래의 처리성능은 샤드의 수가 증가함에 따라 선형적으로 증가하게 됩니다. 질리카(Zilliqa)는 [12] 샤딩을 사용하여 확장성 문제를 해결하는 것을 제안한 최초의 공개적 블록체인입니다. 그러나 질리카(Zilliqa)의 샤딩 접근법은 두 가지 면에서 결함을 가지고 있습니다. 첫째, 그것은 블록체인 데이터의 저장소를 분리하지 않습니다 (상태 샤딩). 이것은 자원이 한정된 컴퓨터들이 네트워크에 참여할 수 없도록 하며 따라서 분산화의 정도를 감소시킵니다. 둘째, 질리카(Zilliqa)의 샤딩 프로세스는 난수생성 체계로서 작업증명(PoW)에 의존하기 때문에 단일샤드 탈취공격(single-shard takeover attack)에 취약성을 가집니다.

우리는 완전한 확장성을 가지며 안전성을 입증할 수 있고 에너지 효율적이며 샤딩에 근거한 차세대 블록체인인 하모니(Harmony)를 소개합니다. 하모니는 최적적으로 조율된 시스템 내에서 최선의 연구결과들과 기술적 방법들을 결합함으로써 기존 블록체인들의 문제점들을 해결합니다. 특히, 하모니는 다음과 같은 측면들에서 획기적인 개선을 이룩하였습니다.

- **완전한 확장성**: 하모니는 질리카와 같이 네트워크의 통신과 거래의 검증을 샤딩할 뿐 아니라 블록체인의 상태까지도 샤딩합니다. 이렇게 함으로써 하모니는 완전한 확장성을 가지는 블록체인이 될 수 있습니다.
- **안전한 샤딩**: 하모니의 샤딩 프로세스는 예측불가능하고 비편향적이며 검증가능하고 그리고 확장성을 가진, 분산화된 난수생성 (DRG) 프로세스 덕분에 안전하다는 것을 입증할 수 있습니다.

하모니는 또한 완만한 적응속도를 가진 능동적인 적(Byzantine adversary)들을 방어하기 위하여 시스템을 중단시키지 않는 방법으로 네트워크를 재샤딩(resharding)합니다.

- **능률적이고 신속한 합의**: 검증자를 선택하기 위해서 작업증명(PoW)을 필요로 하는 샤딩(sharding)에 근거한 다른 블록체인들과 달리, 하모니는 지분증명(PoS)에 근거하고 있으며 따라서 에너지 효율적입니다. 합의는 PBFT보다 100배 빠르며 선형적으로 확장가능한 BFT 알고리즘을 사용하여 도달됩니다.
- **적응적인 최소기준 지분증명(PoS):** 각 노드가 네트워크에 참여하기 위하여 요구되는 최소한의 지분은 악의의 지분소유자가 단일의 샤드에서 그들의 권한을 집중시키는 것을 방지할 수 있는 방법으로 전체 지분의 규모에 근거하여 적응적으로 조정됩니다. 더욱이, 최소한의 요구 지분은 소규모의 지분 소유자도 네트워크에 참여하여 보상을 받을 수 있도록 하기 위하여 충분히 낮습니다.
- **확장가능한 네트워크 기반구조**: 랩터큐 파운틴코드(RaptorQ fountain code)와 함께 적응적 정보분산 알고리즘(Adaptive Information Dispersal Algorithm)을 사용함으로써 하모니는 샤드의 내부에서 그리고 네트워크의 전체에 걸쳐 블록들을 신속하게 유포시킬 수 있습니다. 하모니는 또한 샤드의 수가 증가함에 따라 대수적으로 (logarithmically) 확장되는, 샤드들 사이의 교차적 거래를 가능하게 하기 위하여 카뎀리아 경로설정 방법(Kademlia routing)을 [37] 채택합니다.
- **일관적인 샤드들 사이의 교차적 거래**: 하모니는 각 샤드들이 서로 직접적으로 교신을 할 수 있도록 함으로써 샤드들 사이의 교차적 거래를 지원합니다. 샤드들 간의 교차적 거래의 일관성을 유지하기 위하여 최소단위 잠금처리 체계(atomic locking mechanism)가 사용됩니다.

프로토콜과 네트워크 수준 모두에서 혁신을 이룸으로써 하모니는 새롭게 대두되는 분산화 경제를 지원할 수 있는 확장가능하고 안전한 블록체인 시스템을 선보입니다. 하모니는 대용량의 분산화 된 거래소와 상호반응적인 공정한 게임, 비자(visa) 급의 지급결제 시스템 그리고 사물인터넷 거래를 포함하여 이전에는 블록체인에서 불가능하였던 애플리케이션들을 가능하게 할 것입니다. 하모니는 수십억의 사람들에 대하여 신뢰를 구축하고 근본적으로 공정한 경제체제를 만들기 위하여 노력하고 있습니다.

2. 합의 체계

합의 프로토콜은 모든 블록체인에서 중요한 구성요소입니다. 그것은 다음 번의 블록에서 블록체인 검증자들[1]이 얼마나 안전하고 신속하게 합의에 도달하는 지를 결정합니다. 비트코인을 구동하는 최초의 블록체인 합의 프로토콜은 작업증명(PoW) 합의 프로토콜입니다. 작업증명(PoW)은 채굴자들이 암호적 수수께끼에 대한 답을 찾기 위하여 경쟁하는 프로세스이며 – 승자는 다음 번의 블록을 제안하고 약간의 토큰 보상을 받을 수 있는 권리를 획득합니다. 작업증명(PoW)의 보안 상의 가정은 해싱(hashing) 처리능력의 50% 이상이 정직한 노드들에 의하여 통제된다고 하는 것입니다. 그러한 가정을 근거로 하여 합의의 규칙은 가장 긴 체인이 정본이라고 간주하는 것이며 따라서 작업증명(PoW) 합의는 또한 체인에 근거한 합의라고도 합니다.

학계에서 20년 이상 연구되어 온 또 다른 유형의 합의 프로토콜은 PBFT (Practical Byzantine Fault Tolerance)라고 [14] 하는 것입니다. PBFT에서 하나의 노드가 "리더"로 선택되며 나머지의 노드들은 "검증자"가 됩니다. PBFT 합의의 각 회전은 준비단계와 확정단계 두 개의 주요 단계를 포함합니다. 준비단계에서 리더는 자신의 제안을 모든 검증자들에게 배포하고 이들은 차례로 해당 제안에 대한 자신의 투표를 나머지의 모든 검증자들에게 배포합니다. 모든 검증자들에게 투표를 재배포하는 이유는 각 검증자의 투표가 다른 모든 검증자들에 의하여 집계되어야 하기 때문입니다. 준비단계는 f가 악의의 검증자들의 수이고 모든 검증자들과 리더의 총수가 3f + 1일 때, 2f + 1개의 일관적인 득표가 얻어질 때 종료됩니다.

[1] 거래들을 검증하고 합의를 위한 과정을 수행함으로써 블록체인 네트워크를 지원하는 컴퓨터 장치들.

확정단계는 그와 유사한 득표 집계과정을 포함하며 2f + 1개의 일관적인 득표가 있을 때 합의가 도달됩니다. 검증자들 사이에서의 투표의 재배포 때문에 PBFT는 $O(N^2)$의 통신 복잡도를 가지며 이것은 수백 또는 수천의 노드들을 가진 블록체인 시스템에서 확장성을 허용할 수 없습니다.

PBFT를 개선하는 것으로서 [14] 하모니의 합의 프로토콜은 통신 복잡도의 면에서 선형적으로 확장이 가능하며 따라서 우리는 그것을 FBFT(Fast Byzantine Fault Tolerance)라고 합니다. FBFT에서는 모든 검증자들이 그들의 투표를 배포하도록 요구하는 대신에 리더는 하나의 $O(1)$ 크기의 다중서명 파일에 검증자들의 투표를 취합하기 위한 다중서명 절차를 진행하고 나서 그 결과를 배포합니다. 따라서 각 검증자는 $O(N)$개의 서명을 받는 대신에 단 하나의 다중서명을 받으며 따라서 통신 복잡도는 $O(N^2)$에서 $O(N)$으로 감소됩니다.

하나의 $O(1)$ 크기 다중서명을 사용하는 아이디어는 일정한 크기의 다중서명 집합체를 위하여 Schnorr 서명체계를 사용하며 메시지의 전달을 쉽게 하기 위하여 검증자들 간에 다중전송 방식의 트리를 구성하는 ByzCoin의 BFT로부터 [15] 영감을 얻었습니다. 그러나, Schnorr 다중서명은 비밀의 확정절차를 필요로 하며 이것은 하나의 다중서명을 위해 총 두 개의 왕복 교신을 필요로 합니다. 하모니는 BLS (Boneh–Lynn–Shacham) 다중서명을 [28] 사용함으로써 그것을 개선하였으며 이것은 하나의 왕복 교신만을 필요로 합니다. 따라서 FBFT는 ByzCoin의 BFT와 비교하여 최소한 50% 더 빠릅니다. 그 외에도 하모니는 블록의 배포 프로세스를 (§6.2단원에서 설명됨) 가속화하기 위하여 랩터큐 파운틴코드(RaptorQ fountain code)를 사용합니다. 파운틴코드의 배포기술은 또한 ByzCoin의 원래의 트리에 근거한 다중배포 설계의 보안문제를 회피할 수 있도록 하여 줍니다 [16, 17].

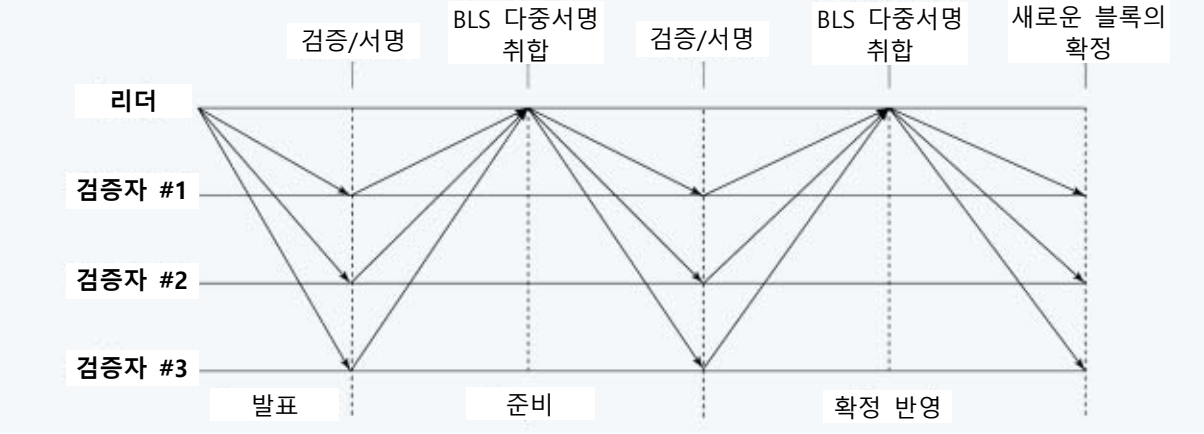

그림 1. 하나의 합의 회전에서의 네트워크의 통신

특히, 하모니의 FBFT 합의에는 다음과 같은 단계들이 포함됩니다:

1. 리더가 새로운 블록을 생성하고 해당 블록의 헤더(header)를 모든 검증자들에게 배포한다. 그리고 리더는 블록의 내용을 삭제 코드와 함께 (내용이 §6.2 단원에서 설명됨) 배포한다. 이것은 "발표" 단계라고 합니다.
2. 검증자들은 블록 헤더(header)의 유효성을 확인하고 블록 헤더에 BLS 서명을 추가한 다음 해당 서명을 리더에게 반송한다.
3. 리더는 검증자들로부터 (리더 자신을 포함하여) 최소한 2f + 1개의 유효한 서명이 도착할 때까지 기다린 다음 그들을 BLS 다중서명으로 취합한다. 그 후에 리더는 어떤 검증자들이 서명을 하였는지를 나타내는 비트맵 정보와 함께 취합된 다중 서명을 배포한다. 제2단계와 함께 이 과정은 PBFT의 "준비' 단계를 완성한다.
4. 검증자들은 다중서명에 최소한 2f + 1명의 서명자들이 포함되었는지 확인하고 제1단계에서 리더에 의해 배포된 블록 내용의 거래를 검증하고 제3단계에서 받은 메시지에 서명을 하고 그것을 다시 리더에게 반송한다.

5. 리더는 제4단계로부터 최소한 2f + 1개의 유효한 서명이 (제3단계와 서명자들이 다를 수 있음) 도착할 때까지 기다린 다음 그들을 BLS 다중서명으로 취합하고 모든 서명자들을 기록하는 비트맵 정보를 생성한다. 최종적으로 리더는 모든 다중서명들과 비트맵 정보를 첨부하여 새로운 블록을 확정한 뒤 모든 검증자들이 확정 절차를 수행할 수 있도록 새로운 블록을 배포한다. 제4단계와 함께 이것은 PBFT의 '확정' 단계를 완성한다.

하모니의 합의에서 검증자들은 지분증명에 근거하여 선택됩니다. 따라서, 하나의 서명에 대해서 하나의 투표권이 대응되는 것이 아니고 더 많은 투표권 지분을 가지고 있는 검증자는 다른 검증자들과 비교하여 더 많은 투표권을 가진다는 점에서 실제의 프로토콜은 위에서 기술된 것과 약간의 차이가 있습니다. 따라서 검증자들로부터 최소한 2f + 1개의 서명이 도착하기를 기다리는 대신에, 리더는 합산하여 최소한 2f + 1개의 투표권을 가지는 검증자들의 서명이 도착하기를 기다립니다. 지분증명 선택 체계의 내용은 §3.3 단원에서 설명될 것입니다.

3. 샤딩 (Sharding)

확장성을 위한 솔루션으로서 블록체인 샤딩은 2017년 이래로 많은 관심을 받아왔습니다. 업계와 학계 모두에서 다양한 샤딩 솔루션들이 제안되어 왔습니다.

업계에서는, 질리카(Zilliqa)가 [12] 2,800 TPS의 처리성능을 표명하는, 샤딩에 근거한 최초의 공개적인 블록체인이었습니다. 질리카는 신분의 등록 프로세스로서 (즉, 시빌 공격의 (Sybil attack) [1] 방지) 작업증명(PoW)을 사용합니다. 질리카의 네트워크는 하나의 디렉토리 서비스 위원회(directory-service committee)와 다중의 샤드 위원회(shard committee)를 (즉, 네트워크 샤딩) 포함하며 각각은 수백 개의 노드들을 포함합니다. 거래는 서로 다른 샤드들에 배정되며 각각 별도로 처리됩니다 (즉, 거래 샤딩). 모든 샤드 들로부터 처리결과로 생성된 블록들은 디렉토리 서비스 위원회(directory-service committee)에 의하여 취합되고 병합됩니다. 거래를 처리하기 위하여 각 노드는 전체 블록체인의 상태정보를 보유하여야 하기 때문에 질리카는 상태 샤딩 솔루션(state sharding solution)은 아닙니다.

학계에서는, 옴니레저(Omniledger)와 [8] 래피드체인(RapidChain) [7] 같은 프로젝트가 각 샤드가 블록체인 상태의 일 부분만을 보유하는 상태 샤딩의 기능을 포함하는 솔루션을 제안하였습니다. 옴니레저는 안전한 난수를 생성하기 위하여 랜드하운드(RandHound)라고 [25] 하는 다중참여자 연산체계를 사용하며 해당 난수는 노드들을 무작위적으로 샤드에 배정하기 위하여 사용됩니다. 옴니레저는 공격자들이 시간에 걸쳐서 한 샤드의 노드들을 점점 더 많이 부정 변조시킬 수 있는 서행의 적응적 부정변조 모델(slowly adaptive corruption model)을 가정합니다. 그러한 보안모델 하에서 궁극적으로 하나의 샤드가 부정 변조될 수 있습니다. 옴니레저는 샤드들 내에 포함된 모든 노드들을 에포크(epoch)라고 하는 일정 시간을 간격으로 하여 이동 및 재편함으로써 샤드들의 부정변조를 방지합니다. 래피드체인은 옴니레저를 기반으로 하여 구현되며 시스템의 중단 없이 노드들을 이동 및 재편성하기 위하여 제한된 쿠쿠 규칙(bounded cuckoo rule)의 사용을 제안합니다 [19].

하모니는 이들 세 개의 이전의 솔루션들로부터 [7, 8, 12] 영감을 얻었으며 선형적으로 확장가능하고 안전하다는 것이 입증될 수 있으며 지분증명(PoS)에 근거한 완전한 샤딩 체계를 설계합니다. 하모니는 하나의 신호 체인(beacon chain)과 다중의 샤드 체인(shard chain)들을 포함합니다. 샤드 체인들은 별개의 블록체인 상태들을 저장하고 병렬적으로 거래를 처리하는 반면에 신호 체인은 무작위의 신호장치로서 그리고 신분의 등록부로서의 역할을 담당합니다. 하모니는 검증가능한 랜덤함수(VRF)와 검증가능한 지연함수(VDF)를 결합함으로써 난수 생성을 위한 효율적인 알고리즘을 제안합니다. 하모니는 또한 샤딩 프로세스에 지분증명(PoS)를 포함시키며 이것은 샤딩의 보안을 위한 고려사항이 노드들의 최소한의 수로부터 [7, 8, 12] 최소한의 투표권의 수로 변화되는 결과를 가져옵니다.

3.1 분산된 난수 생성

배경
무작위에 근거한 샤딩과 [7, 8] 위치에 근거한 샤딩 [34] 그리고 중앙에 의해 통제되는 샤딩과 [35] 같이, 노드들을 샤드로 배정하기 위하여 여러 가지 방법들이 제안되었습니다. 모든 방법들 중에서 무작위에 근거한 샤딩이 가장 안전한 솔루션으로 인식되었습니다. 무작위에 근거한 샤딩에서 각 노드의 샤딩에 대한 배정을 결정하기 위하여 상호 간에 합의된 난수가 사용됩니다. 이러한 난수는 다음과 같은 성질을 충족시켜야 합니다.

1. 예측 불가능성: 난수가 생성되기 전에는 어느 누구도 그것을 예측할 수 없어야 한다.
2. 비편향성: 난수를 생성하기 위한 프로세스는 어떠한 참여자에 의하여도 편향(bias)이 발생하지 않아야 한다.
3. 검증가능성: 생성된 난수의 유효성은 어떠한 관찰자에 의하여도 검증이 가능하여야 한다.
4. 확장성: 난수의 생성을 위한 알고리즘은 많은 참여자들의 수에 대하여 확장이 가능하여야 한다.

옴니레저는 [8] 랜드하운드(RandHound) [25] 프로토콜을 사용하며 이것은 PVSS와 (공개적으로 검증가능한 비밀 분산, Publicly Verifiable Secret Sharing) 비잔틴 합의(Byzantine Agreement)를 수반하며 리더에 의해 주도되는 분산된 난수발생 (DRG) 프로세스입니다. 랜드하운드는 참여 노드들을 c 규모의 다중 그룹으로 분할하는 하나의 $O(n * c^2)$ 프로토콜입니다. 이것은 위의 세 가지 성질들을 충족시키지만 확장성이 있다고 하기에는 실행이 어려울 정도로 느립니다.

래피드체인은 [7] 각 참여자들이 VSS를 (검증가능한 비밀분산, Verifiable Secret Sharing) [22] 수행하도록 한 뒤, 결합된 비밀의 분산 값들을 난수 결과값으로 사용함으로써 더 간단한 방법을 채택합니다. 불행하게도, 악의의 노드들이 비일관적인 분산 값들을 서로 다른 노드들에게 보낼 수 있기 때문에 이 프로토콜은 안전하지 않습니다 [25]. 또한 래피드체인은 재조합된 난수의 가능한 여러 가지 버전들에 대해 노드들이 어떻게 합의에 도달하는 지를 설명하고 있지 않습니다.

더욱이, 알고랜드(Algorand)는 [18] 합의 검증자들의 그룹을 선택하기 위하여 VRF에 (검증가능한 랜덤함수, Verifiable Random Function) 근거한 암호적인 추첨에 의존합니다. 이더리움 2.0 버전의 설계는 최종 공개자 공격을 (last-revealer attack) 방지하기 위하여 실제 난수의 공개를 지연시키는 VDF의 (검증가능한 지연함수, Verifiable Delay Function) [20] 사용을 제안합니다 [36]. VDF는 새로 고안된 원시적인 암호 알고리즘이며 이것은 계산을 위하여 조절가능한 최소한의 시간을 필요로 하며 그 결과는 즉시 검증될 수 있습니다.

VRF 및 VDF를 사용하는 확장가능한 난수의 생성

하모니의 접근방법은 위의 솔루션들의 장점들을 결합하는 것입니다. 먼저, 하모니의 DRG (distributed randomness generation) 프로토콜의 복잡도는 $O(n)$이며 이것은 실제적으로 최소한 랜드하운드보다는 훨씬 빠릅니다. 둘째, 래피드체인의 VSS에 (검증가능한 비밀분산, Verifiable Secret Sharing) 근거한 단순한 방법과 달리 우리의 방법은 비편향적이며 검증가능합니다. 셋째, 이더리움 2.0 버전의 솔루션과 비교하여 우리의 방법은 난수가 최종적인 것이 되도록 하기 위하여 BFT 합의를 사용합니다. 세부적으로 이 프로토콜은 다음과 같은 단계들을 포함합니다.

1. 리더는 최종 블록의 해시값 $H(B_{n-1})$과 함께 init 메시지를 모든 검증자들에게 전송한다.
2. init 메시지를 받은 후에 각 검증자 i에 대하여 난수 r_i와 증명값 p_i를 생성하기 위하여 VRF가 산출되며: $(r_i, p_i) = VRF(sk_i, H(B_{n-1}), v)$, 여기에서 sk_i는 검증자 i의 비밀 키이고 v는 현재의 합의 참조 수이다. 그 뒤에 각 검증자는 (r_i, p_i)를 리더에게 반송한다.

3. 리더는 최소한 f + 1개의 유효한 난수가 도착할 때까지 기다린 뒤, 최종 난수 pRnd의 원상(preimage)을 얻기 위하여 XOR 연산으로 그들을 결합한다.
4. 리더는 pRnd에 관한 합의에 도달하고 그것을 블록 B_n에 확정 반영하기 위하여 모든 검증자들 사이에서 BFT를 (§2 단원에서 설명됨) 수행한다.
5. pRnd가 확정 반영된 뒤에, 리더는 실제의 난수 *Rnd = VDF (pRnd, T)*의 계산을 시작하며 여기에서 T는 VDF의 난이도이며 난수가 k개의 블록 이후에만 계산될 수 있도록 알고리즘적으로 설정된다.
6. Rnd가 계산된 뒤에, Rnd의 유효성에 관해 합의를 이루고 최종적으로 해당 난수를 블록체인에 확정 반영하기 위하여 리더는 모든 검증자들 사이에서 BFT 과정을 시작한다.

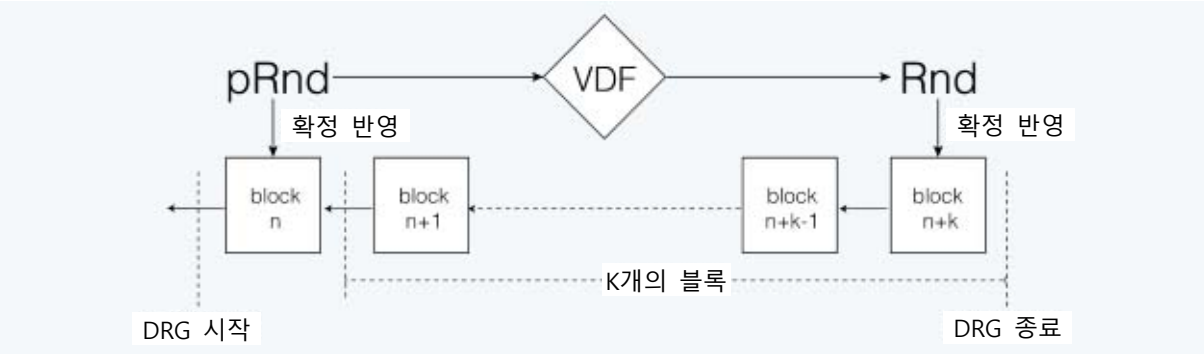

그림 2. VDF는 (Verifiable Delay Function) 최종 난수의 공개를 지연시킵니다.

VDF는 Rnd의 공개를 입증 가능하게 지연시키고 기호에 따라 VRF 난수의 부분적인 하위 집단을 선택함으로써 악의의 리더가 난수를 편향되게 하는 것을 방지하기 위하여 사용됩니다. VDF 때문에, pRnd가 블록체인에 확정 반영되기 전에는 리더가 실제의 최종 난수를 알 수 없습니다. VDF에 의해 Rnd가 계산될 때까지, 리더가 그것을 더 이상 변조할 수 없도록 pRnd는 이미 이전의 블록에 확정 반영됩니다. 따라서 악의의 리더가 할 수 있는 최선의 방법은 난수 pRnd를 무작정 확정 반영하든가 아니면 pRnd를 확정 반영하지 않음으로써 프로토콜을 중단시키는 것입니다. 전자는 정직한 행동과 같은 결과를 가져옵니다. 후자의 경우는 리더를 교체하고 프로토콜을 다시 시작하기 위하여 PBFT에서와 [14] 같은 제한시간 (timeout) 체계가 적용될 것이기 때문에 커다란 손상을 초래하지 않을 것입니다.

장기적으로 우리는 VDF의 계산을 위해 ASIC의 존재를 가정하며 여기에서 ASIC(애플리케이션 전용 집적회로, Application-Specific Integrated Circuit)을 구동시키는 소수의 이타적인 노드들이 결과를 발표할 것이며 누구도 시스템에 관해 내기를 할 수 없을 것입니다. VDF ASIC이 실제로 활용되기 전에는 더 빠른 컴퓨터 장비를 가진 공격자가 다른 정직한 노드들보다 먼저 결과 값을 계산하는 것이 가능합니다. 이러한 상황이 발생될 때까지는 공격자는 정직한 노드들보다 약간 앞서서만 난수 값을 알 수 있습니다. 원칙적으로는 공격자가 이에 따라 이익을 얻을 수 있을 것이지만 (예를 들어, 스마트 계약에서의 내기 결과가 자신에게 불리한 경우, 자금을 회수함으로써), 자금의 회수가 가능하기 전에 난수가 프로토콜에 확정 반영될 때까지 일정한 대기시간이 있도록 적절한 지연을 적용함으로써 스마트 계약 계층에서 이 문제가 완화될 수 있습니다.

3.2 에포크 (Epochs)

하모니에서 합의와 샤딩 프로세스는 에포크(epoch)의 개념에 의하여 조정됩니다. 에포크는 그 내부에서 샤딩의 구조가 확정되고 각 샤드는 동일한 집단의 검증자들을 사용하여 계속하여 합의과정을 진행하는, 사전에 결정된 하나의 일정한 시간간격입니다 (예를 들어, 24시간). 각 에포크의 시점에 §3.1 단원에서 기술된 DRG 프로토콜을 사용하여 하나의 난수가 생성되며 해당 난수에 근거하여 샤딩의 구조가 결정될

것입니다. 에포크 e에서 거래를 검증하기를 희망하는 검증자들은 에포크 $e - 1$의 기간 내에 그들의 토큰을 담보금으로 납입하여야 합니다. 담보금의 마감시간은 난수의 원상(preimage) pRnd가 블록체인에 확정 반영되기 이전입니다.

3.3 담보금에 근거한 샤딩

검증자 등록

시빌 공격(Sybil attack)을 [1] 방지하는 것은 공개적인 블록체인에서 하나의 중요한 고려사항이 됩니다. 비트코인과 이더리움은 그들이 블록을 제안할 수 있기 전에 채굴자들이 암호 수수께끼의 값을 (작업증명, PoW) 계산할 것을 요구합니다. 그와 유사하게, 질리카(Zilliqa)나 [12] 쿼크체인(Quarkchain) [11] 같이 샤딩에 근거한 블록체인들도 또한 시빌 공격을 방지하기 위하여 작업증명(PoW)을 사용합니다. 하모니는 검증자들의 등록 또는 시빌 공격의 방지 체계로서 지분증명(PoS)을 사용함으로써 다른 방법을 채택합니다. 하모니의 검증자가 되기 위하여 가능성 있는 참여자들은 (또는 내기 참여자) 자격을 얻을 수 있도록 일정한 양의 토큰을 담보금으로 납입하여야 합니다. 이 때, 납입된 토큰의 수가 해당 검증자에게 부여되는 투표권의 수를 결정합니다. 각 투표권 지분은 BFT 합의에서 (§2 단원에서 설명된) 하나의 투표권에 해당합니다.

투표권에 의한 샤딩

그림 3. 담보금을 낸 참여자는 담보금으로 납입한 토큰에 비례하여 투표권을 받습니다. 그 뒤 투표권은 무작위적으로 샤드들에 배정됩니다. 담보금을 낸 참여자는 그들의 투표권이 배정된 샤드(들)에서 검증자가 됩니다.

투표권은 합의에서 어떤 검증자가 하나의 표결권을 행사할 수 있도록 하여 주는 하나의 가상 티켓입니다. 검증자들은 토큰을 담보금으로 납입함으로써 투표권을 얻을 수 있습니다. 하나의 투표권을 위하여 요구되는 토큰의 양은 알고리즘적으로 조절됩니다. 각 에포크의 시작 시점에 새로운 검증자들의 투표권은 무작위적으로 샤드에 배정될 것입니다. 새로운 검증자들은 그들의 투표권이 배정되는 샤드에 합류합니다. §2 단원에서 기술된 것과 같이, 샤드에서의 합의는 합산하여 블록에 서명하기 위한 최소한 $2f + 1$개의 투표권을 가지는 검증자들에 의하여 도달됩니다.

하나의 샤드의 안전성을 보증하기 위하여, 악의의 검증자들의 투표권의 양은 해당 샤드의 모든 투표권의 1/3 미만으로 유지되어야 합니다. BFT 합의의 성격에 따라서 이것이 요구됩니다. 하모니의 적응적인 최소기준 지분증명(adaptive thresholded PoS)은 투표권의 가격을 적응적으로 조절하고 개별적인 검증자가 아니고 각 투표권을 샤드에 배정함으로써 위의 보안 요건을 충족시킵니다.

우리의 보안 가정은 담보금으로 납입된 모든 토큰들 중에서 1/4까지만 악의의 검증자들에게 귀속된다는

것입니다. 우리가 검증자들을 기준으로 샤딩을 수행한다면 (즉, 각각의 검증자를 하나의 샤드에 배정함으로써) 한 명의 악의의 검증자가 담보금으로 납입된 모든 토큰의 (또는 투표권의) 1/4을 보유하는 최악의 경우에, 해당 검증자는 자신의 샤드에서 쉽게 1/3 이상의 투표권을 가지게 될 것입니다. 그 이유는 m이 샤드의 수를 나타낼 때, 각 샤드에서의 담보금은 전체 네트워크의 담보금보다 m배가 적기 때문입니다. 우리는 이러한 공격 시나리오를 거액담보금 공격(large-stake attack)이라고 부릅니다 (단일 샤드 탈취공격의 한 가지 특수한 형태).

거액담보금 공격을 방지하기 위하여, 검증자를 기준으로 샤딩을 수행하는 대신에, 우리는 투표권을 기준으로 샤딩을 수행하였습니다 (즉, 각 투표권을 하나의 샤드에 배정함으로써). 특히, 현재의 에포크의 시작 시점에 Rnd가 공개되고 난 뒤에, 모든 투표권에 대해 무작위의 순열(random permutation)이 (Rnd를 seed로 하여) 수행되고 투표권의 순열 목록은 m개의 그룹으로 균등하게 나누어 지며 여기에서 m은 샤드의 수를 나타냅니다. i번째 그룹에 포함되는 투표권들은 샤드 i에 배정되고 그에 해당하는 검증자들도 또한 같습니다. 실제로, 어떤 한 명의 검증자가 여러 샤드들로 배정된 투표권을 가진다면 해당 검증자도 또한 여러 개의 샤드들로 배정될 수 있을 것입니다. 샤드의 리더는 해당 그룹에서 첫번째 투표권을 가지는 검증자로 결정됩니다.

더 많은 담보금을 가지는 검증자가 리더로 선택될 가능성이 높다는 것에 유의할 필요가 있습니다. 거액의 담보금은 담보금 몰수에 대한 두려움 때문에 프로토콜을 준수할 커다란 인센티브로 작용할 것이기 때문에 우리는 이것이 실제로 바람직한 시나리오라고 주장합니다 (인센티브 체계는 §7 단원에서 설명될 것입니다). 더욱이, 그들은 또한 빠르고 안정적인 네트워크와 함께 더 강력한 컴퓨터를 소유할 가능성이 높습니다.

적응적인 최소기준의 지분증명 (PoS)

악의의 담보금참여자들이 그들의 투표권을 하나의 샤드에 집중시킬 수 없도록 하기 위하여 투표권의 가격은 충분히 적은 금액이 되도록 알고리즘적으로 설정됩니다. 구체적으로, 우리는 투표권의 가격이 다음과 같이 P_{vote}개의 토큰이 되도록 설정합니다.

$$P_{vote} = \frac{TS_{e-1}}{NumShard * \lambda}$$

여기에서, λ는 보안 매개변수이고, $NumShard$는 샤드의 수 그리고 TS_{e-1}은 에포크 $e-1$에 담보금으로 납입된 토큰의 총 금액입니다.

이제 우리는 $\lambda > 600$일 때, 어떤 하나의 샤드가 1/3 이상의 악의의 투표권을 가질 확률이 (즉, 실패의 확률) 무시할 수 있을 정도로 낮다는 것을 증명합니다.

P_{vote}의 정의에 따라서, 투표권의 총 수는 $N = TS_{e-1} / P_{vote} = NumShard * \lambda$가 될 것입니다.
신뢰할 수 있는 난수의 생성과 (§3.1에서 설명된) 난수에 근거한 샤딩 프로세스가 주어졌을 때, 각 샤드에서 악의의 투표권 수의 확률분포는 초기하 분포(hypergeometric distribution)로 (즉, 비복원 무작위 표본추출) 모델화 될 수 있습니다.

$$P(X = k) = \frac{\binom{K}{k}\binom{N-K}{n-k}}{\binom{N}{n}}$$

여기에서 N은 투표권의 총 수이고 $K = N/4$는 악의의 투표권의 최대 수, $n = N/NumShard$는 각 샤드의 투표권의 수 그리고 k는 어떤 한 샤드의 악의의 투표권의 수입니다. 어떤 한 샤드의 실제적인 실패율 $P(X \leq k)$는 누적적인 초기하분포 $CDF_{hg}(N, K, n, k)$를 따르고 이것은, N이 큰 수일 때, 이항분포와 (즉, 복원 무작위 표본추출) 같게 됩니다.

$$P(X \leq k) = \sum_{i=0}^{k} \binom{n}{i} p^i (1-p)^{n-i}$$

우리는 n이 충분히 큰 수일 때, 어떤 한 샤드가 악의의 참여자에 의해 소유되는 토큰을 1/3 이상 포함할 확률이 무시할 수 있을 정도로 낮다는 것을 입증할 수 있습니다. 실제로, $n = 600$일 때, 어떤 한 샤드가 1/3 미만의 악의의 투표권을 포함할 확률은 $P(X \leq 200) = 0.999997$이며 이것은 샤드의 실패율이 (즉, 합의에 도달하지 못하는) "약 1000년에 한 번" 발생할 수 있는 정도라는 것을 나타냅니다 (24시간의 에포크 시간간격을 가정할 때). 따라서, 우리는 우리의 샤드들에 대하여 높은 수준의 보안을 보증하기 위하여 $\lambda = 600$으로 설정할 것입니다. (직관적으로, λ는 어떤 하나의 샤드가 포함하여야 하는 최소한의 투표권의 수에 영향을 미칩니다. 작업증명(PoW) 기반의 다른 샤딩 솔루션들에서 [7, 8, 12] 설명된 것과 같이, 이것은 기능적으로 어떤 한 샤드에 포함된 최소한의 노드들의 수와 유사합니다.)

이 방법은 검증자들의 수의 변동을 극복할 수 있습니다. 우리는 질리카와 [12] 같은 다른 솔루션들처럼 각 샤드에서의 검증자들의 수의 한도를 낮은 수준으로 설정하지 않습니다. 대신에, 우리는 악의의 참여자들이 하나의 샤드에서 1/3 이상의 투표권을 절대로 보유하지 못하도록 하기 위하여, 적응적인 지분증명(PoS) 기반의 모델을 채택하며 따라서 안전을 보호합니다.

3.4 재 샤딩 (Resharding)

우리는 악의의 검증자들이 하나의 샤드를 탈취할 수 없도록 하는 안전한 샤딩 체계에 대하여 설명하였습니다. 그럼에도 불구하고 샤딩의 구조가 일정하게 유지된다면 악의의 공격자들은 어떤 주어진 샤드의 검증자들을 부정 변조되도록 함으로써 해당 샤드를 탈취할 수 있습니다. 다음과 같이 세 가지 모델의 공격자들이 존재합니다.

1. 정적인 순차적응적: 공격자가 정해진 단계에서 노드들의 부분적 집단만을 부정 변조시킬 수 있는 경우. 엘라스티코(Elastico)는 [9] 공격자가 각 에포크의 시작 시점에서만 노드들을 부정 변조시킬 수 있다고 가정합니다.
2. 서행 적응적: 공격자가 에포크의 기간에 걸쳐서 노드들의 부분적 집단을 부정 변조시킬 수 있는 경우 [7, 8].
3. 완전 적응적: 공격자가 순간적으로 그리고 언제든지 노드들의 부분적 집단을 부정 변조시킬 수 있는 경우 [18].

하모니는 공격자가 일정한 수의 노드를 부정 변조시킬 수 있고 그것을 위해 일정한 시간이 필요한 서행 적응적 부정 변조를 가정합니다. 옴니레저는 [8] 동일한 부정 변조 모델을 가정하며 각 에포크에서 모든 샤드의 검증자들을 교체시킴으로써 공격을 방지합니다. 이 방법은 두 가지 주요한 문제점을 가지고 있습니다. 첫째는 각 에포크에서 부트스트래핑(bootstrapping)을 위한 높은 비용입니다. 둘째는 합의 과정에서 모든 노드들이 교체될 때의 보안에 대한 우려입니다.

하모니는 쿠쿠 규칙에 (Cuckoo-rule) 근거한 재샤딩 (resharding) 체계를 채택함으로써 이러한 문제들을

완화시킵니다 [7, 19]. 하나의 에포크의 종료 후에, 자신의 담보금을 회수하는 검증자들은 네트워크로부터 제거되고 담보금을 유지하는 검증자들만이 남게 될 것입니다. 이 에포크의 기간 중에 담보금을 납입한 새로운 검증자들은 새로운 투표권을 얻게 됩니다. 이 투표권들은 총 투표권의 중앙값 이상을 가지는 샤드들에게 무작위적으로 배정될 것입니다. 그 다음, 모든 샤드들로부터의 일정한 수의 투표권들이, 총 투표권 수의 중앙값 미만을 가지는 나머지 절반의 샤드들에게 무작위적으로 재분배 될 것입니다. [7]에서 이러한 재샤딩 체계가 보안 요건을 충족시키면서 동시에 모든 샤드들에서 투표권이 균형적으로 유지될 수 있도록 한다는 것이 입증되었습니다.

3.5 신속한 상태 동기화

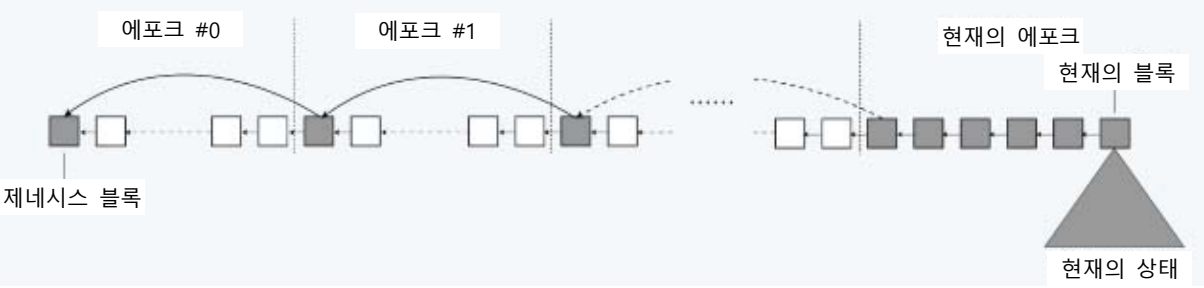

그림 4. 한 에포크의 최초 블록은 직전 에포크의 최초 블록으로의 해시 링크(hash link)를 포함합니다. 이것은 새로운 노드들의 신속한 상태 동기화를 가능하게 하며 그들은 현재의 상태를 신속하게 검증하기 위하여 회색의 블록들에만 의존하면 됩니다.

검증자들이 새로운 샤드에 합류할 때, 그들은 새로운 거래를 검증하기 위하여 샤드의 현재 상태에 신속하게 동기화 할 필요가 있습니다. 블록체인의 이력을 다운로드하고 현재의 상태를 재조합하는 재래식의 절차는 재샤딩을 구현하기 위하여 너무 늦습니다 (이더리움 블록체인의 이력을 완전히 동기화 하기 위하여는 수일의 기간이 소요됩니다). 다행히도, 현재의 상태정보는 전체적인 블록체인의 이력보다 그 크기가 훨씬 적습니다. 전체 이력을 다운로드하는 것과 비교하여 현재의 상태를 다운로드하는 것은 에포크에서 허용되는 시한 내에 가능합니다.

하모니에서, 샤드에 합류하는 새로운 검증자들은 신속하게 거래의 검증을 시작할 수 있도록 먼저 해당 샤드의 현재 상태 트리(current state trie)를 다운로드합니다. 다운로드된 현재 상태가 유효한 것이 되도록 하기 위하여, 새로운 노드는 적절한 검증을 수행하여야 합니다. 현재의 상태를 검증하기 위하여 전체 블록체인의 이력을 다운로드하고 모든 거래들을 다시 처리하는 대신에, 새로운 노드는 이력 블록의 헤더(header)를 다운로드하고 그들의 서명을 확인함으로써 헤더들을 검증합니다. 현재의 상태로부터 블록체인 최초의 블록(genesis block)까지 암호적인 추적이 (예를 들어, 해시 포인터와 서명) 가능하다면 상태는 유효한 것입니다. 그럼에도 불구하고, 서명의 검증은 연산의 비용을 수반하며 블록체인 최초의 블록(genesis block)으로부터 시작하여 모든 서명을 검증하기 위하여는 상당한 시간이 소요됩니다. 이 문제를 완화하기 위하여, 각 에포크의 최초 블록은 직전 에포크의 최초 블록으로 추가적인 하나의 해시 포인터를 포함하게 됩니다. 이러한 방법으로, 새로운 노드는 제네시스 블록으로 해시 포인터를 추적할 때 각 에포크 내부의 블록들을 건너 뛸 수 있습니다. 이것은 현재의 블록체인 상태를 검증하는 것을 현저하게 가속화시킬 수 있을 것입니다.

상태 동기화 프로세스를 더욱 최적화하기 위하여, 우리는 블록체인 상태 그 자체를 가능한 한 작게 만들 것입니다. 이더리움 블록체인의 상태를 관찰한 결과에 따르면, 많은 수의 계정이 비어 있으며 블록체인 상태정보의 귀중한 공간을 낭비하고 있는 것으로 나타났습니다. 이더리움에서, 과거의 거래가 삭제된 계정으로 재제출되는 잠재적인 재전송 공격(replay attack)의 가능성 때문에 특정의 임시값(nonce)를 가진 빈

계정들을 제거할 수 없습니다 [32]. 하모니는 거래들이 현재 블록의 해시 값을 표시하도록 함으로써 재전송 공격을 회피하는 하나의 다른 모델을 채택할 것이며, 이 경우 거래는 해당 해시 값의 블록 이후의 일정한 수의 (예를 들어, 100) 블록 이전에만 유효합니다. 이러한 방법으로 구 계정들은 안전하게 삭제될 수 있으며 블록체인 상태는 간결하게 유지될 수 있습니다.

4. 샤드 체인과 신호 체인 (Beacon Chain)

4.1 샤드 체인

샤드 체인은 자신의 거래를 처리하고 검증하며 자신의 상태를 저장하는 블록체인입니다. 각 샤드는 자신에게 적합한 거래만을 처리합니다. 샤드 체인이 상대적으로 독립적이기는 하지만, 그것은 샤드 간의 교신을 통하여 다른 샤드 체인들과 통신을 하게 될 것입니다.

샤드 간의 교신
샤드 간의 교신은 샤딩에 근거한 모든 블록체인에서 중요한 구성요소가 됩니다. 샤드 간의 교신 기능은 샤드들 간의 장벽을 없애주고 하나의 샤드의 유용성을 그 자체의 범위를 벗어나서 확장시킬 수 있도록 하여 줍니다. 전체적으로, 세 가지 부류의 샤드 간 교신이 존재합니다.

1. 주 체인 주도적: 질리카(Zilliqa)와 [12] 같은 프로젝트들은 샤드들 간의 거래를 처리하기 위하여 주 체인에 의존합니다.
2. 클라이언트 주도적: 옴니레저(Omniledger)는 [8] 샤드들 간의 메시지가 클라이언트들에 의해 수집되고 대상 샤드로 전송되는 클라이언트 주도적 샤드간 거래체계를 제안하였습니다. 이것은 클라이언트에 추가적인 부담이 되며 임시적인 경량화 클라이언트에 대해 바람직하지 않습니다.
3. 샤드 주도적: 래피드체인(RapidChain)은 [7] 샤드들 간의 메시지가 외부의 도움 없이 샤드 내의 노드들에 의해 직접적으로 전송될 것을 제안하였습니다.

하모니는 그 간소함과 클라언트에 대해 부담이 없는 이유 때문에 샤드 주도적 방법을 채택하였습니다. 우리는 샤드 주도적 통신의 이점이 그 단점보다 더 중요하다고 생각하고 있습니다. 모든 샤드들 간의 메시지가 네트워크 수준에서 전파되며 이것은 $O(N)$의 네트워크 비용을 발생시키기 때문에 샤드 주도적 통신을 위한 전체 네트워크에 대한 비용은 무시할 수 없는 수준입니다. 이 문제를 해결하기 위하여, 하모니는 통신의 복잡도를 $O(log(N))$으로 낮출 수 있도록 카뎀리아 경로설정 프로토콜(Kademlia routing protocol)을 사용합니다. 또한 전송되는 데이터는 샤드들 간의 통신의 안전성을 위해 삭제코드를 사용하여 코드화 됩니다. 상세한 내용은 §6 단원에서 설명될 것입니다.

4.2 신호 체인 (Beacon Chain)

하모니의 신호 체인은 샤드 체인과 비교하여 추가적인 목적을 수행하는 특별한 블록체인입니다. 실제로, 신호 체인도 또한 하나의 샤드 체인입니다. 다른 샤드 체인들처럼 거래를 처리하는 것에 더하여 신호 체인은, 난수를 생성하고 (§3.1 단원에서 설명된) 담보금을 수납하는, 두 가지의 추가적인 중요 기능들을 담당하며 이것은 신호 체인이 참여자들이 검증자가 되기 위하여 그들의 토큰을 담보금으로 납입하는 체인이라는 것을 의미합니다.

신호 체인에서 검증자들은 다른 샤드 체인에서와 유사한 방법으로 결정됩니다. 샤드에 대한 배정 시에, 투표권은 무작위적으로 $NumShard + b$개의 그룹으로 나누어지며 여기에서 추가적인 b개의 그룹은 신호 체인을 위한 것입니다.

샤드 체인으로부터의 해시 링크 (Hash Link)

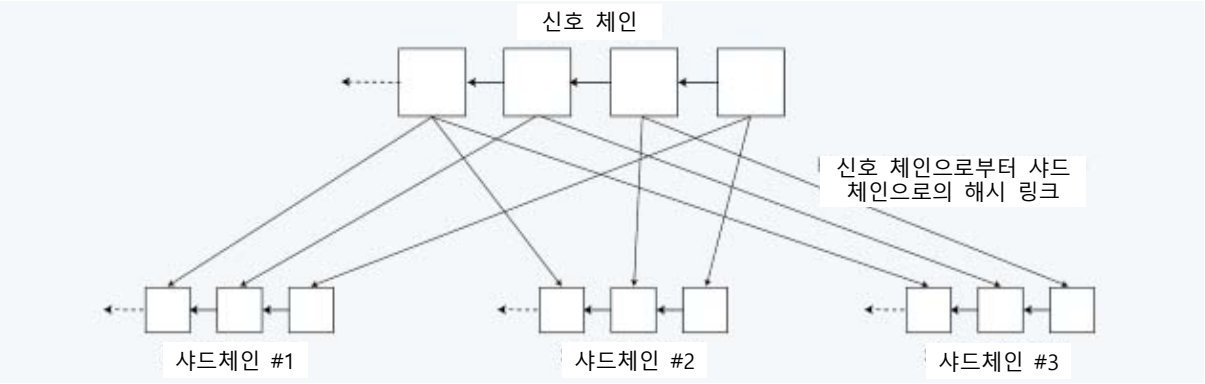

그림 5. 신호체인 블록으로부터 샤드체인 블록으로의 해시 링크(hash link)

신호 체인은 각 샤드 체인으로부터의 블록 헤더(header)를 포함함으로써 샤드 체인들의 상태의 안전성과 일관성을 더욱 강화시킬 수 있도록 하여 줍니다. 특히, 새로운 블록이 샤드 체인에 확정 반영된 이후에 해당 블록 헤더는 (카뎀리아(Kademlia)에 근거한 샤드 간의 교신을 통하여) 신호 체인으로 전송됩니다. 신호 체인은 다음의 내용을 근거로 블록 헤더의 유효성을 확인합니다.

1. 그 전 블록의 해시 값, 이것은 이미 신호 체인에 확정 반영되었을 것입니다.
2. 블록의 다중 서명의 서명자들, 이들은 해당 샤드에 대해 정확한 검증자들이어야 합니다.

신호 체인에 확정 반영된 블록 헤더들은 그 후에 전체 네트워크로 전파됩니다. 각 샤드는 다른 모든 샤드들에 대한 유효한 블록 헤더들의 체인을 유지하며 이것은 다른 샤드들로부터의 거래의 (즉, 간단한 지급 검증) 유효성을 확인하기 위하여 사용될 것입니다. 샤드 체인들의 블록 헤더들을 신호 체인에 추가하는 것은 다음과 같은 두 가지의 주요 목적을 위하여 유용합니다.

1. 단일 샤드에 대한 공격을 더욱 어렵게 합니다.
   공격자는 샤드 체인의 대체적인 블록이 유효하다는 것을 다른 참여자들에게 확신시키기 위하여 샤드 체인과 신호 체인 둘 모두를 부정 변조하여야 합니다.
2. 샤드들 간에 블록 헤더를 배포하기 위한 네트워크 비용을 감소시킵니다.
   각 샤드가 자신의 헤더를 별도로 배포하도록 한다면 $O(N^2)$의 네트워크 비용이 발생할 것입니다. 신호 체인이 중앙의 중계자 역할을 한다면 복잡도는 $O(N)$으로 감소됩니다.

5. 블록체인 상태 샤딩

UTXO (Unspent Transaction Output) 데이터 모델을 채택한 다른 상태-샤딩 블록체인들과는 [7, 8] 달리, 하모니의 상태 샤딩은 계정에 근거한 데이터 모델을 기반으로 하여 적용됩니다. 각 샤드 체인은 자신의 계정 상태를 포함하며 존재하는 모든 토큰은 모든 샤드들 간에 유포됩니다.

우리는 샤딩에서 사용자 계정과 스마트 계약 계정을 다르게 처리합니다. 하나의 사용자 계정은 서로 다른 샤드들에 여러 개의 잔액을 가질 수 있습니다 (예를 들어, 샤드 A에는 100개의 토큰 그리고 샤드 B에는 50개의 토큰). 사용자 계정은 샤드간 거래를 발행함으로써 샤드 간에 잔액을 이전시킬 수 있습니다. 스마트 계약 계정은 해당 계약이 생성된 특정의 샤드로 한정됩니다. 그러나 하나의 샤드가 처리할 수 있는 것보다 더 높은 처리성능을 요구하는 분산화 된 애플리케이션에 대하여, 분산 애플리케이션(Dapp)의 개발자는 동일한 스마트 계약의 여러 복제본들이 서로 다른 샤드들에서 구동되도록 하고 그들 각각이 유입되는 거래의 부분적 집단들을 처리하도록 할 수 있습니다. 동일한 스마트 계약의 서로 다른 본제본들은 동일한

상태를 공유하지는 않지만 그들은 샤드 간의 교신을 통하여 상호 간에 메시지를 교환할 수 있다는 것을 유의하여야 합니다.

6. 네트워크 구현

이전의 한 연구는 [33] 블록체인 시스템에 대하여 네트워크의 용량이 주요한 병목 요인들 중의 하나라는 것을 지적하였습니다. 성능을 개선하기 위하여, 하모니는 네트워크의 활용효율을 개선하는 것에 초점을 두고 있습니다. 하모니는 또한 실제 세계의 네트워크 시나리오와 관련하여 많은 개선책을 제안하고 있습니다.

6.1 카뎀리아(Kademlia)에 근거한 경로설정(Routing)

래피드체인(RapidChain)으로부터 [7] 영감을 얻어서, 우리는 샤드 간의 메시지 전송을 위한 경로설정(Routing) 체계로서 카뎀리아(Kademlia)를 [37] 채택할 것입니다. 하모니 네트워크의 각 노드는 여러 샤드들로부터의 노드들을 포함하는 경로설정 표(routing table)를 유지합니다. 샤드들 간의 거리는 샤드의 식별번호들의 XOR 거리로서 정의됩니다. 샤드 A로부터의 메시지가 샤드 B로 전달되어야 하는 경우, 샤드 A의 노드들은 경로설정 표(routing table)를 참조하여 가장 근거리의 샤드 식별번호를 가진 노드들로 메시지를 전송할 것입니다. 카뎀리아(Kademlia)에 근거한 경로설정(Routing)에서, 주어진 메시지는 목표 샤드에 도달하기 전에 *O(logN)*개의 노드만을 경유합니다. *O(N)*의 네트워크 복잡도를 필요로 하는 일반적인 메시지의 배포와 비교하여, 카뎀리아(Kademlia) 경로설정(Routing) 체계는 샤드 블록체인에서 전체적인 네트워크 부하를 현저하게 줄여 줄 수 있습니다.

6.2 삭제 코드를 사용하는 효율적인 배포

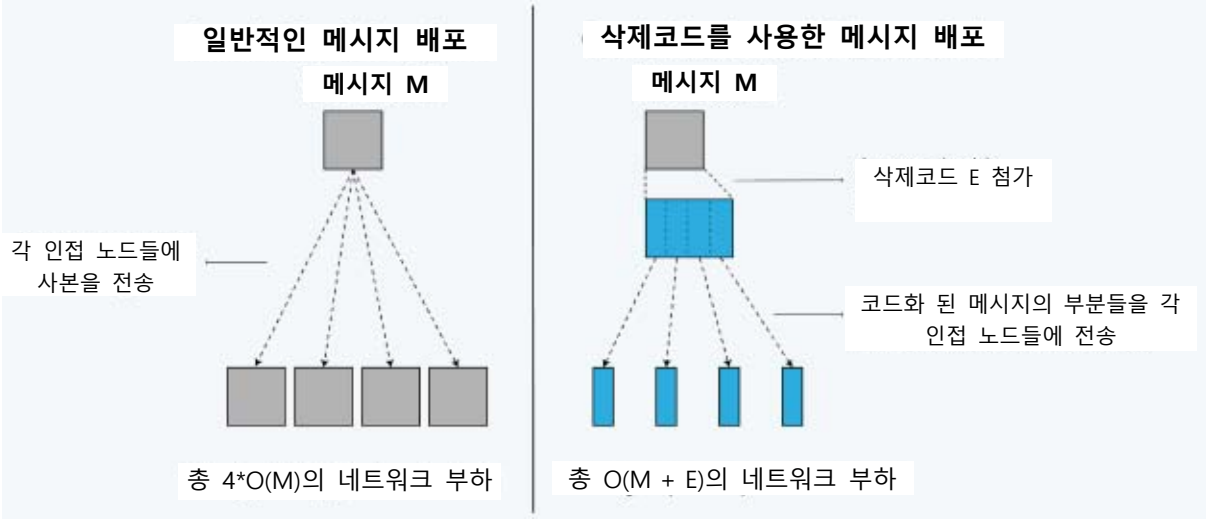

그림 6. 일반적인 메시지 배포와 삭제코드를 사용한 메시지 배포의 비교

P2P 네트워크에 기반하여 구축된 모든 블록체인 시스템에서 배포는 빈번하게 발생하는 네트워크 활동입니다. 특히, 우리의 합의 프로토콜에는 배포를 필요로 하는 세 가지의 시나리오가 존재합니다.

1. 새롭게 제안된 블록은 리더에 의해 모든 검증자들에게 배포되어야 합니다.
2. 새롭게 생성된 주 체인 (master chain) 블록은 전체 네트워크로 배포되어야 합니다.
3. 샤드들 간의 교신은 샤드들 간의 메시지의 전송을 필요로 합니다.

일반적인 P2P 배포에서, 최초의 전송자는 메시지의 사본을 자신과 인접한 각 노드들에게 전송하여야 합니다. 이것은 전송자에 대하여 $O(d * M)$의 네트워크 부하를 발생시킬 것이며, 여기에서 d는 전송자의 인접 노드들의 평균 수이고 M은 메시지의 크기입니다. 대신에 하모니에서 전송자는 먼저 삭제코드를 사용하여 메시지를 코드화 한 다음 코드화된 메시지의 분할된 부분들을 각 인접 노드들로 전송합니다. 이것은 전송자 측의 부하를 $O(M + e)$로 감소시키며, 여기에서 e는 삭제코드의 크기이며 이것은 일반적으로 원래의 메시지 M보다 크기가 작습니다. 따라서, 하모니의 네트워크 배포체계는 배포 전송자의 네트워크 부하를 크게 감소시킵니다. 또한, 하모니는 데이터가 궁극적으로 목표 노드에 도달할 가능성을 더욱 증가시키기 위하여 배포자가 항상 더 많은 삭제코드를 전송할 수 있도록 원래의 리드-솔로몬 (Reed-Solomon) 삭제코드를 랩터큐 파운틴 코드로 (RaptorQ fountain code) 교체함으로써 IDA의 안전성을 개선시키는 것을 제안하였습니다.

6.3 FEC에 근거한 특정수신자에 대한 송신 (Unicast)

TCP와 [42] 같이 신뢰성을 보증하는 재래의 통신 프로토콜은 소실되는 메시지 패킷을 방지하기 위하여 재전송과 확인(ACK)에 근거한 신호체계에 의존합니다. 이것은 전송자와 수신자 사이의 왕복 교신시간에 비례하는 대기시간 증가를 유발하는 것으로 알려져 있습니다. 또한 – 대부분의 TCP 시행사례에서 사용되고 있는 Reno, NewReno 그리고 CUBIC과 같은 – 한정된 범위창(window)에 근거한 과잉 및 정체통제는 모두 AIMD (additive increase/multiplicative decrease) 알고리즘을 사용하며 그 대역효율은 일시적인 메시지 패킷의 소실에 따라 심각하게 영향을 받는 것으로 알려져 있습니다.

하모니는 이들 두 가지 문제점들을 해결하기 위하여 랩터큐 파운틴 코드(RaptorQ fountain code)를 사용합니다. 각 메시지는 심볼로 코드화 되고 수신자가 수취된 심볼을 사용하여 메시지를 성공적으로 해독하였다는 것을 확인할 때까지 심볼은 선로를 통하여 전송됩니다. 심볼이 소진되는 경우 송신이 불가능한 리드-솔로몬(Reed–Solomon)과 같이 일정 율의 코드를 사용하는 경우와 달리, 파운틴 코드는 코드화 심볼의 무한정 적시적인 생성과 사용을 가능하게 합니다.

6.4 주거 노드의 지원

일반적인 거주지역 네트워크의 P2P 노드들은 주요하고 독특한 문제점을 발생시키며 그것은 해당 거주지역의 인터넷 라우터(Internet router)를 통하지 않고는 그들에게 도달할 수 없다는 것이며 해당 인터넷 라우터들은 네트워크 주소변환(network address translation 또는 NAT)이라고 하는 기술을 사용합니다. 이들 라우터들에 의한 수신 메시지의 지원은 그 유형이 다양하며 서로 다른 라우터들에 적용하기 위하여 서로 다른 방법들이 개발되었습니다. 특히, 대칭적인 NAT 프로토콜을 사용하는 라우터들은, IGDP(Internet Gateway Device Protocol)와 같이 다른 소통경로 구축체계(hole-punching mechanisms)를 지원하도록 명시적으로 설정되지 않은 경우 문제를 해결하는 것이 쉽지 않습니다.

하모니의 P2P 계층은 주어진 노드의 전방에서 운영되는 NAT 체계를 감지하고 STUN이나 TURN, IGDP 등과 같이 적절한 대응적 체계를 적용합니다. 특히, 하모니는 ICE(Interactive Connectivity Establishment)라고 하는 전반적인 감지 및 문제완화 프로토콜을 사용합니다.

6.5 탐지장치 이동성의 지원

노드들은 그들의 IP 주소를 변경할 수 있으며 일부 유형의 노드들은 다른 것들보다 더욱 그러합니다. 그러한 한 가지 예가 노트북 컴퓨터이며 이들은 빈번하게 서로 다른 Wi-Fi 네트워크 사이를 옮겨 다니며 그때마다 IP 주소가 달라집니다. 어떤 노드의 IP 주소가 변경될 때, IP 주소를 발신 및 수신 주소로 사용하는 모든

기존의 통신접속은 중단되고 그러한 통신접속을 직접적으로 사용하는 애플리케이션은 작업을 계속 진행하기 위하여 새로운 IP 주소를 사용하여 접속을 다시 설정하여야 합니다. 그러한 접속의 변환은 애플리케이션 계층의 최소한의 서비스 중단으로는 올바르게 처리하기가 어렵습니다. 또한 접속의 변환을 자주 처리하여야 하는 경우 애플리케이션 계층의 프로토콜에 (기본 합의 프로토콜과 같은) 어려운 문제를 초래할 수 있습니다.

이러한 문제를 해결하기 위하여, 하모니의 네트워크 계층은 산업 표준인 HIPv2(Host Identity Protocol Version 2)를 사용하여 노드의 정체성과 (노드에 의해 소유되는 암호 키 쌍) 노드의 탐지장치를 (노드와의 접속을 위해 사용되는 네트워크/트랜스포트 계층의 탐지장치) 분명하게 구분합니다. HIPv2는 탐지장치의 발견, 노드 대 노드 사이의 보안 연결 그리고 해당 발신 또는 수신 접속 주소와 관련된 상위계층의 교신접속 보호 등의 기능은 제공함으로써 노드의 정체성은 유지하되 노드의 탐지장치가 시간에 걸쳐 변경될 수 있도록 합니다.

7. 인센티브 모델

7.1 합의 보상

어떤 블록의 확정 반영이 성공적으로 수행되고 나면, 해당 블록에 서명한 모든 검증자들에게 그들의 투표권에 비례하여 프로토콜에 의하여 규정된 수의 새로운 토큰이 보상으로서 제공됩니다. 거래 수수료도 그와 유사하게 검증자들에게 보상으로 제공됩니다.

7.2 담보금의 몰수

네트워크에 의하여 탐지된 모든 부정행위에 대하여, 일정한 양의 담보금 토큰이 몰수될 것입니다. 예를 들어, 어떤 리더가 합의 프로세스를 완료하지 못하고 리더 교체 프로세스가 개시되도록 하는 경우, P_{vote}개에 해당하는 담보금 토큰이 몰수될 것입니다. 검증자들이 부정 변조된 블록에 서명을 한 것이 판명되는 경우, 해당 샤드 내에서 그들의 모든 담보금이 몰수될 것입니다. 이러한 엄격한 벌칙은 모든 부정직한 행동을 강력하게 차단하고 네트워크를 가능한 한 안전하게 만들기 위한 것입니다. 부정한 행동의 증명은 상호간에 비일관적인 두 개의 서명된 블록이 될 수 있습니다. 모든 검증자들은 어떤 다른 검증자의 부정을 증명하기 위해 거래를 제출할 수 있으며, 부정이 확인되는 경우, 몰수된 토큰은 부정 입증자에게 보상으로 제공됩니다.

7.3 담보금의 회수

장기간 잠복 공격 (Long-range Attacks)
작업증명(PoW) 블록체인과 달리, 지분증명(PoS) 블록체인은 장기간 잠복 공격(long-range attacks)에 취약성을 가질 수 있습니다. 이들은 증명이 자원 집약적인 작업이 아니고 서명에 근거하고 있다는 사실을 이용한 공격입니다. 장기간 잠복 공격(long-range attacks)에서 정직한 검증자들의 개인 비밀키가 사용되고 나서 오랜 시간이 지난 뒤에 도난 되고 공격자는 해당 키로 허위의 블록에 서명을 함으로써 새로운 버전의 블록체인을 (forked blockchain) 생성할 수 있습니다. 이러한 경우, 네트워크에 합류하는 새로운 검증자들은 원래의 합법적인 체인과 공격자의 위조 체인을 구분할 수 있는 방법이 없습니다.

장기간 잠복 공격(long-range attacks)은 다음의 두 가지 시나리오에서 발생합니다. 검증자 측의 보안의 취약점에 의해서 또는, 더 일반적으로는, 어떤 검증자가 자신의 담보금 토큰을 회수하고 난 뒤에 자신의 개인 비밀키를 구매하기를 원하는 공격자가 있다면 재무적 이익을 얻을 수 있다는 사실 때문에 개인 비밀키가 부정 유용될 수 있습니다. 또한, 설계에 따라 검증자들의 각 집단은, 검증자들의 다음 집단을 결정하는 거래들의 블록을 승인할 수 있도록 신뢰를 받습니다. 충분한 수의 (즉, 합산하여 어떤 하나의 샤드에서 투표권의 2/3 이상을 보유하는) 개인 비밀키가 누출되고 난 뒤에 공격자는 누가 다음번의 검증자가 될 것인지를 전적으로 통제할 수 있습니다.

장기간의 방어: 공명 정족수

작업증명(PoW) 블록체인은 정직한 검증자들에게 포크 (fork) 선택의 객관적인 방법을 제공함으로써 위와 같은 공격을 방지합니다. 작업증명(PoW) 블록체인에서 포크의 선택을 위해 사용될 수 있는 단 하나의 객관적인 척도는 각 블록을 승인하기 위하여 사용된 서명들의 총 가중치입니다. 만일 우리가 두 개의 서로 다른 블록들을 비교하기 위하여 이들 서명의 가중치를 사용한다면, 어떤 하나의 체인이 언제 새로운 버전으로 분리(fork)될 수 있는지를 결정하기 위하여 우리는 다음의 방정식을 고려하게 됩니다.

안전 = "블록 승인 키의 가중치" – "누출된 키의 가중치"

"블록 승인 키의 가중치"는 블록에 서명한 키들의 투표권을 의미합니다. 담보금의 가중치를 기준으로, 만일 블록을 승인하기 위해 사용된 것보다 더 많은 개인 비밀키들이 누출되었다면 해당 블록은 새로운 버전으로 분리(fork)될 수 있습니다. 그때까지, 검증자들은 언제나 원래의 합법적인 버전의 블록을 선호할 것입니다.

하모니는 이 방정식을 극대화함으로써 자신의 지분증명(PoS) 블록체인에서 각 블록의 안전을 극대화합니다. 장기적으로 개인 비밀키의 누출을 방지하는 것은 불가능합니다. 대신에 하모니는 정족수가 도달된 뒤에 각 블록의 승인 가중치를 극대화하기 위하여 검증자들에게 인센티브를 제공합니다. 이것은 검증자들이 담보금을 회수하기 전에, 해당 검증자들이 정족수에 의해 승인된 각 블록에 서명을 하도록 요구함으로써 수행됩니다. 이들 추가적인 새로운 서명들은 블록체인 내에만 존재하면 되며 합의 시점에 각 블록에 대하여 생성될 필요는 없습니다. 이러한 점 때문에, 검증자들이 그들의 담보금을 회수하기로 결정할 때, 새로운 서명들이 이후의 블록들에 추가될 수 있으며 따라서 그들은 체인의 활성 상태에 영향을 미치지 않으면서 체인의 안전을 개선시킬 수 있습니다.

8. 미래의 연구

8.1 부정 증명

검증자들의 부정을 증명할 수 있는 능력은 경량화된 클라이언트의 입장에서 그들이 받은 블록 데이터를 신뢰할 수 있도록 하기 위하여 중요합니다. 샤드들 간의 교신의 경우에, 각 샤드는 다른 샤드들의 경량화된 클라이언트입니다. 샤드들 간에 전송되는 메시지에 관해 신뢰성을 확보하는 것은 샤드들 간의 데이터의 일관성을 위하여 중요합니다. 우리는 우리의 프로토콜을 안전하게 하기 위하여 데이터 가용성과 [29] 부정 증명의 [2] 문제들을 심도 있게 연구하고 있습니다.

8.2 상태 정보가 없는 검증자들

높은 처리성능을 가진 블록체인에서, 블록체인 데이터의 규모는 기존의 체인들과 비교하여 더 빠르게 증가할 것이며 이것은 새로운 검증자들이 신속하게 동조화 할 수 있도록 하는데 있어서 주요한 문제가 됩니다. 새로운 검증자들이 적시에 동조화할 수 없다면 새로운 블록이 승인되도록 하기 위하여 검증자들의 정족수를 충족시킬 수 없을 수도 있고 정족수의 조건이 갖추어 진다고 하더라도 프로토콜의 안전이 취약해질 것입니다. 상태 블록을 제거하는 것이 문제를 완화시키기 위한 한 가지 방법이 될 것이나 상태 그 자체가 비대해질 수 있기 때문에 이것은 적절한 해결책이 아닙니다. 우리는 검증자들이 거래를 검증하기 위하여 전체 상태정보를 동조화할 필요가 없도록 상태정보가 없는 클라이언트가 가능하도록 하는 것을 활발하게 연구하고 있습니다.

참조문헌

[1] 제이 알 두세르 (J.R. Douceur), 시빌 공격 (The Sybil attack), 개인 대 개인 간의 (P2P) 시스템에 관한 제1차 국제 워크숍으로부터 (IPTPS 02), 2002년.

[2] 알-바쌈(Al-Bassam), 엠 손니노 에이(M. Sonnino, A) 그리고 부테린 브이(Buterin, V.) (2018년). 부정증명: 경량화 클라이언트 안전의 극대화 그리고 대다수의 참여자가 부정직한 환경에서 블록체인의 확장 (Fraud Proofs: Maximising Light Client Security and Scaling Blockchains with Dishonest Majorities). CoRR, abs/1809.09044.

[3] 바신, 피 (Vasin, P.) (2014년). 블록체인의 지분증명 프로토콜 버전 2 (Blackcoin's Proof-of-Stake Protocol v2), https://blackcoin.co/blackcoin-pos-protocolv2-whitepaper.pdf

[4] 에이 키아이아스(A. Kiayias), 아이 콘스탄티노우(I. Konstantinou), 에이 러셀(A. Russell), 비 데이비드(B. David) 그리고 알 올리니코프 오로보로스(R. Oliynykov. Ouroboros): 안전성을 입증할 수 있는 블록체인 지분증명 프로토콜 (A provably secure proof-of-stake blockchain protocol). Cryptology ePrint Archive, 보고서 2016/889, 2016년. http://eprint.iacr.org/.

[5] 피 다이안(P. Daian), 알 파스(R. Pass) 그리고 이 샤이(E. Shi), 백설공주: 확고하게 재구성이 가능한 합의 그리고 입증 가능하게 지분증명의 보안성을 강화할 수 있는 애플리케이션 (Snow White: Robustly reconfigurable consensus and applications to provably secure proofs of stake), Cryptology ePrint Archive, 보고서 2016/919, 2017년.

[6] 라파엘 파스(Rafael Pass) 그리고 일레인 샤이(Elaine Shi). 썬더렐라: 낙관적인 즉시적 확인 기능을 가진 블록체인 (Thunderella: Blockchains with optimistic instant confirmation). https://eprint.iacr.org/2017/913.pdf.

[7] 엠 자마니(M. Zamani), 엠 모바헤디(M. Movahedi) 그리고 엠 레이코바(M. Raykova), "래피드체인: 완전한 샤딩을 통한 고속의 블록체인 프로토콜 (RapidChain: A Fast Blockchain Protocol via Full Sharding)." Cryptology ePrint Archive, 보고서 2018/460, 2018년. https://eprint.iacr.org/2018/460.

[8] 이 코코리스-코지아스(E. Kokoris-Kogias), 피 조바노빅(P. Jovanovic), 엘 개서(L. Gasser), 엔 게일리(N. Gailly), 이 사이타(E. Syta) 그리고 비 포드(B. Ford), "옴니레저: 샤딩을 통한 안전하고 확장성 있으며 분산화 된 원장 (Omniledger: A secure, scale-out, decentralized ledger via sharding)," 보안과 비밀보호에 관한 2018년 IEEE 심포지엄, pp. 19-34, 2018년

[9] 로이 루(Loi Luu), 비스웨시 나라야난(Viswesh Narayanan), 챠오동 젱(Chaodong Zheng), 쿠날 바웨자(Kunal Baweja), 세쓰 길버트(Seth Gilbert) 그리고 프라티크 삭세나(Prateek Saxena). 공개적인 블록체인을 위한 안전한 샤딩 프로토콜 (A secure sharding protocol for open blockchains). 컴퓨터와 통신 보안에 관한 2016 ACM SIGSAC 컨퍼런스 회보, CCS '16, 17-30 페이지, 뉴욕, 뉴욕, 미국, 2016년 ACM.

[10] 조지 다네지스(George Danezis) 그리고 사라 메이클레존(Sarah Meiklejohn). 집중적으로 은행예치된 암호화폐 (Centrally banked cryptocurrencies). 제23차 연례 네트워크 및 분산 시스템 보안 심포지엄에서, NDSS, 2016년

[11] 쿼크체인 팀(The QuarkChain Team). 샤드 간의 거래 (Cross Shard Transaction). https://github.com/QuarkChain/pyquarkchain/wiki/Cross-Shard-Transaction

[12] 질리카 팀(The Zilliqa Team). 질리카 기술백서 (The zilliqa technical whitepaper). https://docs.zilliqa.com/whitepaper.pdf, 2017년 8월

[13] 사토시 나카모토(Satoshi Nakamoto). 비트코인: 개인 대 개인의 전자 현금시스템 (Bitcoin: A peer-to-peer electronic cash system), 2008년. https://bitcoin.org/ bitcoin.pdf로부터.

[14] 미구엘 카스트로(Miguel Castro) 그리고 바바라 리스코프(Barbara Liskov). 실무적인 비잔틴 오류 허용범위 (Practical Byzantine Fault Tolerance). 운영체제 설계 및 시행에 관한 제3차 심포지엄 (OSDI '99) 회보, 뉴올리언스, 루이지애나, 1999년 2월

[15] 이 코코리스-코지아스(E. Kokoris-Kogias), 피 조바노빅(P. Jovanovic), 엔 게일리(N. Gailly), 아이 코피(I. Khoffi), 엘 개서(L. Gasser) 그리고 비 포드(B. Ford). 집단적 서명을 통한 높은 일관성 그리고 비트코인의 안전성과 성능 개선 (Enhancing Bitcoin Security and Performance with Strong Consistency via Collective Signing). 보안 심포지엄에 관한 제25차 USENIX 컨퍼런스 회보, 2016년

[16] 드리즈버스 엠(Drijvers, M.), 에달라트네자드 케이(Edalatnejad, K.), 포드 비(Ford, B.) 그리고 네벤 지(Neven, G.) (2018년). 오카모토 슈노르를 제압하다: 다중서명의 입증가능한 보안에 관하여 (Okamoto Beats Schnorr: On the Provable Security of Multi-Signatures). IACR Cryptology ePrint Archive, 2018sus 417.

[17] 비 알란고트(B. Alangot), 엠 수레쉬(M. Suresh), 에이 에스 라즈(A. S Raj), 알 케이 파티나루포티(R. K Pathinarupothi) 그리고 케이 아추탄(K. Achuthan). "높은 일관성과 함께 블록체인의 확장을 위한 신뢰성 있는 집단서명 (Reliable collective cosigning to scale blockchain with strong consistency)" 2018년 네트워크 및 분산시스템 보안 심포지엄 (DISS '18) 회보.

[18] 와이 갈리드(Y. Gilad), 알 헤모(R. Hemo), 에스 미칼리(S. Micali), 지 블라코스(G. Vlachos) 그리고 엔 젤도비치(N. Zeldovich). 알고랜드: 암호화폐를 위한 비잔틴 합의 확장 (Algorand: Scaling Byzantine Agreements for Cryptocurrencies). Cryptology ePrint Archive, 보고서 2017/454, 2017년.

[19] 바루크 에워부츠(Baruch Awerbuch) 그리고 크리스찬 쉐델러(Christian Scheideler). 확장가능하고 확고한 DHT를 향하여 (Towards a scalable and robust DHT). 알고리즘과 설계에서의 병렬처리에 관한 제18차 연례 ACM 심포지엄 회보, SPAA '06, 318-327 페이지, 뉴욕, 뉴욕, 미국, 2006년. ACM.

[20] 단 보네(Dan Boneh), 조세프 보노(Joseph Bonneau), 베네딕트 분츠(Benedikt Bünz) 그리고 벤 피시(Ben Fisch). 검증가능한 지연함수 (Verifiable delay functions). CRYPTO 2018에서, 2018년.

[21] 디 보네(D. Boneh), 비 분츠(B. Bünz) 그리고 비 피시(B. Fisch). 두 검증가능한 지연함수의 연구 (A survey of two verifiable delay functions). Cryptology ePrint Archive, 보고서 2018/712, 2018년. https://eprint.iacr.org/2018/712.

[22] 폴 펠드만(Paul Feldman). 검증가능한 비대화형 비밀분산을 위한 실무적 체계 (A practical scheme for non-interactive verifiable secret sharing). 컴퓨터 과학의 기초에 관한 제28차 연례 심포지엄 회보, SFCS '87, 427-438 페이지, 워싱턴 DC, 미국, 1987, IEEE 컴퓨터 소사이어티.

[23] 이 사이타(E. Syta), 아이 토마스(I. Tamas), 디 비셔(D. Visher), 디 아이 월린스키(D. I. Wolinsky), 피 조바노빅(P. Jovanovic), 엘 개서(L. Gasser), 엔 게일리(N. Gailly), 아이 코피(I. Khoffi) 그리고 비 포드(B. Ford). 분산화된 증인 공동서명을 이용하여 권한을 "정직하거나 또는 엉망"으로 유지하는 것 (Keeping Authorities "Honest or Bust" with Decentralized Witness Cosigning). 보안과 비밀보호에 관한 제37차 IEEE 심포지엄, 2016년 5월.

[24] 비탈릭 부테린(Vitalik Buterin) 그리고 버질 그리피스(Virgil Griffith). 우호적인 최후의 장치 캐스퍼 (Casper the friendly finality gadget). CoRR, abs/1710.09437, 2017년.

[25] 이 사이타(E. Syta), 피 조바노빅(P. Jovanovic), 이 코코리스-코지아스(E. Kokoris-Kogias), 엔 게일리(N. Gailly), 엘 개서(L. Gasser), 아이 코피(I. Khoffi), 엠 제이 피셔(M. J. Fischer) 그리고 비 포드(B. Ford). 확장가능한 내편향성 분산화 무작위성 (Scalable Bias-Resistant Distributed Randomness). 보안과 비밀보호에 관한 제38차 IEEE 심포지엄, 2017년.

[26] 티 한케(T. Hanke), 엠 모바헤디(M. Movahedi) 그리고 디 윌리엄스(D. Williams). 디피니티 기술 개관 시리즈 요약 (Dfinity technology overview series consensus), 2018년 1월

[27] 이더리움 재단(Ethereum Foundation). 이더리움 백서 (Ethereum Whitepaper). https://github.com/ethereum/wiki/wiki/White-Paper.

[28] 디 보네(D. Boneh), 비 린(B. Lynn) 그리고 에이치 샤참(H. Shacham). 베일 페어링으로부터의 짧은 서명 (Short Signatures from the Weil Pairing). 암호학과 정보보안의 이론 및 응용: 암호학의 발전에 관한 제7차 국제 컨퍼런스 회보, ASIACRYPT '01, 514-532 페이지, 런던 영국, 영국, 2001년. 스프링어 펄라크(Springer-Verlag). https://www.iacr.org/archive/asiacrypt2001/22480516.pdf

[29] 이더리움 팀(The Ethereum Team). 데이터의 가용성과 삭제 코딩에 관한 주석. https://github.com/ethereum/research/wiki/A-note-on-data-availability-and-erasure-coding
[30] 에이 체푸르노이(A. Chepurnoy), 시 파라만토우(C. Papamanthou) 그리고 와이 장(Y. Zhang). 이드랙스: 상태정보가 없는 거래의 검증에 근거한 암호화폐 (Edrax: A Cryptocurrency with Stateless Transaction Validation). Cryptology ePrint Archive, 보고서 2018/968.
[31] 브이 부테린. 상태정보가 없는 클라이언트 개념 (The Stateless Client Concept). https://ethresear.ch/t/the-stateless-client-concept/172.
[32] 데릭 륭(Derek Leung), 애덤 설(Adam Suhl), 요시 질라드(Yossi Gilad) 그리고 니콜라이 젤도비치(Nickolai Zeldovich). 금고: 암호화폐를 위한 최초의 가동 (Vault: Fast bootstrapping for cryptocurrencies). Cryptology ePrint Archive, 보고서 2018/269, 2018년
[33] 이더리움 위키(Ethereum Wiki). 샤딩 블록체인에 관하여 (On Sharding Blockchain). https://github.com/ethereum/wiki/wiki/Sharding-FAQs
[34] 엠 에프 놀란(M. F. Nowlan), 제이 팔레이로(J. Faleiro) 그리고 비 포드(B. Ford). 요점: 지역성 보존 분산화 시스템 (Crux: Locality-preserving distributed systems). CoRR, abs/1405.0637, 2014년.
[35] 조지 다네지스(George Danezis) 그리고 사라 메이클레존(Sarah Meiklejohn). 집중적으로 은행예치된 암호화폐 (Centrally banked cryptocurrencies). 제23차 연례 네트워크 및 분산 시스템 보안 심포지엄에서, NDSS 2016년, 샌디에고, 캘리포니아, 미국, 2016년 2월 21일-24일. 더 인터넷 소사이어티, 2016년.
[36] 폴 드보르잔스키(Paul Dworzanski). 수임자 난수 생성, 확정-공개 그리고 최후 공개자 공격에 관한 주석 (A note on committee random number generation, commit-reveal, and last-revealer attacks). http://paul.oemm.org/commit_reveal_subcommittees.pdf.
[37] 페타르 메이몬코프(Petar Maymounkov) 그리고 데이비드 마지에르(David Mazières). 카뎀리아: xor 계량치에 근거한 개인 대 개인 (P2P) 정보시스템 (Kademlia: A peer-to-peer information system based on the xor metric). 개인 대 개인 시스템에 관한 최초의 국제 워크숍으로부터의 수정 논문에서, IPTPS '01, 53–65 페이지, 런던, 영국, 영국, 2002년. 스프링어-펄라크.
[38] 데이비드 케이 지포드(David K. Gifford), 박사학위 논문, 분산화된 컴퓨터 시스템에서의 정보 저장 (Information Storage in a Decentralized Computer System).
[39] 프린스 메하잔 (Prince Mahajan), 로렌조 알비시(Lorenzo Alvisi) 그리고 마이크 달린(Mike Dahlin). 일관성, 가용성 그리고 집중성 (Consistency, Availability, and Convergence), 기술 보고서 (UTCS TR-11-22) http://www.cs.cornell.edu/lorenzo/papers/cac-tr.pdf.
[40] 와이어트 로이드(Wyatt Lloyd) 외, 궁극적인 것을 위해 합의하지 마십시오: COPS를 사용한 광역 저장을 위한 확장가능한 인과적 일관성 (Don't Settle for Eventual: Scalable Causal Consistency for Wide-Area Storage with COPS), 운영체제 원칙에 관한 제23차 ACM 심포지엄 회보 (SOSP'11). https://www.cs.cmu.edu/~dga/papers/cops-sosp2011.pdf.
[41] 피터 베일리스(Peter Bailis), 알리 고드시(Ali Ghodsi), 조셉 엠 헬러스타인(Joseph M. Hellerstein), 아이온 스토이카(Ion Stoica). 쉬운 인과적 일관성 (Bolt-on Causal Consistency), [SIGMOD'13].
[42] 포스텔 제이(Postel, J.) (1981년). 전송제어 프로토콜 설명서 (Transmission control protocol specification). RFC 793.
[43] 루비 엠(Luby, M.), 쇼크롤라히 에이(Shokrollahi, A.), 와트슨 엠(Watson, M.), 스톡해머 티(Stockhammer, T.) 그리고 민더 엘(Minder, L.) (2011년). 대상물 인도를 위한 랩터큐 전진적 오류수정 체계 (RaptorQ forward error correction scheme for object delivery) (RFC 제6330호).