

체인링크(Chainlink)

해당 번역본은 이용자가 디지털자산 관련 정보를 편리하게 확인할 수 있도록 단순 참고용으로 번역된 것이며, 투자 권유를 목적으로 하지 않습니다. 그 내용이 정확하지 않을 수 있으므로 정확한 정보 습득을 위해서는 원문을 참고하시거나 원문 작성 주체 측에 문의하시기 바랍니다. 두나무는 해당 번역본 내용의 진실성 및 정확성을 보장하지 않고, 제공된 정보에 의한 투자 결과에 대하여 어떠한 책임도 지지 않으며, 투자 손실은 투자자에게 귀속됩니다.

Chainlink 2.0 탈중앙화 오라클 네트워크 진화의 다음 단계

로렌츠 브라이덴바흐 외

2021년 4월 15일 v1.0

초록

이 백서에서는 본래 체인링크(Chainlink) 백서에서 제시했던 초기 개념을 넘어 진화에 관한 명확한 비전을 제시합니다. 우리는 오라클 네트워크가 스마트 컨트랙트를 위한 빠르고 안정적이며 기밀성을 유지하는 범용 연결 및 오프체인(off-chain) 연산을 제공하여 기존 및 신규 블록체인을 보완하고 향상시키는 데 점점 더 큰 역할을 하게 될 것이라고 예상합니다.

우리 계획의 기초는 탈중앙화 오라클 네트워크(Decentralized Oracle Network, DON)입니다. DON은 체인링크 노드 위원회(committee)가 유지 관리하는 네트워크입니다. DON은 위원회가 배포를 선택한 제한되지 않은 범위의 오라클 기능을 지원합니다. 따라서, DON은 강력한 추상화 레이어 역할을 담당하여 DON 내에서 자체적으로 광범위한 오프체인 리소스 및 매우 효율적이면서도 탈중앙화된 오프체인 컴퓨팅 리소스에 대한 스마트 컨트랙트용 인터페이스를 제공합니다.

체인링크는 DON을 발판으로 다음 7가지 핵심 영역의 발전에 집중할 계획입니다.

- 하이브리드 스마트 컨트랙트 : 온체인 및 오프체인 컴퓨팅 리소스를 하이브리드 스마트 컨트랙트로 안전하게 구성하여, 기존의 스마트 컨트랙트 기능을 강화하기 위한 강력하고 일반적인 프레임워크를 제공합니다.
- 복잡성의 추상화 : 개발자와 사용자에게 간단한 기능을 제공하여 복잡한 기본 프로토콜 및 시스템 경계에 억지로 익숙해질 필요가 없습니다.
- 확장성 : 오라클 서비스가 고성능의 탈중앙화 시스템에서 요구하는 대기 시간과 처리량을 달성하도록 합니

다.

- 기밀성 : 민감한 데이터를 위한 새로운 강력한 기밀성 보호 기능과 블록체인의 타고난 투명성을 결합한 차세대 시스템을 구현합니다.
- 트랜잭션을 위한 공정한 순서 : 최종 사용자를 위한 공정한 방식의 트랜잭션 순서화를 지원하고 봇 및 악의적인 채굴자에 의한 선행매매(front-running) 및 기타 공격을 방지합니다.
- 신뢰의 최소화 : 높은 보안성의 탈중앙화된 블록체인에 대한 강력한 앵커링, 암호화 기법 및 암호화 경제의 보장을 활용하여 스마트 컨트랙트 및 기타 오라클 종속 시스템에 매우 신뢰할 수 있는 지원 레이어를 생성합니다.
- 인센티브 기반 (암호경제 차원의) 보안 : DON의 노드가 풍부한 리소스를 지닌 적에게 대치할 경우에도 안정적이고 올바르게 행동할 수 있도록 강력한 경제적 인센티브를 보장하는 메커니즘을 엄격히 설계하고 강력하게 배포합니다.

본 백서에서는 각 영역에서 체인링크 커뮤니티 사전 및 지속적 혁신을 제시하고 체인링크 네트워크를 위해 계획된 더 강력하고 광범위한 기능에 대한 큰 그림을 제시하고자 합니다.

목차

1. 서론
 - 1.1 탈중앙 오라클 네트워크(DON)
 - 1.2 7가지 주요 설계 목표
 - 1.3 문서의 구성
2. 보안 모델 및 목표
 - 2.1 현재 아키텍처 모델
 - 2.2 컨센서스 가정
 - 2.3 표기법
 - 2.4 신뢰 모델에 대한 참고 사항
3. 탈중앙화 오라클 네트워크 인터페이스 및 기능
 - 3.1 네트워킹
 - 3.2 연산
 - 3.3 스토리지
 - 3.4 트랜잭션-실행 프레임워크(TEF)
 - 3.5 댄풀(Mempool) 서비스
 - 3.6 발판: 기존 체인링크 기능
 - 3.6.1 오프체인 보고(OCR)
 - 3.6.2 DECO와 타운크라이어
 - 3.6.3 기존의 온체인 체인링크 서비스
 - 3.6.4 노드 평판/실적 내역
4. 탈중앙화 오라클 네트워크로 구현된 탈중앙화 서비스
 - 4.1 준비금 증명
 - 4.2 엔터프라이즈/레거시 시스템용 인터페이스
 - 4.3 탈중앙 ID
 - 4.4 우선 채널

- 4.5 기밀성 보존 디파이/믹시클
- 5. 공정한 시퀀싱 서비스
 - 5.1 선행 문제
 - 5.1.1 오라클의 선행(front-running)
 - 5.1.2 선행 매매 사용자 트랜잭션
 - 5.2 FSS 상세 정보
 - 5.2.1 트랜잭션 처리
 - 5.2.2 트랜잭션의 원자성
 - 5.3 공정한 트랜잭션 시퀀싱
 - 5.4 네트워크 레이어 고려 사항
 - 5.5 엔티티 수준의 공정 정책
- 6. DON 트랜잭션 실행 프레임워크(DON-TEF)
 - 6.1 TEF 개요
 - 6.2 트랜잭션 라우팅
 - 6.3 동기화
 - 6.4 재편성
- 7. 신뢰 최소화
 - 7.1 데이터 소스 인증
 - 7.1.1 데이터 출처 인증(ADO)의 한계
 - 7.1.2 기밀성
 - 7.1.3 소스 데이터 결합
 - 7.1.4 소스 데이터 처리
 - 7.2 DON 신뢰 최소화
 - 7.2.1 페일오버(failover) 클라이언트
 - 7.2.2 마이노리티 리포트
 - 7.3 가드 레일
 - 7.4 최소 신뢰의 거버넌스
 - 7.5 공개키 인프라
- 8. DON 배포 관련 고려 사항
 - 8.1 롤아웃(Rollout) 접근 방식
 - 8.2 동적인 DON 멤버십
 - 8.3 DON 책임성
- 9. 경제학 및 암호경제학
 - 9.1 예치 개요
 - 9.2 배경
 - 9.3 모델링 가정
 - 9.3.1 1단계 인센티브 모델: 합리적 행위자
 - 9.3.2 2차 판정 모델: 가정에 의한 정확성
 - 9.3.3 적대적 모델
 - 9.3.4 암호경제학적 보안은 어느 정도면 충분한가?
 - 9.4 예치 메커니즘: 개요
 - 9.4.1 메커니즘 추가 상세 정보
 - 9.4.2 제곱 예치 영향
 - 9.4.3 2단계의 실현
 - 9.4.4 허위 보고 보험

- 9.5 단일 라운드 변형
- 9.6 암묵적 인센티브 프레임워크(IIF)
 - 9.6.1 미래 수수료 기회
 - 9.6.2 투자성 FFO
 - 9.6.3 외부 평판
 - 9.6.4 개방형 IIF 분석
- 9.7 종합적 논의: 노드 운영자 인센티브
- 9.8 경제적 보안의 선순환
- 9.9 네트워크 성장을 견인하는 추가 요인
- 10. 결론
 - A. 용어 정리
 - B. DON 인터페이스: 추가 세부사항
 - B.1 네트워킹
 - B.1.1 무결성
 - B.1.2 기밀성
 - B.1.3 가용성
 - B.2 연산
 - B.2.1 신뢰할 수 있는 실행 환경(TEE)
 - B.2.2 TEE 보안
 - B.2.3 안전한 다자간 연산(MPC)
 - B.3 스토리지
 - B.4 리소스 가격 책정
 - C. 어댑터 예시
 - C.1 오라클 중재 데이터 소스 액세스(MediatedReport)
 - C.2 교차 원장 보고서(XL-Report-Read)
 - C.3 기밀 스위치(ConfSwitch)
 - D. 기능 서명
 - D.1 데이터 결합을 위한 기능 서명
 - D.2 이산화된 기능 서명
 - E. 잠재적 뇌물
 - F. 무작위 오라클과 위원회 기반 오라클 중 선택

1. 서론

블록체인 오라클(Blockchain oracles)은 오프체인 리소스(off-chain resources)에서 블록체인으로 데이터를 전달하는 단일의 목표를 가진 탈중앙화 서비스로 간주됩니다. 그러나 이는 데이터 전달에서 컴퓨팅, 저장 또는 양방향 전송에 이르는 짧은 단계를 말합니다. 이러한 관점은 오라클 기능에 대한 훨씬 더 광범위한 개념을 허용하게 됩니다. 증가하는 스마트 컨트랙트(smart contracts) 서비스 요구 사항과 오라클 네트워크의 연관성은 다양한 경제적 전술을 또한 마주치게 합니다. 간단히 말하면, 오라클은 온체인(on-chain)과 오프체인(off-chain) 시스템 간 양방향 컴퓨팅 지원 인터페이스가 될 수 있으며, 그렇게 되어야 할 것입니다. 블록체인 생태계에서 오라클 역할은 다양한 산업에 새로운 신뢰 모델과 투명성을 제공할 수 있도록 스마트 컨트랙트 성능, 기능, 상호 운용성을 향상하는 것입니다. 이러한 변화는 블록체인의 특별한 속성을 오라클

네트워크와 같은 오프체인 시스템의 고유한 기능과 융합하여 개별적인 온체인 시스템보다 훨씬 더 큰 범위와 효과를 창출하는 **하이브리드 스마트 컨트랙트** 사용을 통해 전개될 것입니다.

이 백서에서 우리는 원래 체인링크 백서[98]에서 제시했던 초기 개념을 넘어 체인링크 진화형인 체인링크 2.0에 대한 비전을 명확히 제시하고자 합니다. 우리는 오라클 네트워크가 하이브리드 스마트 컨트랙트를 위해 빠르고 안정적인 기입물용가로서 범용 연결 및 연산을 제공하여 기존 및 신규 블록체인 보안 시스템과의 더 없이 적절한 결합을 담당할 것이라고 예상합니다. 또한, 유효한 데이터워크를 무결성이 높은(high-integrity) 블록체인 등급 데이터로 블록체인 생태계를 초월한 시스템으로 내보내기 위한 여러 유틸리티로 진화할 것이라 믿습니다.

오늘날 다양한 주체의 집합에 의해 실행되는 체인링크 노드는 오라클 네트워크에서 함께 소위 **보고서(reports)**라고 알려진 스마트 컨트랙트에 데이터를 전달합니다. 이러한 오라클 노드는 고전적인 컨센서스 블록체인[72]과 유사한 위원회라고 볼 수도 있겠지만, 이는 독립적인 기능을 제공하기보다는 기존 블록체인을 지원하는 것을 목표로 합니다. 검증 가능한 랜덤 함수(verifiable random functions, VRF) 및 오프체인 리포팅(Off-Chain Reporting, OCR)을 갖춘 체인링크는 미리 스마트 컨트랙트에 고급 기능에 요구되는 연산 리소스를 제공할 수 있는 하나의 범용 프레임워크 및 인프라로 진화하고 있습니다.

체인링크 2.0 계획은 탈중앙화 오라클 네트워크(Decentralized Oracle Networks, DON)를 토대로 삼습니다. 기존의 체인링크 백서[98]에서 ‘오라클 네트워크’라는 용어를 도입한 이래로 오라클은 더욱 풍부한 기능과 광범위한 애플리케이션을 개발해 왔습니다. 이 논문에서는 체인링크 생태계에 대한 우리의 미래 비전을 바탕으로 이 용어를 새롭게 정의합니다. 이러한 관점에서 볼 때, DON은 체인링크 노드 위원회(committee of Chainlink nodes)에서 유지 관리하는 네트워크라 할 수 있습니다. 컨센서스 프로토콜에 기반을 둔 위원회에 의해 배포하도록 선택한 무작위 범위의 오라클 기능을 지원합니다. 따라서 DON은 블록체인의 수장과 연계함으로써 스마트 컨트랙트 및 블록체인 시스템 모두에 대해 오프체인 리소스에 대한 인터페이스를 제공합니다. 더불어 매우 효율적인 데이터 집합 및 탈중앙화된 오프체인 컴퓨팅 리소스에 대한 액세스를 제공합니다. 일반적으로 DON은 **하나의 애플리케이션 내에 존재**해서 작업을 지원합니다. 그 목표는 오프체인 및 온체인 연산을 모두 지원할 수 있는 새로운 신뢰 모델을 만들며 하이브리드 스마트 컨트랙트를 오프체인 리소스 및 외부 시스템에 연결하는 것입니다.

DON에서 위원회를 사용하더라도 체인링크 자체는 본질적으로 무허가(permissionless) 상태입니다. DON은 무허가 방식의 프레임워크 기초 역할을 제공하며 여러 노드가 함께 모여 하향식이거나 무허가형인, 자체적인 체제의 맞춤형 오라클 네트워크를 구현할 수 있도록 합니다.

우리는 이러한 DON을 기초로 하여, 체인링크 2.0에서는 **하이브리드 스마트 컨트랙트**, 복잡성의 추상화, 확장성, 기밀성, 트랜잭션 순서 공정성, 신뢰 최소화, 인센티브 기반(암호화 경제) 보안 등 7가지 핵심 영역의 발전에 집중할 계획입니다. 섹션 1.1에서는 탈중앙화 오라클 네트워크(DON)를 개괄하고 섹션 1.2에서는 7가지 핵심 혁신 분야를 설명합니다. 섹션 1.3에서는 본 백서의 구성에 관해 설명합니다.

1.1 탈중앙 오라클 네트워크(DON)

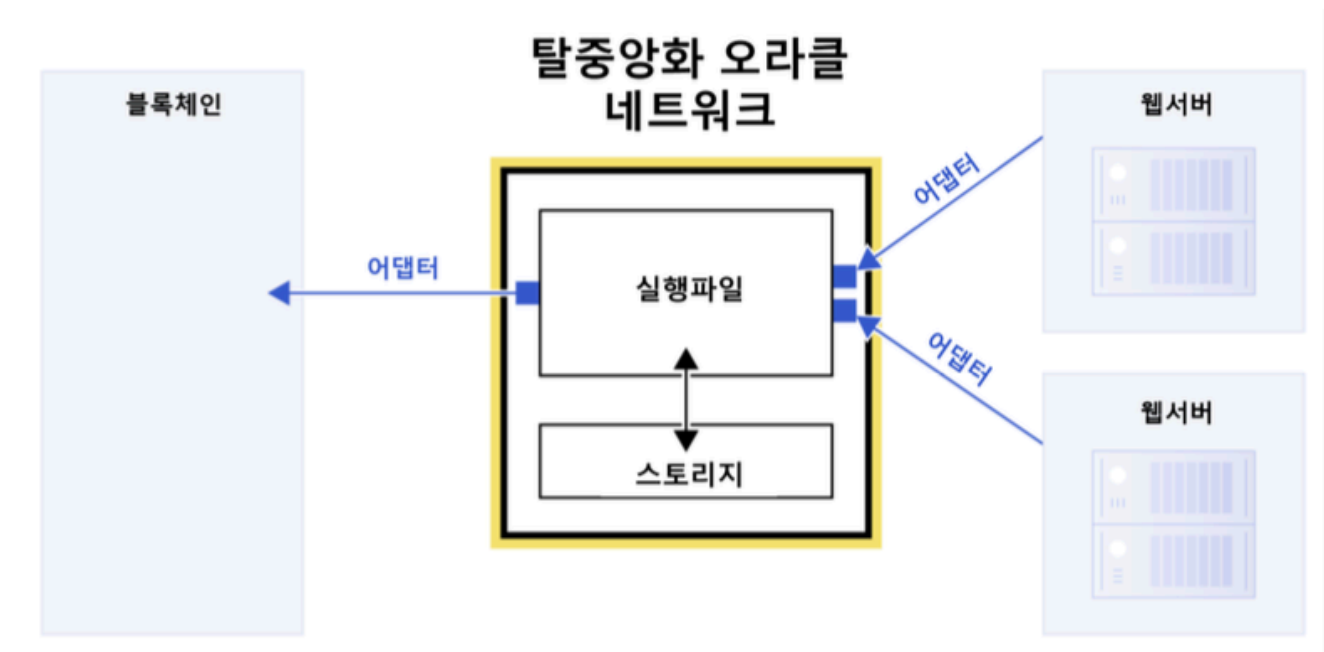
탈중앙 오라클 네트워크(DON)는 기본 기능 외의 기능을 통해 **대상 블록체인 또는 메인 체인** 상에서 스마트 컨트랙트의 기능을 향상 및 확장하도록 설계되었습니다. DON은 컴퓨팅 시스템에 존재하는 세 가지 기본 리소스의 네트워크, 스토리지 및 연산을 제공하여 이를 수행합니다. DON은 이러한 리소스를 강화한 기밀성(Confidentiality), 무결성(Integrity) 및 가용성(Availability)의 세 가지 속성과 함께 제공하는 것을

목표로 합니다.

DON은 특정 작업을 수행하기 위해 협력하거나 클라이언트에게 지속적인 서비스를 제공하기 위해 장기적인 관계를 구축하는 오라클 노드 위원회로 구성됩니다. DON은 블록체인에 구매받지 않는 방식으로 설계되었고, 애플리케이션 개발자가 지원되는 모든 메인 체인 상에서 스마트 컨트랙트에 대한 오프체인 지원을 생성할 수 있도록 하는 강력하고 유연한 도구를 제공할 것입니다.

DON은 **실행파일(executables)** 과 **어댑터(adapters)** 라는 두 가지 유형의 기능을 통해 실현됩니다. 실행파일은 DON에서 지속해서 탈중앙화 방식으로 실행되는 프로그램입니다. 이는 메인 체인 자산을 직접 저장하지는 않지만, 고성능 및 기밀 연산을 수행할 수 있는 기능 등 중요한 이점을 제공합니다. 실행파일은 DON에서 자율적으로 실행되며 결정적인 작업을 수행하는데, DON을 외부 리소스에 연결하고 실행파일에 의해 호출될 수 있는 어댑터에 의해 자동화가 가능합니다. DON용 어댑터는 오늘날 체인링크에 존재하는 외부 어댑터와 유사한 개념입니다. 기존의 어댑터는 일반적으로 데이터 소스에서만 데이터를 가져오는 반면, 이 어댑터는 양방향으로 작동할 수 있습니다. DON에서는 DON 노드의 공동 연산을 추가로 활용하여 실행파일의 개인정보보호 소비법 대한 보고서를 암호화하는 등의 추가 기능을 달성할 수 있습니다.

DON의 기본 층위에 대한 이해를 돕기 위해, DON은 블록체인에 보고서를 전송하여 기존의 전통적인 오라클 기능을 달성하는 방법을 개념적으로 정리하면 그림 1과 같습니다. 그러나 DON은 체인링크의 기존 네트워크 외에도 많은 추가 기능을 제공할 수 있습니다. 예컨대, 그림 1의 일반적인 구조 내에서 실행파일은 가져온 자산 가격 데이터를 DON에 기록하고 이러한 데이터를 사용하여 보고서의 후행 평균을 연산할 수 있습니다.



- 그림 1: 탈중앙화 오라클 네트워크(DON)가 기본 오라클 기능 즉, 오프체인 데이터를 컨트랙트로 중계하는 방법. 실행파일은 어댑터를 사용하여 오프체인 데이터를 가져오고 이를 연산하여, 다른 어댑터를 통해 대상 블록체인으로 출력값을 보냅니다. (어댑터는 작은 파란색 상자로 표현되는 DON의 코드로 시작되며, 화살표는 본 예시에 국한된 데이터 흐름 방향을 나타냅니다.) 실행파일은 상태를 유지하거나 다른 실행파일과 통신하기 위해 로컬 DON 저장소를 추가로 읽고 쓸 수 있습니다. DON의 유연한 네트워킹, 연산 및 스토리지는 호스트의 새로운 애플리케이션을 도입 가능하게 합니다.

DON의 주요한 이점은 새로운 블록체인 서비스를 부트스트래핑(bootstrap) 시킬 수 있다는 것입니다. DON은 오늘날 특수 제작된 네트워크가 필요한 서비스 애플리케이션을 기존의 오라클 네트워크로도 빠르게 운용할 수 있도록 하는 수단이 됩니다. 섹션 4에서는 이러한 애플리케이션에 관한 다양한 사례를 제공합니다.

섹션 3에서는 개발자와 사용자에게 제공하는 인터페이스의 측면에서 DON의 기능을 설명하고 DON에 대한 자세한 내용을 제공합니다.

1.2 7가지 주요 설계 목표

이 섹션에서는 체인링크 진화에 대해 위에서 열거한 7가지 핵심 중점 사항을 간략하게 검토합니다.

하이브리드형 스마트 컨트랙트: 체인링크에 대한 우리 비전의 핵심은 스마트 컨트랙트에서 온체인 및 오프체인 구성요소를 안전하게 결합한다는 발상에 있습니다. 이 아이디어를 실현하는 컨트랙트를 **하이브리드 스마트 컨트랙트(hybrid smart contracts)** 또는 ‘하이브리드 컨트랙트(hybrid contracts)’라고 합니다.

블록체인은 탈중앙화 서비스 생태계에서 두 가지 중요한 역할을 현재는 물론, 앞으로도 계속 담당할 것입니다. 블록체인은 디지털 자산 소유권이 표현되는 장소이자 탈중앙화 서비스를 구현할 강력한 앵커입니다. 그러므로 스마트 컨트랙트는 체인에서 표현되거나 실행되어야 하는 온체인 기능은 심각하게 제한적인 반면, 수많은 온체인 컨트랙트 코드는 느리고 비싸고 활용 범위가 좁아 다양한 형태의 기밀성 연산, 채굴 자문부터 안전한 (의사무작위) 생성/유효성 검사 조차 등 체인에서 본질적으로 달성할 수 없는 다양한 기능과 실제 데이터의 이점을 누릴 수 없습니다.

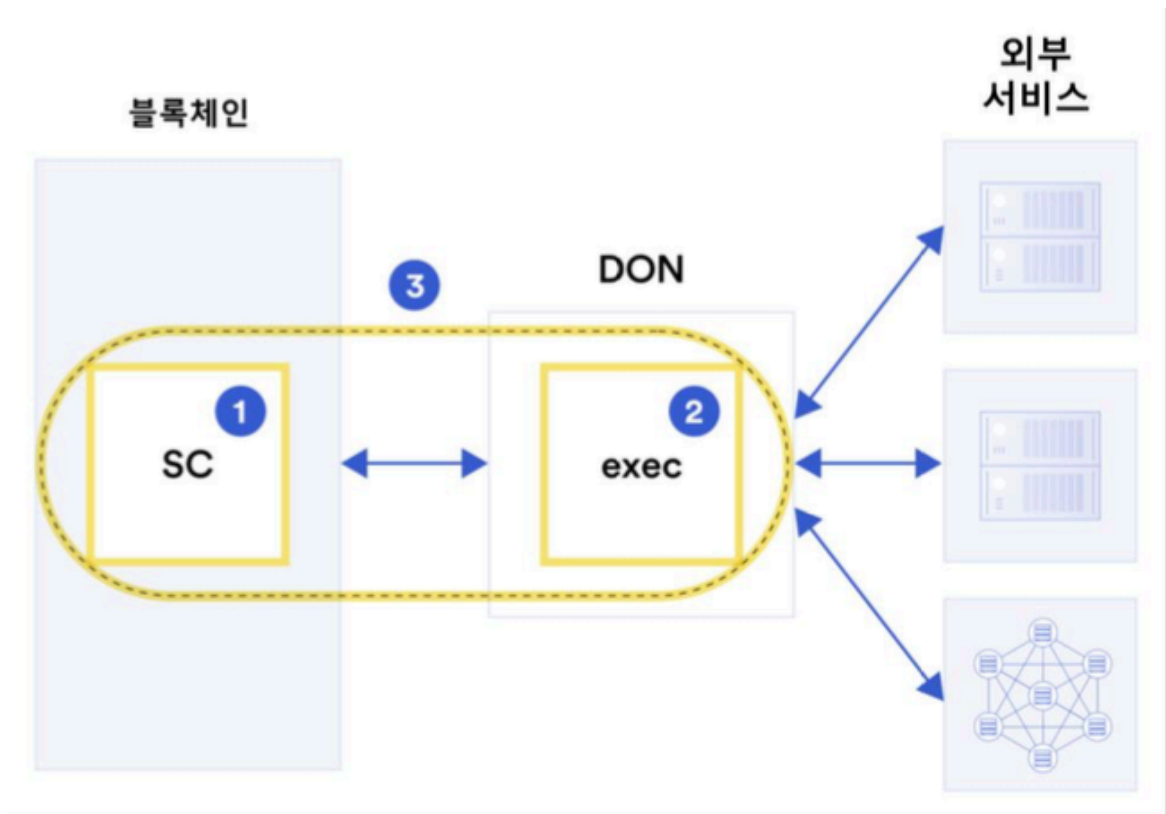
따라서 스마트 컨트랙트의 잠재력을 최대한 발휘하려면 스마트 컨트랙트가 온체인(일반적으로 SC로 표시)과 오프체인(DON에서 실행되는 실행파일로 일반적으로 exec으로 표시함)이라는 두 가지 부분으로 구조화되어야 합니다. 최종 목표는 DON이 제공하고자 하는 다양한 오프체인 서비스와 함께 온체인 기능이 안전한 구성을 달성하는 것입니다. 이 두 부분은 함께 하나의 하이브리드 컨트랙트를 구성합니다. 이러한 발상을 도식화하면 그림 2와 같습니다. 이미 오늘날 데이터 피드 및 VRF와 같은 체인링크 서비스는 디파이(Decentralized Finance, DeFi)에서 공정하게 생성된 NFT, 탈중앙화 보험에 이르기까지, 다른 방법으로는 달성할 수 없는 스마트 컨트랙트 애플리케이션으로 보다 일반적인 프레임워크를 향한 첫 번째 단계를 밟아 나가고 있습니다. 체인링크 서비스가 이 백서의 비전에 따라 확장되고 성능이 향상되면, 모든 블록체인에서 스마트 컨트랙트 시스템의 비중은 더욱 커질 것입니다.

이 백서에서 제시하는 다른 6가지 핵심 사항은 가장 중요한 하이브리드 스마트 컨트랙트 서비스를 위한 것으로도 볼 수 있습니다. 이러한 사항에는 하이브리드 컨트랙트에서 가시적인 복잡성을 제거하고, 훨씬 더 유용한 하이브리드 컨트랙트를 구성할 수 있는 추가 오프체인 서비스를 생성하는 과정이 수반되며, **신뢰 최소화**의 경우 하이브리드 컨트랙트로 달성한 보안 속성을 강화하는 과정 또한 포함됩니다. 백서의 많은 부분에서는 하이브리드 컨트랙트에 대한 아이디어를 암묵적으로만 제시합니다. 그러나 메인체인 로직과 DON의 모든 조합은 하이브리드 컨트랙트로 간주할 수 있습니다.

복잡성의 추상화

DON은 그 강력하고 유연한 서비스의 예전에 뒤에 숨은 복잡한 구조를 추상화하여 개발자와 사용자가 탈중앙 시스템을 쉽게 사용할 수 있도록 합니다. 기존 체인링크 서비스도 이러한 특성이 있습니다. 예컨대, 오늘날 체인링크 데이터 피드는 개발자가 프로토콜 수준의 세부 사항(OCR) 탈중앙 또는 집합 간의 컨센서스 그룹을

서빙하는 수단 등 많은 필수 요인은 복원된 인터페이스를 제공합니다. 체인링크가 개발자에게 고급 서비스를 간소화시켜 사용하기 쉽게 추상화 인터페이스를 제공할 수 있는 서비스의 범위를 확장시킨다는 점에서 볼 때 DON의 의미는 더 크다고 할 수 있습니다.



- 그림 2: 온체인/오프체인 컨트랙트의 구성 개념. 하이브리드 스마트 컨트랙트 ③은 블록체인에 상주하는 온체인 구성요소 SC(1) 및 DON에서 실행되는 오프체인 구성요소 exec(2)가 둘이 서로 보완하는 형태로 보완된 형태로 구성됩니다. DON은 이 두 구성요소 사이의 다리 역할을 하며 웹 서비스, 기타 블록체인, 탈중앙 스토리지 등과 같은 오프체인 리소스와 하이브리드 컨트랙트를 연결합니다.

섹션 4에서는 이러한 접근 방식을 강조하는 몇 가지 응용 사례를 제시합니다. 예컨대 우리는 기업에서 레거시 시스템을 블록체인에 연결하기 위해 보안 미들웨어의 한 가지 형태로 DON을 사용할 것으로 예상할 수 있습니다(섹션 4.2 참조). 이러한 DON의 사용은 일반적인 블록체인 역학(수수료, 재편성 등)의 복잡성을 추상화합니다. 또한 특정 블록체인 기능을 추상화하여 기업이 이러한 시스템을 이용할 수 있게 하거나, 탈중앙 시스템의 개발에 대한 전문 지식이 없어도 계속해서 확장되는 블록체인 시스템에 기존 시스템을 연결할 수 있게 해줍니다.

궁극적으로 우리는 체인링크에 의해 달성되는 추상화의 수준을 이른바 탈중앙화 메타레이어(decentralized metalayer) 단계까지 구현하고자 합니다. 이러한 레이어는 디앱(Decentralized Application, DApp)의 모든 클래스 개발자와 사용자에게 온체인/오프체인 구분을 추상화하여 탈중앙화 서비스를 원활하게 생성하고 사용할 수 있게 해줍니다.

개발 프로세스를 단순화하기 위해 개발자는 메타레이어 디앱 기능을 통한 모델 기반 애플리케이션으로 지정할 수 있습니다. 그런 다음 탈중앙화 메타레이어 컴파일러를 사용하여 블록체인, DON 및 외부 서비스에 걸쳐 상호 운용되는 탈중앙화 기능 컨테이너를 자동으로 인스턴스화할 수 있습니다. 이러한 외부 서비스 중 하나가 엔터프라이즈 시스템이라면 메타레이어는 레거시 엔터프라이즈 시스템과 관련된 애플리케이션에 유용

하게 쓸 수 있습니다. 이러한 컴파일은 최신 컴파일러 및 소프트웨어 개발 키트(software-development kits, SDK)가 프로그래머들이 범용 CPU 및 GPU, 머신러닝 가속기, 신뢰할 수 있는 엘리먼트 등을 사용하여 하드웨어 기종 하드웨어 아키텍처의 잠재력을 최대한 활용하도록 지원하는 방법과 유사합니다. 그림 3에서는 이러한 발상을 개념적으로 보여줍니다.

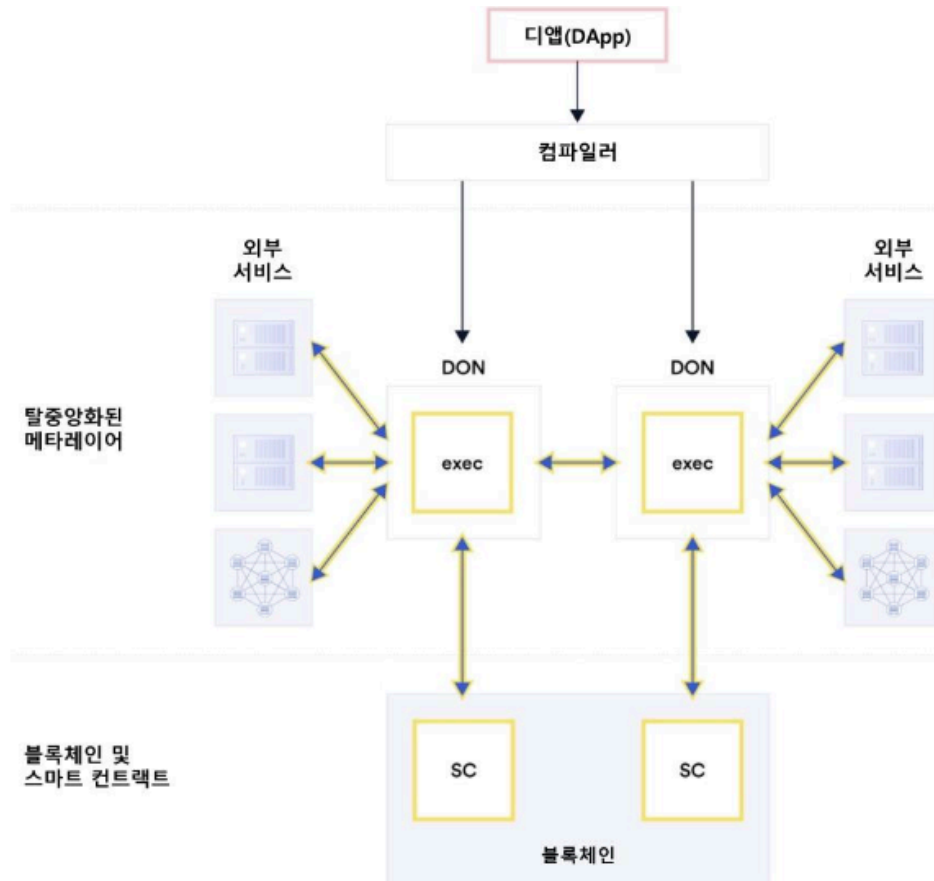
하이브리드형 스마트 컨트랙트는 이러한 비전과 메타 컨트랙트라는 개념을 추구하는 첫 번째 단계입니다. 메타 컨트랙트는 탈중앙화 메타레이어에 코딩된 애플리케이션을 의미하며, 암묵적으로 온체인 로직(스마트 컨트랙트)은 물론 다양한 블록체인과 기존 오프체인 서비스 간 오프체인 연산 및 연결을 포함합니다. 그러나 어떤 컴파일러 지원, 새로운 보안 모델, 이중 기술 개념 및 기술 조합에 대한 필요성을 감안할 때 진정한 탈중앙화 메타레이어 실현은 장기간에 걸쳐 추구해야 할 야심찬 목표라 할 수 있습니다. 그렇듯 불구하고 이는 유일한 이상적인 모델입니다. 이와 관련한 자세한 내용은 본 백서에서는 다루지 않지만, 체인링크에 관한 다른 작업에서 중점적으로 다룰 계획입니다.

확장성: 우리의 진화하는 설계에 있어 가장 중요한 목표는 체인링크 네트워크가 블록체인 생태계의 증가하는 확장성 요구를 충족할 수 있도록 하는 것입니다.

기존의 무허가 방식의 블록체인에서 네트워크 혼잡이 지속적으로 문제시 됨에 따라[86] 새롭고 더 뛰어난 성능의 블록체인 설계(예: [103, 120, 203])와 보완적인 레이어 0-2 확장 기술(예: [5, 12, 121, 141, 169, 186, 187])이 사용되고 있습니다. 오라클 서비스는 이러한 시스템의 성능 요구 사항을 충족하는 대기 시간과 처리량을 달성하는 동시에 컨트랙트 운영자와 일반 사용자 모두를 위한 온체인 요금(예: 수수료 비용)을 최소화해야 합니다. 나아가 DON을 갖춘 체인링크 기능은 순수한 웹 기반 시스템에 매우 높은 성능을 제공하는 것을 목표로 합니다.

DON은 지원하는 블록체인과 결합한 위원회 기반 또는 무허가 방식의 빠른 컨센서스 프로토콜을 통해 성능 향상을 유도합니다. 각자 다른 구성을 가진 많은 DON이 병렬로 실행될 것으로 예상됩니다. 서로 다른 디앱과 사용자는 애플리케이션 요구 사항에 따른 기본적인 컨센서스 선택에 있어서의 절충(trade-off)을 모색할 수 있습니다.

DON은 사실상 레이어-2 기술로 볼 수 있습니다. 우리는 다른 서비스 중에서도 DON이 트랜잭션 실행 프레임워크(Transaction Execution Framework, TEF)를 지원할 것으로 예상합니다. 이 프레임워크는 DON의 효율적인 통합을 진전하고 롤업(성능 개선을 위해 오프체인 트랜잭션을 번들링하는 시스템)과 같은 다른 고성능 레이어-2 시스템과 오라클을 쉽게 통합할 수 있습니다. TEF는 섹션 6에서 소개하도록 하겠습니다.



- 그림 3: 탈중앙화 메타레이어의 이상적 구현 개념. 개발에 용이하도록 개발자는 분홍색으로 표시된 디앱을 통한 시스템 모델에서 가상 애플리케이션으로 지정할 수 있습니다. 탈중앙화 메타레이어 컴파일러는 해당 상호 운용 기능인 스마트 컨트랙트(SC로 표시), DON에 대한 로직(exec으로 표시), 대상 외부 서비스에 연결하는 어댑터 등(노란색 강조 표시)을 자동으로 생성합니다.

그림 4에서는 DON이 온체인이 아닌 오프체인에서 트랜잭션 및 오라클 보고서 처리에 집중하여 블록체인(스마트 컨트랙트) 확장성을 개선하는 개념적 방법을 보여줍니다. 이와 같은 주요 연산 위치 이동은 트랜잭션 처리량을 높이는 동시에 트랜잭션 대기 시간과 수수료를 줄입니다.

기밀성: 블록체인은 스마트 컨트랙트와 애플리케이션 간에 전례 없는 수준의 투명성을 제공합니다. 그러나 투명성과 기밀성 사이에는 기본적으로 갈등이 존재합니다. 예를 들어, 오늘날 사용자의 탈중앙화 거래소 트랜잭션이 체인에 기록되기 때문에 거래를 쉽게 모니터링할 수 있을 뿐만 아니라 사용자의 금융 트랜잭션을 공개적으로 조회할 수 있습니다. 마찬가지로 스마트 컨트랙트에 중계된 데이터는 체인에 남게 됩니다. 데이터를 편리하게 감사할 수 있지만 민감한 데이터나 독점 데이터가 포함된 스마트 컨트랙트를 제공하고자 하는 데이터 제공자에게는 방해가 됩니다.

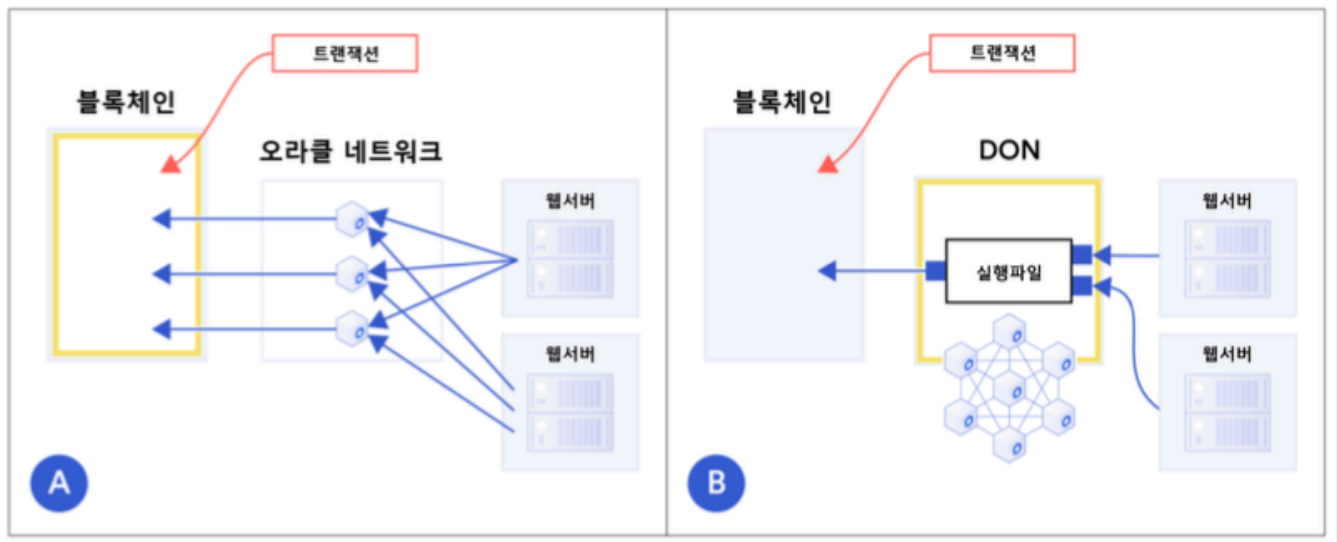
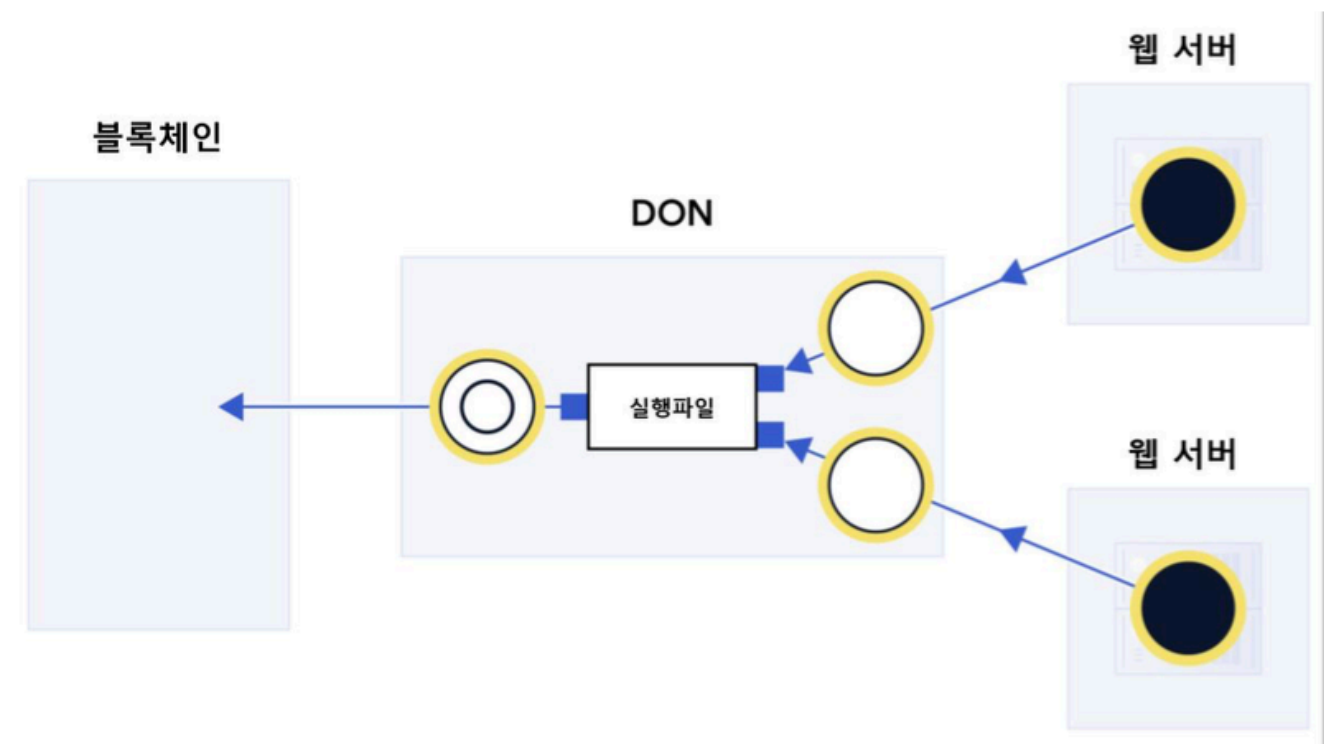


그림 5에서는 DON에서 기밀성 보존 어댑터와 기밀 연산을 통해 민감한 데이터가 외부 소스에서 스마트 컨트랙트로 흐르는 방식을 개념적으로 보여줍니다.



- 그림 5: 민감 데이터(노란색 강조 표시)에 대한 DON의 기밀성 보존 작업 개념도. 웹 서버의 민감한 소스 데이터(검은색 원)는 기밀성 보존 어댑터(파란색, 이중 화살표 선)를 사용하여 DON으로 추출됩니다. DON은 이러한 어댑터로부터 파생된 데이터(중공 원)를 수신합니다. 즉, 민감한 소스 데이터에 기능 또는 비밀 공유 등을 적용한 결과입니다. DON의 실행파일은 파생 데이터에 기밀 연산을 적용하여 보고서(이중 원)를 구성하고 이를 어댑터를 통해 블록체인으로 보낼 수 있습니다.

우리는 기밀 데이터를 처리하는 강력한 도구가 모든 범위의 애플리케이션을 실현할 것이라 믿습니다. 이러한 애플리케이션으로는 민간 부문의 탈중앙화(및 중앙화) 금융, 탈중앙화 신원(ID), 신용 기반 온체인 대출, 보다 효율적이고 사용자 친화적인 고객 신원확인(KYC) 및 인증 프로토콜 등이 존재합니다. 이와 관련한 내용은 섹션 4에서 논의합니다.

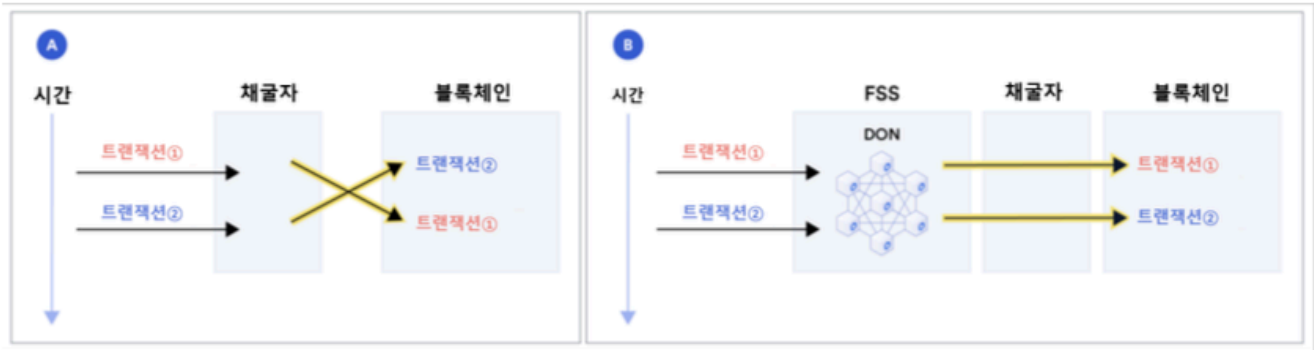
트랜잭션을 위한 공정한 순서: 오늘날의 블록체인 설계에는 아는 사람은 아는 불편한 비밀이 한 가지 있습니다. 블록체인이 짧은 기간 동안 중앙화된다는 것입니다. 채굴자(miner)와 검증자 validator)은 트랜잭션을 원하는 대로 정렬할 수 있습니다. 트랜잭션 순서는 사용자가 지불하는 네트워크 수수료(예: 이더리움(ETH)의 수수료 가격)에 따라 빨라지는 네트워크 연결을 활용하여 조작될 수 있습니다. 이러한 조작은 예컨대 선행매매(front-running)의 형태를 취할 수 있습니다. 채굴자인 전략적 행위자가 사용자 트랜잭션을 관찰하고 같은 블록의 이전 위치에 자신의 착취 트랜잭션을 삽입하면, 사용자 트랜잭션에 대한 사전 지식을 활용하여 금전을 효과적으로 갈취할 수 있습니다. 예를 들어, 봇은 사용자보다 먼저 구매 주문을 할 수 있습니다. 이후 사용자 거래로 인해 자산 가격이 상승할 때 이득을 취할 수 있습니다.

일반 사용자에게 해를 입히는 일부 봇에 의한 선행매매(월스트리트의 고빈도 거래와 유사)는 후행매매(back-running)[159] 및 자동 트랜잭션 모방[195]과 같은 관련 공격과 마찬가지로 이미 널리 알려져 있으며 문서화되어 잘 정리되어 있습니다[90]. 최근에는 채굴자들에 의한 순서 탈취를 체계화하자는 제안이 표면화되기도 했습니다[110].

롤업(roll-up)과 같은 레이어-2 기술은 이러한 문제를 해결하지 않지만, 순서를 다시 중앙화 (re-centralize)시켜 롤업을 생성하는 엔티티의 재량에 맡깁니다.

우리 목표 중 하나는 이른바 공정 시퀀싱 서비스(Fair Sequencing Services, FSS)[137]를 체인링크에 도입하는 것입니다. FSS는 스마트 컨트랙트 설계자가 트랜잭션에 대한 공정한 주문을 보장하고 오라클 보고서 전송과 같은 다른 유형의 트랜잭션뿐만 아니라 사용자 트랜잭션에 대한 선행, 후행 및 관련 공격을 방지하도록 지원합니다. FSS는 DON이 [144]에서 도입된 순서 공정성(order-fairness)에 대한 엄격한 시간적 개념과 같은 아이디어를 구현할 수 있습니다. 부수적으로 FSS는 사용자의 네트워크 요금(예: 수수료 비용)을 낮출 수 있습니다.

간단히 말하면, FSS에서 트랜잭션은 대상 스마트 컨트랙트로 직접 전파되지 않고 DON을 통과하게 됩니다. DON은 트랜잭션을 주문한 다음 컨트랙트로 전달합니다.



- 그림 6: FSS의 이점에 대한 예시. 그림에서는 트랜잭션을 주문하기 위해 중앙화된 권한을 활용하는 채굴자가 한 쌍의 트랜잭션을 스왑하는 방법을 보여줍니다. 여기서, 트랜잭션 ①은 ②보다 먼저 도착하지만 채굴자는 대신 ② 이후에 순서를 지정합니다. 이와 대조적으로, 그림에서는 DON이 DON 노드 간의 주문 프로세스를 어떻게 탈중앙화하는지 보여줍니다. 정직한 노드의 과반수가 ②보다①을 먼저

수신하면 FSS는 체인상에 ①이 ②보다 먼저 나타나도록 하며 컨트랙트를 실행할 수 있는 시퀀스 번호를 첨부하여 채굴자에 의한 재정렬을 방지합니다.

그림 6에서는 일반 채굴과 FSS를 비교합니다. 일반 채굴에서 트랜잭션 순서화 프로스가 채굴자에게 중앙화되어 도착 시간에 따라 한 쌍의 트랜잭션을 재정렬하는 등 조작 행위가 가능합니다. 대조적으로, FSS에서는 프로스가 DON 노드 간에 탈중앙화됩니다. 과반수가 정직한 노드라는 조건을 가정할 때 FSS는 트랜잭션의 시간적 순서 등 정책을 시행하는 데 도움이 되며 채굴자 및 기타 엔티티에 의한 조작의 기회를 줄입니다. 또한 사용자는 수수료 가격에 따른 우선 주문을 위해 경쟁할 필요가 없으므로 비교적 저렴한 수수료 가격을 지불할 수 있습니다. (DON의 트랜잭션은 함께 처리해 수수료를 절약할 수 있음)

신뢰의 최소화: DON 설계의 일반적 목표는 탈중앙화, 암호화 도구 및 암호화 경제 보장이라는 수단을 통해 스마트 컨트랙트 및 기타 오라클 종속 시스템에 대한 매우 신뢰할 수 있는 지원 레이어를 활성화하는 것입니다. DON 자체는 탈중앙화되어 있으며 사용자는 운영하려는 메인 체인을 지원하는 가용 DON 중에서 선택하거나 그들이 신뢰하는 노드 위원회와 함께 추가 DON을 생성할 수 있습니다.

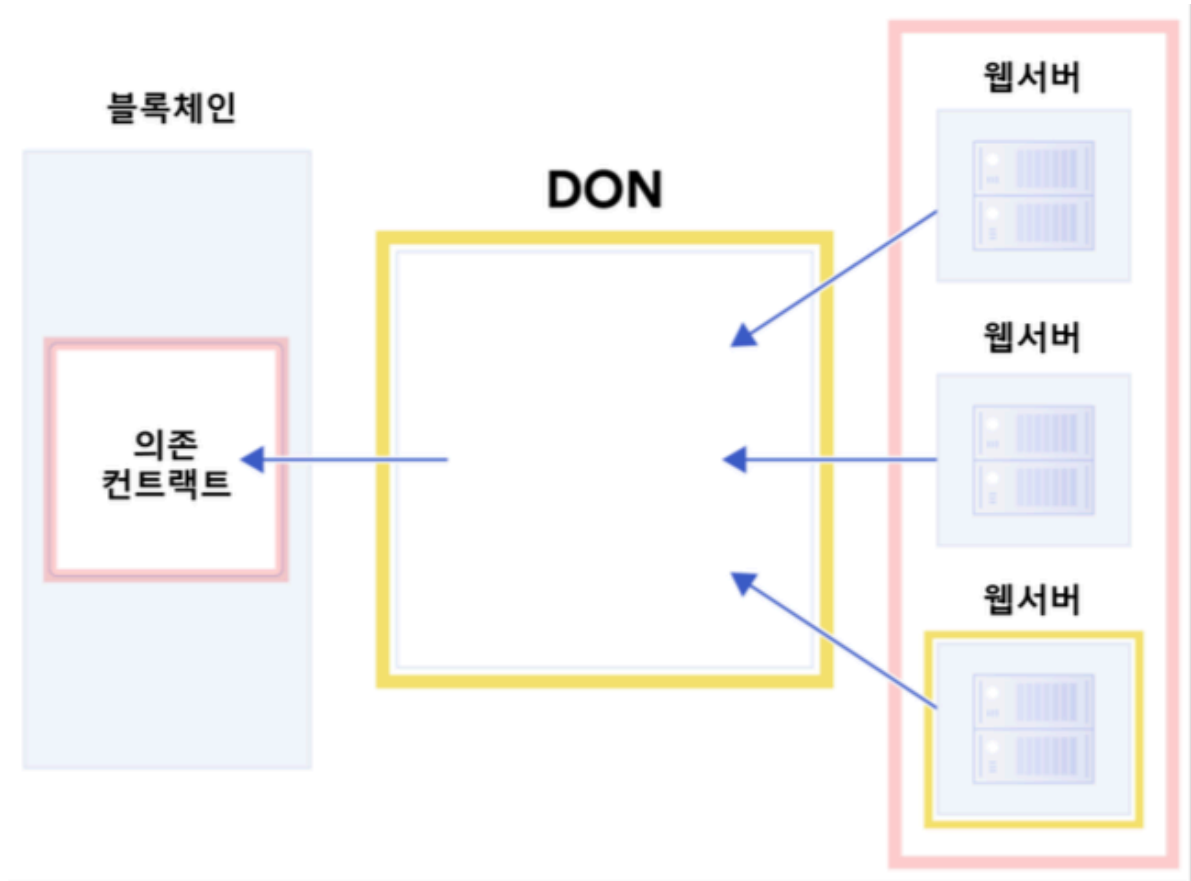
그러나 일부 애플리케이션, 특히 스마트 컨트랙트의 경우 체인링크 사용자는 DON이 지원하는 메인 체인에서 DON 자체보다 신뢰하는 신뢰 모델을 더 선호할 수 있습니다. 그러한 사용자를 위해 우리는 DON이 제공하는 보안 보증을 강화하고자 메인 체인 상에서 컨트랙트를 구현하는 많은 메커니즘을 체인링크 네트워크에 통합하였으며, 이와 동시에 DON이 데이터를 얻는 웹 서버 등의 데이터 소스가 손상될 가능성에 대비한 보호를 시행하고 있습니다.

이러한 메커니즘에 대해서는 섰 7에서 설명합니다. 이는 5가지 주제로 분류됩니다.

- 데이터 소스 인증 : 데이터 공급자가 데이터에 전자 서명을 할 수 있도록 하는 도구로 원본과 신뢰하는 컨트랙트 간의 관리 체인을 강화합니다.
- DON 마이너리티 리포트 : DON에서 다수의 불법 행위를 관찰하는 DON 노드의 소수 하위 집합에 의해 발행된 플래그입니다.
- 가드레일 : 비정상 조건을 감지하고 컨트랙트 실행을 일시 중지하거나 중단하는(또는 다른 교정 조치를 호출하는) 메인 체인 상 로직입니다.
- 신뢰 최소화 거버넌스 : 커뮤니티 조사를 쉽게 하는 점진적 릴리스 업데이트의 사용 및 시스템 장애에 대한 신속한 대응을 위한 탈중앙화된 비상 개입의 사용을 의미합니다.
- 탈중앙 엔티티 인증 : 체인링크 네트워크에서는 엔티티 식별을 위해 공개키 인프라(PKI)를 사용합니다.

그림 7에서는 신뢰 최소화 목표의 개념을 보여줍니다.

인센티브 기반(암호경제) 보안: 오라클 노드 전반에 걸친 보고서 생성 과정의 탈중앙화는 일부 노드가 손상된 경우에도 보안을 보장하는 데 도움이 됩니다.



- 그림 7: 체인링크의 신뢰 최소화 목표 - 웹 서버 등의 데이터 소스 및 DON의 올바른 동작에 대한 사용자의 요구를 최소화. 노란색 강조 표시는 신뢰 최소화 위치(DON 및 웹 서버의 개별 또는 소수 집합)를 나타냅니다. 분홍색은 블록체인과 대다수의 웹 서버 즉, 웹 서버 전체에 대한 컨트랙트와 같이 가정에 따라 매우 신뢰할 수 있는 시스템 구성요소를 나타냅니다.

이와 동등하게 중요한 것은 노드가 올바르게 행동할 재정적 인센티브를 갖도록 해야 한다는 것입니다. 예치(staking), 즉 노드가 LINK의 예치금을 제공하도록 요구하고 부정행위를 할 때 이 예치금을 삭감(몰수)하는 것은 체인링크에서 핵심적인 역할을 할 것입니다. 이는 이미 많은 블록체인에서 사용되는 중요한 인센티브 디자인입니다([81, 103, 120, 204] 참조).

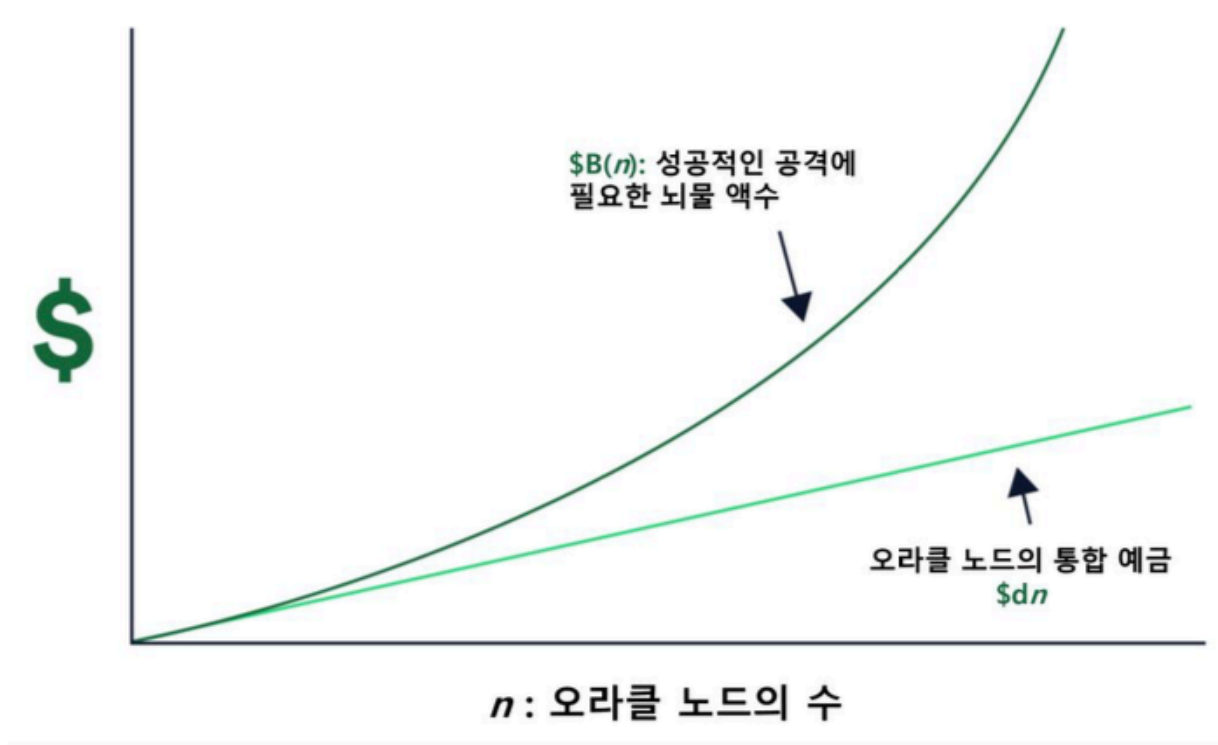
체인링크에서 예치는 독립형 블록체인에서 예치와 매우 다릅니다. 블록체인에서 예치는 컨센서스에 대한 공격 방지를 목표로 합니다. 체인링크의 경우 그 목표는 오라클 보고서를 적시에 제공하는 것입니다. 오라클 네트워크용으로 잘 설계된 예치 시스템은 높은 금전적 가치를 지닌 스마트 컨트랙트일지라도 뇌물과 같은 공격을 무용하게 만들 수 있어야 합니다.

이 백서에서는 다음 가지 주요 혁신을 통해 체인링크에서 예치를 위한 일반적인 접근 방식을 제시합니다.

1. 강력한 적대적 모델 : 기존 접근 방식에서 간과된 공격을 포괄하는 강력한 모델입니다. 우리가 잠재적 뇌물(prospective bribery)이라고 부르는 것이 대표적인 예입니다. 이는 조건부로 뇌물을 받는 노드를 결정하는 형태의 뇌물을 가리킵니다. 예를 들어, 예치 메커니즘이 특정 역할(예: 보고 결정)에 대해 무작위로 선택하는 노드에 미리 보장된 뇌물을 제공하는 방식입니다.
2. 초선형적 예치 영향 (Super-linear staking impact) : 간단하게 말하자면, 공격자가 성공하기 위해 모든 오라클 노드의 예치금을 합친 것보다 더 많은 예산($\$B$)이 있어야 합니다. 정확하게는 각각 고정된 예치금 $\$d$ 를 가진 n 개 오라클 노드 네트워크에서, n 의 함수가 $\$B(n) \gg \dn 이어야 함을 의

미합니다(보다 공식적으로, $\$B(n)$ 는 $\$dn$ 보다 n 에서 점근적으로 더 큼). 그림 8은 이 속성 개념도를 보여줍니다.

3. 암묵적 인센티브 프레임워크 (Implicit - Incentive Framework, IIF) : 노드의 미래 수수료 기회를 포함하여 명시적으로 예치된 예치 펀드에 경험적으로 측정 가능한 인센티브를 포함하도록 고안한 인센티브 모델입니다. IIF는 명시적 노드 예금을 초월하여 예치 개념을 가집니다.



- 그림 8: 체인링크 예치의 초선형적 확장 개념. 공격자가 요구하는 뇌물 $\$B(n)$ 는 모든 오라클 노드의 예금 $\$dn$ 를 합친 것보다 n 에 있어 더 빠르게 증가합니다.

우리는 IIF와 초선형적 예치 영향이 함께 오라클 네트워크에 대한 경제적 보안의 선순환이라는 것을 유도하는 방법을 보여줍니다. 새로운 사용자가 시스템에 진입해 체인링크 노드 실행에 따라 잠재적인 미래 수익이 증가할 때, 현재 및 미래 사용자의 경제적 보안의 한계 비용은 감소하게 됩니다. 탄력적인 수요 체제에서, 이러한 감소한 비용은 추가 사용자가 네트워크를 사용하도록 유도하며 선순환을 통해 채택을 지속시킵니다.

참고: 이 백서는 체인링크의 진화에 관한 우리의 비전에 대해 설명하지만 비공식적인 성격으로 기술적 부 사항은 거의 제시하지 않습니다. 추가 기능 및 접근 방식이 개발될 경우 전문적인 기술 문서가 발표될 예정입니다. 또한 여기에서 제시하는 비전의 많은 요소(확장 개선, 기밀성 기술, FSS 등)는 고급 DON이 체인링크의 기본 기능이 되기 전에도 예비로 배포될 예정이거나 배포될 가능성이 있음을 유념하시기 바랍니다.

1.3 이 문서의 구성

섹션 2에서는 보안 모델과 표기법을 제시하고 섹션 3에서는 탈중앙화 오라클 네트워크 API에 대해 간략히 설명합니다. 섹션 4에서는 DON이 매력적인 배포 플랫폼을 제공하는 여러 애플리케이션을 예시합니다. 여기까지 내용을 파악하면 이 논문의 주요 개념을 대부분 배울 수 있습니다.

나머지 섹션에서는 이와 관련한 자한 내용을 다룹니다. 섹션 5에서는 공정 시퀀싱 서비스(Fair Sequencing Services, FSS), 섹션 6에서는 트랜잭션 실행 프레임워크(Transaction-Execution Framework, TEF)에 대해 설명합니다. 섹션 7에서는 신뢰 최소화에 대한 접근 방식을 설명합니다. 섹션 8에서는 DON 배포와 관련한 몇 가지 중요 요구 사항(기능의 점진적 롤아웃, 동적 원장 멤버십 및 책임성)을 다룹니다. 마지막으로 섹션 9에서는 인센티브 설계에 대한 개발 접근 방식을 개괄합니다. 10장에서는 결론을 도출합니다.

이 문서의 개념에 익숙하지 않은 독자를 위해 부록 A에서는 용어집을 제공합니다. 부록 B에서는 DON 인터페이스 및 기능에 대한 자한 내용을 제시하며, 부록 C에서는 몇 가지 어댑터를 예시합니다. 부록 D에서는 기능적 서명, 즉 신뢰를 최소화한 데이터 소스 인증을 위한 암호 프리미티브를 설명하고, 이산화된 기능적 서명이라는 새로운 변형을 소개합니다. 부록 F에서는 DON에 대한 위원회 선택과 관련된 몇 가지 고려 사항을 논의합니다.

2. 보안 모델 및 목표

탈중앙화 오라클 네트워크(Decentralized Oracle Network, DON)는 이산형 탈중앙화 시스템으로 초기에는 일반적으로(필수적이지는 않음) 위원회 기반 컨센서스 프로토콜에 의해 구현되고 오라클 노드 집합으로 실행될 것으로 예상됩니다. DON은 주로 오라클 보고서 및 기타 서비스로 메인 체인의 스마트 컨트랙트 기능을 강화하도록 설계되었지만 다른 비블록체인 시스템에도 동일한 지원 서비스를 제공할 수 있어 특정한 메인 체인과 연관될 필요가 없습니다. 따라서 우리가 고려하는 모델과 속성은 DON의 특정 애플리케이션 사용과는 상당히 독립적인 성격을 갖습니다.

2.1 현재 아키텍처 모델

오늘날 체인링크는 일체식(monolithic) 서비스가 아닌 무허가(permissionless) 방식의 프레임 워크로, 오라클 노드의 고유하며 독립적인 네트워크를 시작할 수 있습니다[77]. 이러한 네트워크에는 이기종 노드 운영자 및 설계 트가 존재합니다. 또한 네트워크는 데이터 피드, 준비금 증명(Proof of Reserves), 검증

가능한 무작위성(verifiable randomness) 등을 포함하여 제공하는 서비스 유형에 따라 다양할 수 있습니다.

다른 점으로는 탈중앙화의 정도, 지원하는 네트워크가 보유한 가치 측면의 네트워크의 크기, 데이터 빈도 및 정확도와 같은 다양한 서비스 수준의 매개변수 등을 들 수 있습니다.

체인링크 무허가 모델은 제공자들이 커뮤니티에 가장 잘 제공할 수 있는 서비스를 전문화할 수 있는 생태계의 성장을 촉진합니다. 모든 노드와 네트워크에 전체 범위 서비스를 제공하도록 요구하는 다른 모델에 비해, 이 모델은 사용자에게 더 낮은 비용과 더 높은 서비스 품질을 제공할 가능성이 높습니다. 이는 노드에 제공되는 가용 리소스의 최소 공통분모를 갖는 서비스를 시스템 전반에서 채택하도록 쉽게 전환하는 접근 방법이 됩니다.

체인링크 2.0에서 체인링크는 DON 기반의 설계로 발전할 것이며, 우리는 사용자에게 특정 애플리케이션 요건에 가장 잘 부합할 다양한 서비스 선택권을 전 계적으로 제공한다는 목표를 위해 무허가 방식의 개방형 프레임워크 모델을 계속 지원할 것입니다.

2.2 컨센서스 가정

우리는 오라클 노드가 유지 관리하는 데이터 구조와 그 위에 레이어화 된 핵심 API를 모두 설명하는 오라클 시스템의 전체 기능을 포괄적으로 표현하기 위해 ‘탈중앙화 오라클 네트워크(DON)’라는 용어를 사용할 것입니다.

DON이 유지 관리하고 제공하는 특정 서비스의 지원에 사용되는 기본 데이터 구조를 원장 (ledger) 이라고 표현하며 L 로 표시합니다. DON 프레임워크는 L 을 블록체인과 같은 독립 시스템으로 취급하지 않는다는 점에 유념해야 합니다. 그 목적은 블록체인 및 기타 시스템을 지원하는 것입니다. 물론 블록체인은 신뢰할 수 있는 원장을 실현하는 한 가지 방법이지만 다른 방법도 존재합니다. 우리는 많은 경우 DON이 비트코인과 같은 블록체인보다 훨씬 앞선 비잔틴 장애 허용(Byzantine Fault Tolerant, BFT) 시스템을 사용하여 기본 원장을 실현할 것으로 기대합니다[174]. DON은 무허가 컨센서스 프로토콜을 사용해서도 실현될 수 있지만 편의를 위해 이 문서 전반에서는 BFT 형식의 표기와 속성을 사용합니다.

개념적으로 원장 L 은 데이터가 선형으로 정렬된 하나의 게시판입니다. 우리는 일반적으로 원장에는 흔히 블록체인에 일반적으로 귀속되는 몇 가지 주요 속성이 있다고 판단합니다[115]. 원장의 특성은 다음과 같습니다.

- 추가만 가능 : 한 번 추가된 데이터는 제거 또는 수정이 불가능합니다.
- 공개형 : 내용을 누구나 읽을 수 있습니다. 내용은 모든 사용자의 관점에서 시간의 경과에 따라 일관 됩니다.
- 가용성 : 원장은 항상 권한 있는 작성자가 쓸 수 있고 적시에 누구나 읽을 수 있습니다.

위원회라면 DON의 원장에서 속성 변경이 가능합니다. 예컨대, 원장 쓰기 액세스는 일부 애플리케이션에 대한 읽기 액세스와 같이 특정 사용자로 제한될 수 있습니다. 즉, 원장은 위에 정의된 대로 공개될 필요가 없습니다. 마찬가지로 원장 규칙은 데이터 수정 또는 교정을 허용할 수 있습니다. 그러나 이 백서에서는 이러한 변형에 대해 명시적으로 고려하지 않습니다.

DON 모듈식 설계는 다양한 최신 BFT 프로토콜(예: 핫스터프(Hotstuff)[231])을 지원할 수 있습니다. 정확한 선택은 오라클 노드 간 신뢰 가정과 네트워크 특성에 따라 달라집니다. 원칙적으로 DON은 동등하게 확장 가능한 레이어-2 또는 블록체인 시스템을 지원하는 원장 역할과 관련하여 고성능 무허가 블록체인을 원장의 대안으로 사용할 수 있습니다. 이와 유사하게, 하이브리드화도 가능합니다. DON은 원칙상 기존 블록체인의 유효성 검사기에 해당하는 노드로 구성될 수 있습니다(예: 트랜잭션을 실행하기 위해 위원회가 선택되는 지분 증명(Proof-of-Stake) 시스템, [8, 81, 120, 146, 204] 참조). 이 특정한 작동 모드에서는 노드가 이중 모드 즉, 블록체인 노드와 DON 노드로 작동해야 합니다. (위원회 변경의 연속성을 보장하는 기술에 대한 논의는 섹션 8.2를, 무작위 위원회 선택에 대한 주의 사항은 부록 F를 참조.)

실제로, 최신 BFT 알고리즘에서 노드는 원장 위 메시지에 디지털 방식으로 서명합니다. 편의상 L 에 연결된 공개키(pk_L)가 있으며 해당 콘텐츠가 해당 비밀키로 서명되어 있다고 가정합니다. 이 일반 표기법은 임계값 서명을 사용하여 L 상 데이터가 서명된 경우에도 적용됩니다. 임계값 서명은 DON을 실행하는 노드의 구성원이 변경되더라도 DON에 대한 영구 ID를 활성화하므로 편리합니다 (부록 B.1.3 참조). 따라서 우리는 sk_L 가 일부 보안 매개변수 k 에 대해 (k, n) 임계값 방식으로 비밀을 공유한다고 가정합니다. 예를 들면 $k = 2f + 1$

및 $n = 3f + 1$ 등과 같으며, 여기서 f 는 잠재적 결함이 있는 노드의 개수를 나타냅니다. (이러한 방식에 따라 k 의 선택을 수행하면, 결함 있는 노드는 skL 에 대한 학습을 할 수도, 서비스 거부(DoS) 공격을 탑재할 수도 없게 됩니다).

L 상의 메시지는 $M = (m, z)$ 의 형식을 취합니다. 여기서 m 는 문자열이며 z 는 고유 순차 색인 번호입니다. 해당되는 경우, 우리는 $m = \langle \text{MessageType} : \text{payload} \rangle$ 의 형식으로 메시지를 작성합니다. *MessageType* 메시지 유형은 특정 메시지 기능을 나타내는 간결한 표현(syntactic sugar)입니다.

2.3 표기법

원장을 실행하는 n 개의 오라클 노드 집합은 $O = \{o_i\}_{i=1}^n$ 의 식으로 표현합니다. 이러한 노드 집합은 때때로 **위원회(committee)**라 부릅니다. 단순화를 위해, DON 기능을 구현하는 오라클 세트 (즉, SL 위에 있는 서비스 노드의 유지를 담당하는 세트)로 동일하지만 이 둘의 구별이 가능하다고 가정합니다. 플레이어 o_i 의 공개키를 pk_i , 해당 비밀키는 sk_i 로 표현합니다.

대부분 BFT 알고리즘에서는 최소한 $n = 3f + 1$ 개의 노드가 필요합니다. 여기서 f 는 잠재적으로 결함이 있는 노드 수입니다. 나머지 노드는 지정된 대로 정확하게 프로토콜을 따른다는 전제에서 정직합니다. 위원회가 이 요구 사항을 충족할 경우, 즉 정직한 노드의 $2/3$ 이상 비율을 가진 경우 정직한 위원이라고 간주합니다. 달리 명시되지 않는 한 이는 정직하다고 (그리고 부정직의 정직 모델링이라고) 가정합니다. 맥락에 따라 pk_0 / sk_0 은 pk_1 / sk_1 와 혼용합니다.

$\sigma = \text{Sig}_{pk}[m]$ 의 식은 pk 에 관한 (즉, 해당 비밀키 sk 를 사용한) 메시지 m 서명을 나타냅니다. $\text{verify}(pk, \sigma, m) \rightarrow \{false, true\}$ 는 해당 서명 확인 알고리즘을 표현합니다. (본 백서에서 키 생성은 암묵적으로 처리되는 것으로 여깁니다.)

S 는 데이터 소스를 뜻하며 노는 주어진 컨텍스트 상 전체 n_s 개 소스 세트를 나타냅니다. DON이 지원하는 스마트 컨트랙트 기능 블록체인은 **MAINCHAIN**으로 표기합니다. DON과 통신하는 MAINCHAIN의 모든 스마트 컨트랙트는 의존 컨트랙트(relying contract)라는 용어로 표현하고, SC_2 로 표기합니다.

우리는 일반적으로 DON이 단일 메인 체인인 MAINCHAIN을 지원한다고 가정하지만 섹션 4의 예시와 같이 여러 체인을 지원할 수 있습니다. DON은 MAINCHAIN에서 여러 개의 의존 컨트랙트를 지원할 수 있으며 일반적으로 그렇게 될 것입니다. (앞서 언급했듯이 DON은 블록체인이 아닌 서비스를 대안으로 지원할 수 있습니다.)

2.4 신뢰 모델에 대한 참고 사항

앞서 언급한 바와 같이 DON은 위원회 기반 컨센서스 프로토콜을 기반으로 구축될 수 있으며 일반적으로 이러한 프로토콜을 사용할 것으로 예상합니다. 위원회 기반이나 무허가 블록체인의 두 가지 대표 중 하나가 다른 것보다 더 강력한 보안을 제공한다는 설득력 있는 주장이 존재하기 때문입니다.

중요한 것은 위원회 기반 시스템과 무허가 탈중앙 시스템 보안은 비교가 불가능하다는 사실입니다. 51% 공격을 통해 작업증명(PoW) 또는 지분증명(PoS) 블록체인을 손상시키려면 (예를 들면, 공격자가 PoW 시스템에서 해시 파워를 대량 소유하는 것처럼) 일시적, 잠재적으로 대다수 리소스를 막대한 요금으로 확보해야 합니다. 실제로 이러한 공격은 이미 여러 블록체인에 영향을 미친 바 있습니다[200, 34]. 이와는 대조적으로, 위원회 기반 시스템을 손상시키겠다는 것은 잘 알려지지, 풍부한 자원과 신뢰도를 가진 노드들의 임계수(일반적으로 1/3)를 손상시키는 것을 의미합니다.

반면 위원회 기반 시스템(위원회를 지원하는 ‘하이브리드’ 무허가 시스템 포함)은 절대적으로 무허가 시스템보다 더 많은 기능을 지원할 수 있습니다. 여기에는 서명이나 암호화키와 같은 영구적인 비밀을 유지하는 기능이 포함됩니다.

우리는 원칙상 DON이 위원회 기반이나 무허가 컨센서스 프로토콜을 위에 구축될 수 있으며, DON 대표자가 궁극적으로 두 접근 방식을 채택할 수 있다는 점을 강조하고자 합니다.

신뢰 모델의 강화: 현재 체인링크의 핵심 기능은 섹션 3.6.4에서 논의된 것처럼 사용자가 성능 내역의 탈중앙화된 기록을 기반으로 노드를 선택할 수 있도록 하는 기능입니다. 섹션 4에서 소개할 예제 메커니즘과 암묵적 인센티브 프레임워크는 합리 공평하며 엄격한 메커니즘 설계 프레임워크를 구성하며, 사용자에게 크게 확장된 능력을 부여하여 DON의 보안을 측정할 수 있게 합니다. 이 동일한 프레임워크는 DON 자체가 참여 도메인에 다양한 보안 요구 사항을 적용하고 강력한 신뢰 모델 내의 운영을 보장할 수 있도록 할 것입니다.

DON과 관련하여 이 백서에 설명된 도구를 사용하면 규제 요건 준수와 같은 특별한 신뢰 모델 요건을 시행할 수도 있습니다. 예를 들어, 섹션 4.3에서 논의된 기술을 사용할 경우, 노드는 일반 데이터 보호 규정 (General Data Protection Regulation, GDPR) 3조(‘영역 범위’)의 준수를 집행하는 데 사용할 수 있는 노드 운영자 특정(예: 영국 영역)의 증거를 제시할 수 있습니다[105]. 그렇지 않을 경우에는 탈중앙 시스템에서 이러한 규정 준수를 충족하기가 어려울 수 있습니다[45].

추가적으로, 섹션 7에서는 DON이 지원하는 메인 체인에서 신뢰 최소화 메커니즘을 통해 DON의 견고성을 강화하기 위한 계획을 논의합니다.

3. 탈중앙화 오라클 네트워크 인터페이스 및 기능

여기에서는 DON이 실현하도록 설계된, 단순하지만 강력한 인터페이스의 측면에서 DON의 기능을 간략하게 설명합니다.

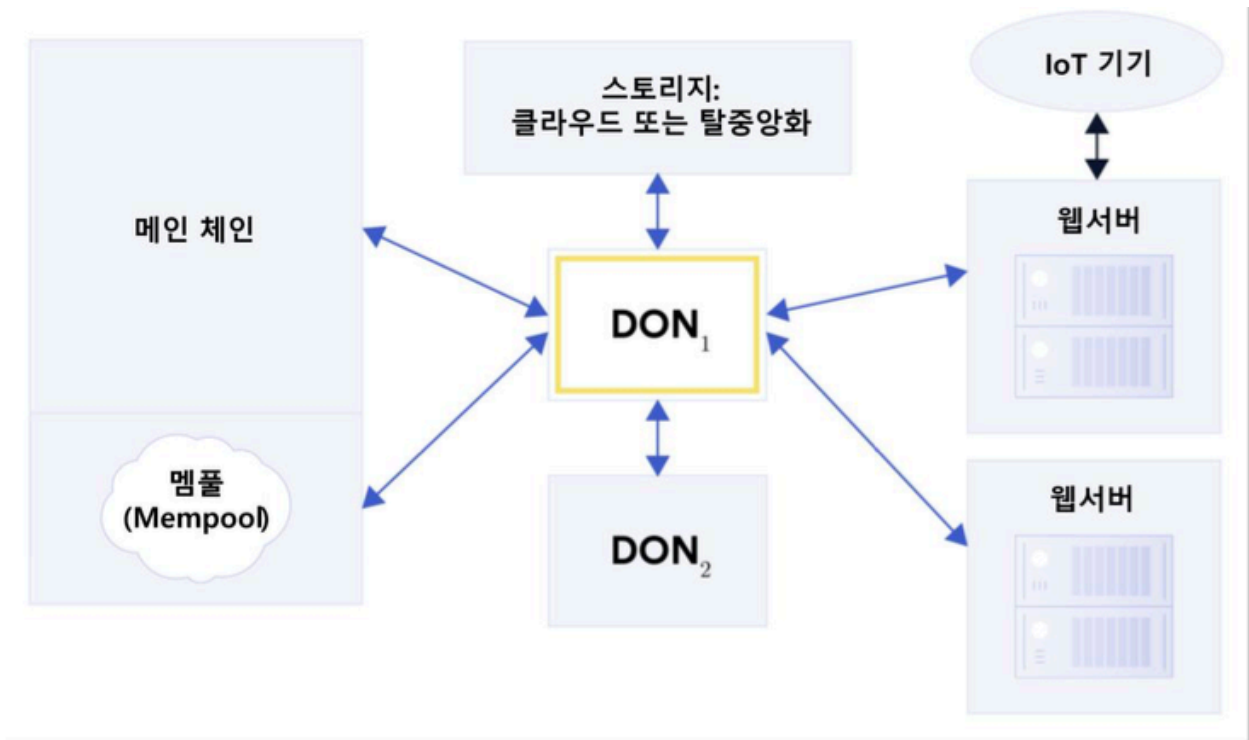
DON 상 애플리케이션은 **실행파일(executables)** 과 **어댑터(adapters)** 로 구성됩니다. 실행파일은 프로그램으로 그 핵심 로직은 스마트 컨트랙트와 유사한 결정적 프로그램(deterministic program)입니다. 실행파일에는 다수의 **개시자(initiator)** 가 수반됩니다. 이는 사전에 결정된 이벤트(크론(cron) 작업 등)가 발생할 때, 즉 가격이 어떤 임계 등을 초월할 경우, 해당 실행파일의 로직에서 진입점을 호출하는 프로그램입니다(키퍼(Keepers)와 상당히 유사함, 섹션 3.6.3 참조). 어댑터는 오프체인 리소스에 대한 인터페

이스를 제공하며 실행파일 개시자나 코어 로직에 의해 호출될 수 있습니다. 어댑터의 동작은 외부 리소스 동작에 따라 달라질 수 있으므로, 개시자와 어댑터는 비결정적으로 동작한다고 볼 수 있습니다.

여기에서는 컴퓨팅 시스템 특성과에 일반적으로 사용되는 세 가지 리소스인 네트워킹, 컴퓨팅, 스토리지의 측면에서 DON 개발자 인터페이스와 실행파일 및 어댑터 기능을 설명하겠습니다. 아래에서는 이러한 각 리소스를 개괄하고 부록 B에서는 그 자세한 내용을 다룹니다.

3.1 네트워킹

어댑터는 인터페이스로서 DON에서 실행되는 실행파일은 이를 통해 DON 외부 시스템과 데이터를 송신합니다. 어댑터는 오늘날 체인링크에서 사용되는 **어댑터**의 일반형으로 간주할 수 있습니다[20]. 어댑터는 양방향일 수 있습니다. 즉, DON에서 웹 서버로 데이터를 가져올 뿐만(pull) 아니라 보낼(푸시, push) 수도 있습니다. 더불어 안전한 다자 간 연산과 같은 암호화 기능뿐만 아니라 탈중앙화된 프로토콜을 활용할 수도 있습니다.



- 그림 9: \$DON_2\$로 표시되는 DON, 블록체인(메인 체인) 및 해당 멤풀, 외부 스토리지, 웹 서버 및 IoT 장치(웹 서버를 통합)를 포함한 DON(\$DON_1\$로 표시)을 연결하는 어댑터

그림 9에서는 어댑터가 생성될 수 있는 외부 리소스를 예시합니다. 여기에는 다음이 포함됩니다.

- 블록체인: 어댑터는 트랜잭션을 블록체인으로 보내는 방법과 블록, 개별 트랜잭션 또는 기타 상태를 읽는 방법을 정의할 수 있습니다. 어댑터는 블록체인의 멤풀에 대해 정의될 수도 있습니다. (섹션 3.5 참조.)
- 웹 서버: 어댑터는 DON 인터페이스용이 아닌 레거시 시스템을 포함, 웹 서버로부터 데이터를 검색할 수 있는 API를 정의할 수 있습니다. 이러한 어댑터에는 서버에 데이터를 보내는 API도 포함될 수 있습니다. DON이 연결되는 웹 서버는 사물 인터넷(IoT) 장치와 같은 추가 리소스에 대한 게이트웨이

역할을 할 수 있습니다.

- 외부 스토리지: 어댑터는 탈중앙 파일 시스템[40, 188] 또는 클라우드 스토리지와 같은 DON 외부 스토리지 서비스를 읽고 쓰는 방법을 정의할 수 있습니다.
- 기타 DON: 어댑터는 DON 간에 데이터를 검색하고 전송할 수 있습니다.

DON의 초기 배에는 일반적으로 사용되는 외부 리소스에 대한 **빌딩 블록 어댑터** 세트가 포함되며 DON 노드에서 DON 특정 어댑터를 게시할 수 있습니다. 이제 스마트 컨트랙트 개발자가 어댑터를 작성하게 되면, 이러한 고급 기능을 사용하여 훨씬 더 강력한 어댑터를 구축할 수 있을 것으로 기대됩니다.

궁극적으로는 모든 사용자가 무허가 방식으로 새 어댑터를 생성할 수 있을 것으로 기대됩니다.

일부 어댑터는 DON에 의해 제어되는 외부 리소스의 **지속성과 가용성**을 보장하는 방식으로 구성되어야 합니다. 예컨대, 클라우드 스토리지는 클라우드 서비스 계정의 유지 관리를 필요할 수 있습니다. 또한 DON은 사용자가[160] 참조 및 실행파일을 대신하여 비밀키 탈중앙 관리를 수행할 수 있습니다. 결과적으로 DON은, 대상 블록체인에서 트랜잭션을 보내는 데 사용할 수 있는 디지털 자산의 리소스를 제어하는 일 등이 가능합니다.

몇 가지 예시 어댑터에 대한 자세한 내용은 부록 B.1을, 어댑터 예시는 부록 C를 참고합니다.

3.2 연산

실행파일은 DON의 기본 코드 단위이며 $\$exec = (logic, init)\$$ 의 한 쌍입니다. 여기에서 $\$logic\$$ 은 다수 지정된 진입점($\$logic_1, logic_2, \dots, logic_n\$$)이 있는 결정적 프로그램이며 $\$init\$$ 는 해당 개시자 $\$(init_1, init_2, \dots, init_i)\$$ 집합입니다. DON의 완전한 감시 가능성을 보장하고자 하는 목적으로 실행파일의 로직은 모든 입력 및 출력력에 기본 원장 $\$L\$$ 을 사용합니다. 예컨대 실행파일에 입력되는 모든 어댑터 데이터는 먼저 $\$L\$$ 에 저장되어야 합니다.

개시자(Initiator): 현재 체인링크 개시자는 체인링크 노드에서 이벤트 종속 작업 실행을 일으킵니다[21]. DON의 개시자는 거의 동일한 방식으로 작동합니다. 그러나 DON 개시자는 구체적으로 실행파일과 관련 있습니다. 개시자는 **외부 이벤트의 상태, 현재 시각 또는 DON 상태의 술어(predicate)**에 따라 달라질 수 있습니다. 물론 이벤트에 종속을 갖기 때문에 개시자는 비결정적으로 동작할 수 있습니다(어댑터도 마찬가지임). 개시자는 개별 DON 노드 내에서 실행할 수 있으므로 어댑터에 의존할 필요가 없습니다(아래의 예시 1 참조).

개시자는 실행파일을 스마트 컨트랙트와 구별하는 중요한 기능입니다. 실행파일은 개시자에 대한 응답으로 실행될 수 있으므로 효과적으로 자동 작동할 수 있습니다. 물론 확장을 통해 실행파일을 통한 하이브리드 컨트랙트도 가능합니다. 현재 개시자 중 하나는 트랜잭션 자동화 서비스를 제공하는 체인링크 키퍼(Keeper)인데, 오라클 보고서를 기반으로 담보 대출 청산 및 제한 순서 거래 실행과 같은 스마트 컨트랙트 실행을 트리거합니다.

DON의 개시자는 DON이 호출해야 하는 상황을 정의한다는 점에서 볼 때, 편의 차원에서 실행파일에 적용될 서비스 계약을 지정하는 방법으로 간주할 수도 있습니다.

다음 예시에서 실행파일 및 개시자가 작동하는 방식을 보여줍니다.

예시 1 (편차로 유발된 거래 피드). 스마트 컨트랙트 SC는 하나의 자산 쌍(예: ETH-USD) 사이의 환율에 상당 변동(예: 1%)이 있을 때마다 새로운 가격 피드 데이터(섹션 3.6.3 참조)를 요구할 수 있습니다. 변동성에 민감한 가격 피드는 현재 체인링크에서 지원되고 있지만 실행파일 \$exec_{\{feed\}}\$를 통해 DON에서 어떻게 실행 가능한지 확인하는 것도 유용할 것입니다.

실행파일 \$exec_{\{feed\}}\$는 $\langle \text{NewPrice} : i, r \rangle$ 개 항목 시퀀스 형태로 \$L\$에서 가장 최근 ETH-USD 가격 \$r\$을 유지합니다. 여기서 i 는 가격 업데이트에 따라 증가하는 지수입니다.

개시자 \$init_1\$은 각 노드 \$o_i\$가 지수를 사용하여 가장 최근에 저장된 가격 r 로부터의 최소 1%의 편차에 대해 현재 ETH-USD 가격을 모니터링하도록 유도합니다. 이러한 편차가 감지되면 이는 새 가격에 대한 현재 보기 r' 라는 형식 $\langle \text{PriceView} : i, j, r' \rangle$ 의 한 항목을 사용하여 \$L\$에 씁니다.

고유 노드에 의해 생성된, 지수 $i + 1$ 에 대한 새로운 가격을 가진 최소 k 개 PriceView 항목이 L 에 누적될 경우 두 번째 개시자 \$init_2\$가 실행됩니다. \$init_2\$는 집입점 \$logic_2\$로 하여금 첫 번째부터 k 번째까지의 중앙값 r (유효한 PriceView 값)을 새로 연산한 다음, 새 값 $\langle \text{NewPrice} : j + 1, r \rangle$ 을 \$L\$에 씁니다. (운영상, 노드는 지정된 작성자로 교대 작동할 수 있음)

세 번째 개시자 \$init_3\$은 L 의 NewPrice 항목을 감시합니다. 새 보고서 $\langle \text{NewPrice} : j, r \rangle$ 가 나타날 때마다 어댑터를 사용하여 \$SC_2(j, r)\$을 푸시하는 진입점 \$logic_3\$을 호출합니다.

앞서 언급한 바와 같이, 실행파일은 기능 측면에서 스마트 컨트랙트와 유사합니다. 일반적인 메인 체인 컨트랙트보다 더욱 높은 성능을 제공할 뿐만 아니라 두 가지 본질적인 면에서 다릅니다.

- 1. 기밀성:** 실행파일은 기밀 연산을 수행할 수 있습니다. 즉, 비밀 프로그램이 일반 텍스트 입력을 처리하거나 공개된 프로그램이 비밀 입력 데이터를 처리할 수 있으며, 이 두 조합을 처리할 수도 있습니다. 단순한 모델에서 DON 노드는 비밀 데이터에 액세스할 수 있으며, 노드는 중간 결과를 숨기고 처리 및 삭제된 값만을 MAINCHAIN에 공개합니다. DON 자체로부터 민감 데이터를 숨기는 것도 가능합니다. DON은 이러한 목적을 위해 다자간 연산(multi-party computation)([42, 157]) 및 신뢰할 수 있는 실행 환경(trusted execution environments, TEE)(84, 133, 152, 229)과 같은 접근 방식을 지원합니다.¹⁶
- 2. 지원 역할:** 실행파일은 스마트 컨트랙트를 대체하기보다는 메인 체인에서 스마트 컨트랙트를 지원하는 데 의의가 있습니다. 실행파일에는 스마트 컨트랙트에 없는 몇 가지 제한 사항이 있습니다.
 - (a) 신뢰 모델:** 실행파일은 DON이 정의한 신뢰 모델 내에서 작동합니다. 올바른 실행은 이 정의된 행동에 달려 있습니다. (물론, 메인 체인 의존 실행은 섹션 7.3에서 논의된 것처럼 DON 불법 행위에 대한 일부 가드레일을 제공할 수 있습니다.)
 - (b) 자산 액세스:** DON은 블록체인 계정을 제어할 수 있으므로 어댑터를 통해 해당 계정의 자산을 제어할 수 있습니다. 그러나 DON은 기본 체인(예: 블록 또는 ERC-20 토큰)에서 생성된 자산을 권위 있는 방식으로 표현할 수 없습니다. 기본 체인의 소유권에 대한 권위 있는 기록을 유지하기 때문입니다.
 - (c) 수명 주기:** DON과 의존 컨트랙트의 소유자 사이에 존재하는 온체인 서비스 수준 계약에 정의된 바에 따라 의도적으로 제한된 수명을 가질 수 있습니다. 이와 대조적으로 블록체인은 영구 보관 시스템으로 가능합니다.

DON 연산에 대한 자세한 내용은 부록 B.2를 참조합니다.

3.3 스토리지

위원회 기반 시스템인 DON은 무허가 블록체인보다 훨씬 저렴한 비용으로 적당한 양의 데이터를 \$L\$에 지속적으로 저장할 수 있습니다. 또한 DON은 어댑터를 통해 데이터 저장을 위한 외부 탈중앙 시스템(예: 파일 코인(Filecoin)[85])을 참조할 수 있으며, 시스템을 스마트 컨트랙트와 연결할 수 있습니다. 이러한 옵션은 블록체인 시스템에 만연한 "팽창(Bloat)" 문제를 해결하는 수단으로써 특히 대량 데이터에 적합합니다.

DON은 지원되는 서비스에서 사용하기 위해 특별히 로컬 또는 외부에 데이터를 저장할 수 있습니다. DON은 기밀성을 유지하는 방식으로 데이터를 추가로 사용할 수 있으며, (1) **DON 노드 간에 공유되는 비밀**, 안전한 다자간 연산 및 부분적/완전 동형 암호화에 적합한 방식으로 DON 노드가 관리하는 키로 암호화된 데이터, 또는 (2) **신뢰할 수 있는 실행 환경을 사용하여 보호되는 데이터**를 연산할 수 있습니다.

우리는 DON이 스마트 컨트랙트 시스템에 공통적인 간단한 메모리 관리 모델을 채택할 것으로 기대합니다. 실행파일은 자체 메모리레벨을 쓸 수 있지만, 다른 실행파일의 메모리를 읽을 수 있습니다.

DON 저장에 대한 자세한 내용은 부록 B.3을 참조합니다.

3.4 트랜잭션-실행 프레임워크(TEF)

DON은 단일 메인 체인 MAINCHAIN(또는 여러 메인 체인)에서 컨트랙트를 지원하기 위한 것입니다. 섹션 6에서 자세히 논의할 **트랜잭션 실행 프레임워크(Transaction-Execution Framework, TEF)**는 범용 접근 방식을 MAINCHAIN과 DON에서 컨트랙트를 효율적으로 실행합니다. TEF는 FSS와 레이어-2 기술을 (필요시 동시적으로) 지원하기 위한 것입니다. 이는 실제로 FSS 사용에 주요 수단으로 작용할 가능성이 높습니다(이에 따라 이 섹션에서는 FSS에 대해 더 이상 논의하지 않음).

간단히 설명하면, TEF에서는 MAINCHAIN을 위해 설계되거나 개발된 원래의 대상 컨트랙트 \$SC\$는 하이브리드 컨트랙트로 리팩토링(refactoring)됩니다. 이러한 리팩토링된 하이브리드 컨트랙트는 6가지 상호 운용 부분인 MAINCHAIN 컨트랙트 \$SC_a\$ (본 백서에서는 TEF의 차원 명확성을 위해 앵커 컨트랙트라고 함) 및 DON의 실행파일 \$exec_s\$를 생성합니다. 컨트랙트 \$SC_a\$는 사용자가 자산을 관리하고, 권한 있는 상태 전환을 실행하며, DON의 실패에 대한 가드레일(섹션 7.3 참조도 제공합니다. 실행파일 \$exec_s\$는 트랜잭션 순서를 지정하고 관련된 오라클 데이터를 이를 트랜잭션에 제공합니다. \$SC_a\$를 위해 여러 가지 방법으로 트랜잭션을 묶을 수도 있습니다(예: 유효성 증명 기반 또는 낙관적 롤업(optimistic rollup) 사용, DON의 기밀 실행 등).

우리는 개발자들이 고급 언어로 쓰인 컨트랙트 \$SC\$를 MAINCHAIN 및 DON 논리 조각(각각 \$SC_a\$ 및 \$exec_s\$)으로 쉽게 분해하여 안전하고 효율적인 작성이 가능한 도구를 개발할 수 있을 것으로 기대합니다.

TEF를 사용하여 고성능 트랜잭션 체계를 고성능 오라클과 통합하는 것은 오라클 확장 접근 방식에 필수적입니다.

3.5 뎀풀(Mempool) 서비스

뎀풀 서비스(Mempool service, MS)는 FSS 및 TEF를 지원하기 위해, 우리가 DON에 배포하려는 중요한 애플리케이션 레이어 기능입니다. MS는 최상의 지원을 받는 어댑터로 볼 수 있습니다.

MS는 레거시 호환 트랜잭션 처리 지원을 제공합니다. 이 용도로 MS는 메인 체인의 뎀풀에서 MAINCHAIN 대상 컨트랙트 \$SC\$를 위한 트랜잭션을 수집합니다. 그런 다음 MS는 이러한 트랜잭션을 DON 상의 실행파일로 전달하며, 트랜잭션은 요구된 방식으로 처리됩니다. MS 데이터는 DON이 트랜잭션을 구성하는 데 사용할 수 있으며 DON에서 \$SC_2\$ 또는 \$SC_c\$를 호출하는 다른 컨트랙트로 직접 전달할 수 있습니다. 예를 들어 DON은 MS를 통해 수집된 트랜잭션을 전달하거나 MS 데이터를 사용하여 MAINCHAIN으로 보내는 트랜잭션에 대한 수수료 가격을 설정할 수 있습니다.

MS는 뎀풀을 모니터링하기 때문에 \$SC_c\$와 상호작용하는 사용자로부터 트랜잭션을 얻을 수 있습니다. 따라서 사용자는 레거시 소프트웨어, 즉 MS 및 MS 구성 컨트랙트의 존재를 인식하지 못하는 애플리케이션을 사용하여 트랜잭션을 계속 생성할 수 있습니다 (이 경우 \$SC_c\$는 원래 트랜잭션을 무시하고 MS에서 처리한 것만 수락하도록 변경되어야 이중 처리를 방지할 수 있음).

대상 컨트랙트 \$SC\$와 함께 사용하기 위해 MS는 FSS 및/또는 TEF와 함께 사용될 수 있습니다.

3.6 발판: 기존 체인링크 기능

3.6.1 오프체인 보고(OCR)

오프체인 보고(Off-Chain Reporting, OCR)[60]는 오라클 보고서 집계 및 의존 컨트랙트 \$SC_c\$로의 전송을 위한 체인링크 상의 메커니즘입니다. 이는 최근에 체인링크 가격 피드 네트워크에 배포되었으며 완전한 DON으로 가는 첫 번째 단계라 할 수 있습니다.

본질적으로 OCR은 부분 동기식 네트워크에서 작동하도록 설계된 BFT 프로토콜입니다. 이는 임의의 결함이 있는 노드가 $f < n/3$ 개 존재할 때 활성(liveness)과 정확성(correctness)을 보장하며 비잔틴 신뢰 브로드캐스트(Byzantine reliable broadcast)의 속성을 보장하고 완전한 컨센서스 프로토콜을 의미합니다. 노드는 모든 관점에서 동일한 원장을 대표한다는 맥락에서 볼 때, 일관적인 메시지 로그를 유지하지 않으며 프로토콜이 리더는 안전 수칙을 위반하지 않고도 모호하게 할 수 있습니다.

OCR은 현재 참여 노드에서 보고한 값과 중앙집중식 집계(최소 $2f+1$)라는 특정 메시지 유형을 위해 설계되어 있습니다. OCR은 자신이 \$SC\$에 대해 출력하는 보고서에 대한 핵심적인 보증을 제공합니다. 이를 **증명된 보고서(attested reports)**라 합니다. 증명된 보고서 중앙값은 정직한 두 노드에서 보고한 값과 같거나 그 사이에 존재하게 됩니다. 이 속성은 OCR이 주요한 안전성 조건이 됩니다. 리더는 증명된 보고서의 중앙값에 영향을 미칠 수 있지만 이 정책상 조건에 국속됩니다. OCR은 다양한 방식으로 값을 집계하는 메시지 유형으로 확장될 수 있습니다.

오늘날 체인링크 네트워크 활성 및 정확성 목표는 OCR이 완전한 컨센서스 프로토콜을 덜 것일 요구하지 않지만, OCR이 기존 BFT 프로토콜에 없는 다음과 같은 몇 가지 추가 기능 형태를 제공할 것을 요구합니다.

1. **전부 아니면 전무 방식 오프체인 보고서 브로드캐스트:** OCR은 모든 정직한 노드가 인증된 보고서를 신속하게 사용할 수 있도록 합니다. 이것은 정직한 노드가 증명된 보고서 전송에 참여할 기회를 갖도록 하는 공정성 속성입니다.
2. **신뢰할 수 있는 트랜스미션:** OCR은 결함이 있거나 악의적인 노드가 있는 경우에도 모든 OCR 보고서 및 메시지가 사전에 정의된 특정 시간 간격 내에 \$SC\$로 전송되도록 합니다. 이것은 활성의 속성입니다.
3. **컨트랙트 기반 신뢰 최소화:** \$SC\$는 잠재적으로 잘못된 OCR 생성 보고서(보고된 값이 최근에 수신된 다른 값과 크게 다른 경우)를 필터링합니다. 이것은 프로토콜 외 정확성(extra-protocol correctness) 시행의 한 형태입니다.

이 세 가지 속성은 모두 DON 내에서 각각의 역할을 수행합니다. DON(전부 또는 전무 방식의 오프체인) 브로드캐스트는 안정적 전송과 관련된 암호 경제학적 보장을 위한 중요한 빌딩 블록이며, 결과적으로 필요한 어댑터 속성이 됩니다. \$SC\$의 신뢰 최소화는 섹션 7.3에서 논의하게 될 일정의 가드레일입니다.

OCR은 또한 체인링크의 오라클 네트워크에서 BFT 프로토콜의 운용 배치 및 개선을 위한 기반을 제공하므로 위에서 언급한 바와 같이 DON의 전체 기능에 대한 경로를 제공합니다.

3.6.2 DECO와 타운크라이어

DECO[234]와 타운크라이어(Town Crier)[233]는 현재 체인링크 네트워크에서 개발 중인 한 쌍의 관련 기술입니다.

오늘날 대부분의 웹 서버에서는 사용자가 전송 레이어 보안(Transport Layer Security, TLS)[94]라는 프로토콜을 사용하여 보안 채널을 통해 연결할 수 있게 해줍니다. (HTTPS는 TLS가 활성화된 HTTP의 변형형을 나타냅니다. 즉, 'https'가 접두사인 URL은 보안을 위해 TLS가 사용 중임을 의미합니다.) 대부분 TLS 지원 서버에는 데이터에 전자 서명을 하지 않는다는 제한이 존재합니다. 결과적으로 사용자 또는 증명자(Prover)는 서버에서 받은 데이터를 해당 데이터 신뢰성을 보장하는 방식으로 오라클이나 스마트 컨트랙트와 같은 제3자 또는 검증자(Verifier)에게 제시할 수 없습니다.

서버가 데이터에 디지털 서명을 한다고 해도 기밀성의 문제가 여전히 남습니다. 증명자는 검증자에게 제공하기 전에 민감한 데이터를 변경 또는 수정하고자 할 수 있습니다. 그러나 디지털 서명은 수정된 데이터를 무효화하도록 특별히 설계됩니다. 따라서 증명자는 데이터에 기밀성을 유지하는 변경을 수행할 수 없습니다. (자세한 내용은 섹션 7.19 참조합니다.)

DECO 및 타운크라이어는 설계된 방식은 증명자가 무결성과 기밀성을 보장하는 방식으로 웹 서버에서 데이터를 가져와 검증자에게 제공할 수 있게 합니다. 이 두 시스템은 증명자가 검증자에 제공하는 데이터가 대상 서버에서 인증된 원본임을 보장하며, 이러한 맥락으로 무결성을 유지합니다. 이 시스템은 증명자가 데이터를 수정하거나 수정할 수 있어 (여전히 무결성을 유지하면서도) 기밀성을 지원합니다.

두 시스템에 있어 한 가지 공통된 특징은 **대상 웹 서버를 수정할 필요가 없는 것**입니다. 또한 기존 TLS 지원 서버와 함께 운영할 수도 있습니다. 실제로 이 두 시스템은 서버에서 보기에 투명합니다. 서버의 관점에서 증명자는 일반적인 연결을 확보하고 있습니다.

두 시스템은 비슷한 목표를 갖지만, 신뢰 모델과 구현 방식은 서로 다릅니다. 이에 대해 간략히 설명하겠습니다.

DECO는 무결성 및 기밀성을 달성하기 위해 암호화 프로토콜을 근본적으로 사용합니다. DECO는 사용자와 대상 서버와 세션을 설정하는 동안 증명자는 대화형 프로토콜에 검증자와 동시에 참여합니다. 이 프로토콜을 통해 증명자는 현재 세션 동안 서버로부터 주어진 데이터 조각 D를 수신했음을 검증자에게 증명할 수 있습니다. 증명자는 대문으로 검증자에게 D의 일부 속성에 대한 영지식 증명을 제시할 수 있으며 경우 D를 직접 공개하지 않습니다.

DECO 일반적 사용에서 사용자 또는 단일 노드는 웹 서버와의 비공개 세션에서 DON의 모든 노드로 데이터 D를 내보낼 수 있습니다. 결과적으로 전체 DON은 D의 진위(또는 영지식 증명을 통해 에서 파생된 사실)를 증명할 수 있습니다. 이 기능은 본 백서의 뒷부분에서 다룰 응용 사례뿐만 아니라 어떤 데이터 소스에 대한 높은 무결성 액세스를 DON으로 중국할 경우에도 사용할 수 있습니다. 예컨대, 데이터 공급자와의 독점 계약으로 인해 한 노드만 데이터 소스에 직접 액세스할 수 있는 경우에도 전체 DON이 해당 노드에서 생성된 보고서의 정확성을 증명할 수 있습니다.

타운크라이어(Town Crier)는 Intel SGX와 같은 신뢰할 수 있는 실행 환경(trusted execution environment, TEE)을 사용합니다. 간단히 말하면, TEE는 보안 방지 및 기밀 방식으로 애플리케이션을 실행하는 플랫폼 블랙박스처럼 기능합니다. 원칙상, TEE가 실행 중일 때는 호스트의 소유자조차 TEE로 보호되는 애플리케이션을 (탐지 불가능하게) 변경할 수 없으며, 비밀 데이터가 포함될 수 있는 애플리케이션 상태를 볼 수도 없습니다.

타운크라이어는 DECO 등 모든 기능 및 그 이상을 달성할 수 있습니다. DECO는 증명자를 단일 검증자와의 상호 작용으로 제한합니다. 대조적으로 타운크라이어는 증명자가 대상 서버에서 가져온 데이터 D에 대해 **공개적으로 검증 가능한 증거**, 즉 스마트 컨트랙트를 포함하여 직접 검증할 수 있는 증거를 생성할 수 있게 해줍니다. 또한, 타운크라이어는 비밀(예: 사용자 자격 증명)을 안전하게 수집하고 사용할 수 있습니다.

타운크라이어의 가장 큰 제약은 TEE에 대한 의존성입니다. 생산 TEE는 해당 기술이 아직 초기 단계에 있고 앞으로 분명 성숙해 지겠지만 최근 여러 심각한 취약점이 확인되었습니다. TEE에 대한 추가 논의는 부록 B.2.1 및 B.2.2를 참조하시기를 바랍니다.

DECO 및 타운크라이어의 몇 가지 응용 사례는 섹션 4.3, 4.5 및 9.4.3과 부록 C.1을 참조하시기를 바랍니다.

3.6.3 기존의 온체인 체인링크 서비스

체인링크 오라클 네트워크는 오늘날 다양한 블록체인 및 기타 탈중앙 시스템 전반에 걸쳐 다양한 주요 서비스를 제공하고 있습니다. 이 백서에 설명된 추가적인 개발 사항은 이러한 기존 서비스 기능과 범위를 확장하게 될 것입니다. 다음 세 가지 예를 들 수 있습니다.

데이터 피드: 오늘날 스마트 컨트랙트에 의존하는 대부분의 체인링크 사용자는 데이터 피드를 사용합니다. 데이터 피드란 권위 있는 오프체인 소스에 따른 주요 데이터 조각의 현재 가치에 대한 보고서를 의미합니다. 예를 들어 **가격 피드**는 거래소 또는 데이터 집계 서비스에 따라 디지털 자산, 상품, 외환, 지수, 주식 등의 자산 가격을 보고합니다. 오늘날 이러한 피드는 에이브(Aave)[147] 및 신세틱스(Synthetix)[208] 등의 디파이 시스템에 사용되며 미국 수십억 달러의 온체인 가치를 확보하는 데 기여하고 있습니다. 체인링크 데이터 피드의 다른 예로는 파라메트릭 작물 보험(parametric crop insurance)[75]의 기후 데이터와 선거 데이터[93]가 있습니다.

이 백서에서 설명하는 DON 및 기타 기술의 배포는 다음을 포함한 다양한 방식으로 체인링크 네트워크의 데이터 피드 제공을 향상시킬 것입니다.

- **확장성:** OCR과 DON은 체인링크 서비스가 지원하는 많은 블록체인에 하여 서비스의 극적인 확장을 목표로 합니다. 예컨대, 우리는 오늘날 메인 체인에서 체인링크를 사용하는 노드가 제공하는 데이터 피드 수를 100개에서 1,000개 이상으로 늘리도록 큰 도움을 줄 것으로 예상합니다. 이러한 확장은 체인링크 생태계가 스마트 컨트랙트와 관련된 컨트랙트를 포함적으로 제공하여 기존 및 미래 요구를 충족하고 예측하는 목표를 달성하도록 할 도움 것입니다.
- **향상된 보안:** DON은 중간 보고서의 저장을 통해 고성능 모니터링 및 성능 및 정확성 측정을 위한 노드 동작 기록을 유지하며, 이를 통해 체인링크 노드에 평판 시스템의 강력한 경험적 기반을 제공합니다. FSS와 TEF는 가격 피드를 트랜잭션 데이터에 유연하게 통합하여 선행 등의 공격을 방지할 수 있도록 합니다. (명시적) 예제는 데이터 피드 보안에 대한 기존의 암호경제학적 보안을 강화합니다.
- **피드 민첩성:** DON은 블록체인 불가저론적 시스템(실제로 당 광범위하게는 소비자 불가저론적 시스템)으로서 다수의 블록체인에 대한 데이터 피드 제공을 촉진할 수 있습니다. 단일 DON은 주어진 피드를 여러 블록체인 세트에 동시에 푸시할 수 있으므로 체인별 오라클 네트워크의 필요성을 없애며, 기존 피드를 새로운 블록체인에 신속하게 배치하고 현재 서비스 중인 블록체인에 추가 피드를 신속하게 배포할 수 있습니다.
- **기밀성:** DON에서 일반화된 연산을 수행하는 기능은 민감 데이터에 대한 연산이 체인 외부에서 발생하도록 하여 체인 내에서의 노출을 방지합니다. 또한 DECO나 타운크라이어를 사용하면 DON 노드에도 노출되지 않는 데이터를 토대로 보고서를 생성할 수 있어 더욱 강력한 기밀성을 달성할 수 있습니다. 이에 관한 예시는 섹션 4.3 및 섹션 4.5를 참조합니다.

검증 가능한 랜덤 함수(Verifiable Random Functions, VRF): 다양한 디앱은 그 자체의 공정한 운영을 검증할 수 있도록 하기 위해 정확하고 검증 가능한 무작위 소스를 필요로 합니다. 대체 불가능 토큰(Non-Fungible Tokens, NFT)을 예로 들어볼 수 있습니다. 에이브고치(Aavegotchi)[23] 및 액시 인피니티(Axie Infinity)[35]에서 NFT 기능의 희귀성은 이더카드(Ether Cards)[102]의 티켓 기반 도면을 통한 무작위 배포 결과가 무작위로 결정되는 다양한 게임 대입, 임의의 승자가 예금을 할당하는 풀투게더(PoolTogether)[89]와 같은 무손실 저축 게임 등의 비전통적인 금융 상품과 마찬가지로 체인링크 VRF에 의해 결정됩니다. 다른 블록체인 및 비블록체인 애플리케이션에서도 탈중앙 시스템 위원회의 선정 및 복권 실행을 포함하는 안전한 무작위 소스가 필요합니다.

블록 해시는 예측할 수 없는 무작위성을 만들어낼 수 있지만, 적대적 채굴자가 (어느 정도는 트랜잭션을 제출하는 사용자) 조작에 취약합니다. 체인링크 VRF[78]는 훨씬 더 안전한 대안을 제공합니다. 오라클에는 비밀키를 오프체인에 유지하고, 공개키 pk 가 공개, 하나와 연관된 개인/공개키 쌍 (sk, pk) 가 존재합니다. 그곳은 일련의 값을 출력하기 위해, 해시를 사용하여 의존 컨트랙트에 의해 제공된 예측할 수 없는 시드 s 에 응답하여 올바른 정확성 증명과 함께 함께 $y = F_{sk}(s)$ 도출합니다. (체인링크에서 사용 가능한 VRF는 [180]을 참조합니다.) VRF를 검증 가능한 해시 함수의 한 부분으로 이해할 수 있으며, 나아가 y 의 정확성을 확인할 수 있다는 사실입니다. 결과적으로 y 의 값은 본질적으로 sk 를 학습할 수 없는 공격자에게 예측할 수 없게 됩니다.

체인링크 VRF는 오프체인 비밀키 관리를 포함하는 하나의 애플리케이션에 제공함으로 간주할 수 있습니다. 보다 일반적으로 DON은 애플리케이션에서 사용자는 위한 개별 키의 안전하고 탈중앙화된 스토리지를 제공하며 이 기능을 일반화된 오프체인 연산으로 결합할 수 있습니다. 다른 많은 애플리케이션에서도 랜덤함수를 필요로 합니다.

준비금 증명(Proof of Reserves, 섹션 4.1 참조) 및 사용자의 탈중앙화된 자격 증명(및 기타 디지털 자산, 섹션 4.3 참조)에 대한 키 관리 등 이러한 애플리케이션에서의 일부 예를 제공합니다.

키퍼(Keeper): 체인링크 키퍼[78]는 개발자가 오프체인 작업의 탈중앙 실행을 위한 코드를 작성하여 일반적으로 의존 스마트 컨트랙트의 실행을 트리거할 수 있도록 해줍니다. 키퍼가 등장하기 전에는 개발자가 이러한 오프체인 로직을 자체적으로 운영하여 중앙화된 실제 지점을 생성하는 것(상당한 중복된 개발 노력의 수반됨)이 관례였습니다. 대신 키퍼는 이러한 작업의 탈중앙 아웃소싱을 위해 사용하기 쉬운 프레임워크를 제공하여 개발 주기를 단축하고 활성화 및 기타 보안 속성을 강화하게 보장할 수 있습니다. 키퍼는 가격에 따른 대출 청산 또는 금융 트랜잭션의 실행, 시간에 따른 에어드롭(airdrop) 시작 또는 수익(yield harvesting)을 제공하는 시스템에서 지불 등과 같은 다양한 트리거 목표를 모두 지원할 수 있습니다.

DON 프레임워크에서 개시자는 여러 의미에서 키퍼의 일반화형이라고 간주할 수 있습니다. 개시자는 어떤 코드를 사용할 수 있으므로 온체인 및 오프체인 시스템에 대한 모듈화된 인터페이스 라이브러리를 활용하여 안전하고 정교한 기능을 신속하게 개발할 수 있습니다. 개시자는 실행파일에서 연산을 시작하며, 자체적으로 DON의 엔진을 다양성을 제공하여 온체인 및 오프체인 애플리케이션에 대해 이 백서에서 제시하는 광범위한 탈중앙화 서비스를 가능하게 합니다.

3.6.4 노드 평판/실적 내역

기존 체인링크 생태계는 기본적으로 체인에 기여하는 노드의 실적 내역을 문서화합니다. 이 기능은 개발 노드 운영자 및 데이터 피드에 대한 실제 데이터를 수집, 필터링 및 시작하는 평판 중심의 리소스 수집 모델로 이어졌습니다. 사용자는 이러한 리소스를 참조하여 노드 선택에서 정의 임작한 의사결정을 내리고 기존 네트워크의 작동을 모니터링할 수 있습니다. 이와 유사한 기능은 사용자와 DON 선택에 도움이 됩니다.

예를 들어, 오늘날 market.link와 같은 무허가 방식의 시장에서는 노드 운영자가 자신의 오라클 서비스를 나열하며, 체인링크에 있는 노드의 프로필은 소유자가 기존 도메인 이름 및 소셜 미디어 계정에 바인딩하는 Keybase와 같은 서비스로 오프체인 신원을 증명할 수 있습니다[44]. 또한 market.link 및 reputation.link에서 이용하는 성능 분석 도구는 사용자들이 평균 응답 대기 시간, 보고서의 값과 체인인 컨센서스 값과 편차, 생성된 수익, 완수된 작업 등을 포함한 개별 노드의 과거 실적에 대한 통계를 조회할 수 있게 해줍니다. 이러한 분석 도구는 사용자가 다른 사용자의 다양한 탈중앙 네트워크 채택을 추적할 수 있게 해줍니다. 이는 노드에 대해 이러한 네트워크를 보강한다는 암묵의 암묵적 보증을 제공합니다. 결과적으로 평판 ‘신뢰의 웹’이 생성됩니다. 여기에서는 높은 가치의 탈중앙 애플리케이션에선 특정 노드를 사용하여 다른 사용자가 자신의 노드 선택 결정을 관찰하고 고려할 수 있는 노드의 신뢰 신호를 생성합니다.

DON(조차 OCR 사용)을 사용하면 트랜잭션 처리 및 컨트랙트 활동이 일반적으로 오프체인으로 전환됩니다. 노드 실적을 기록하는 탈중앙화 모델은 DON 자체 내에서 여전히 제공됩니다. 실제로 DON의 공공성 및 데이터의 용량은 매우 실질적일 것입니다. 결과적으로 이러한 평판 시스템은 DON에 대해 탈중앙화 연산을 수행하며, 평판 서비스에서 사용하고 MAINCHAIN 상에서 체크포인트(checkpoint)로 삼을 수 있도록 신뢰할 수 있는 요약물 생성할 수 있게 해줍니다.

노드의 많은 부분이 손상된 경우, 원칙상 DON이 구성 노드의 동작을 잘못 표현할 수 있지만 온체인 데이터를 전달하는 **DON 자체의 집합적 성능**은 MAINCHAIN상에서 가시성이 제공되기 때문에 잘못 표시될 일이 없습니다. 더불어 우리는 DON에서 노드 동작에 대한 정확한 내부 보고를 촉진하는 메커니즘을 탐색할 계획을 세웠습니다. 예컨대, DON은 자연의 중계된 보고서에 기여하는 데이터를 가장 빠르게 반환하는 고성능 노드의 하위 집합을 보고함으로써 노드가 잘못된 보고서에 이익을 제거하도록 유도합니다. 이 하위 집합에 노드를 잘못 포함하는 것은, 포함해야 할 노드를 잘못 제외해 해당 노드를 잘못 처벌했다는 의미와 같습니다. DON에 의한 반복적인 보고 실패는 정직한 노드가 DON을 떠나도록 유도합니다.

정확한 실적 내역의 탈중앙화된 수집, 결과적으로 사용자가 고성능 노드를 식별하고 노드 운영자가 평판을 구축할 수 있는 능력은 체인링크 생태계의 차별적이고 중요한 기능입니다. 섹션 7에서는 DON이 제공하는 경제적 보안에 대한 엄격하고 광범위한 관점의 핵심 요소로서 이를 어떻게 추론할 수 있는지를 논의합니다.

4. 탈중앙화 오라클 네트워크로 구현된 탈중앙화 서비스

DON의 다양성과 DON이 방대한 새로운 서비스를 가능하게 하는 방법을 설명하기 위해 이 섹션에서는 DON 기반 애플리케이션의 5가에 예를 다음과 같이 제시하고 이를 실행하는 하이브리드 컨트랙트에 관해 설명합니다.

- (1) 크로스체인(Cross-Chain) 서비스의 한 형태인 **준비금 증명(Proof of Reserve)**
- (2) 미들웨어 기반의 추상화 레이어를 생성하여 최소화한 블록체인 특정 코드 또는 전문 지식으로 블록체인 애플리케이션의 개발을 용이하게 하는 **엔터프라이즈/레거시 시스템과의 인터페이스**
- (3) 사용자가 자신의 ID 문서 및 자격 증명을 알고 관리할 수 있게 해주는 **프라이버시 탈중앙 ID**
- (4) 블록체인에 중요한 인프라 트랜잭션(예: 오라클 보고서)을 적시에 포함하도록 보장하는 서비스인 **우선 채널**
- (5) 참여 당사자의 민감한 데이터를 숨기는 금융 스마트 컨트랙트의 **기밀성 보존 기능**

여기에서는 스마트 컨트랙트(SC)를 사용하여 하이브리드 컨트랙트의 MAINCHAIN 부분을 나타내고, DON 구성요소를 개별적으로 또는 실행 가능한 실행파일(exec) 측면에서 설명합니다.

4.1 준비금 증명

많은 애플리케이션에 있어서 블록체인 간의 상태를 중계하는 것은 유용합니다. 이러한 서비스에서 인기 있는 응용 형태는 **랩된 자산(wrapping)**입니다. WBTC[15]와 같은 랩된 코인은 디파이에서 인기 있는 자산이 되고 있습니다. 여기에는 기반 블록체인 **MAINCHAIN(1)**의 ‘래핑된’ 백업 자산을 예치하고 다른 대상 블록체인 **MAINCHAIN(2)**에 해당 토큰을 생성하는 작업이 포함됩니다. 예를 들면, WBTC는 비트코인(BTC) 블록체인의 BTC에 해당하는 ETH 블록체인의 ERC-20 토큰입니다.

\$MAINCHAIN(2)\$에서의 컨트랙트는 \$MAINCHAIN(1)\$에 대한 직접적인 가시성을 갖지 않기 때문에 스마트 컨트랙트에서 랩된 자산의 예금에 대해 보고하기 위해 명시적 또는 암묵적으로 오라클에 의존해야 하며, 이에 따라 종종 준비금 증명(Proof of Reserves)이 생성됩니다. 예를 들어 WBTC[15]에서 관리 주체인 비트코인(BitGo)은 BTC를 보유하고 WBTC를 발행하며 체인링크 네트워크는 준비금 증명[76]을 지원합니다.

DON 자체로도 준비금 증명을 제공할 수 있으며, 그보다 더 많은 것도 할 수 있습니다. DON은 비밀을 관리할 수 있으며 적절한 어댑터를 사용하여 원하는 블록체인에서 트랜잭션을 실행할 수 있습니다. 결과적으로 DON은 랩핑된 자산에 대한 여러 관리 주체 중 하나 또는 단독의 탈중앙화된 관리 주체로서 활동할 수 있습니다. 따라서 DON은 준비금 증명을 사용하는 기존 서비스의 보안을 강화하는 플랫폼 역할을 할 수 있습니다.

예를 들어 \$MAINCHAIN(1)\$이 비트코인이고 \$MAINCHAIN(2)\$가 이더리움이라고 가정하겠습니다. \$MAINCHAIN(2)\$에서 컨트랙트 \$SC_c\$는 랩핑된 BTC를 나타내는 토큰을 발행합니다. DON은 BTC 주소인 \$addr_{DON}\$을 제어합니다. 그런 다음 BTC를 랩핑하기 위해 사용자는 \$MAINCHAIN(2)\$ 주소 \$addr_U\$와 함께 \$BTC\$를 \$addr_{DON}\$에 보냅니다. DON은 \$MAINCHAIN(1)\$ 어댑터를 통해 \$addr_U\$ 및 금액을 \$L\$에 기록합니다. 이 예치금은 관할하고 입금 충분한 보증 확률이 확인되면 \$MAINCHAIN(2)\$ 어댑터를 통해 \$SC_c\$에 메시지를 보냅니다. 이 메시지는 \$SC_c\$에 \$addr_U\$에 해당하는 토큰을 발행하도록 지시합니다.

\$U7H \times\$ 토큰을 릴리스하려면 그 반대의 상황이 발생합니다. 그러나 MAINCHAIN(1)에서는 \$addr_{DON}\$이 \$X\$ BTC를 \$addr_U(1)\$에 (또는 사용자가 요청하면 다른 주소로) 보냅니다. 물론 이러한 프로토콜은 사용자와 직접 연결하지 않고 거래소와 연동되도록 조정될 수 있습니다.

4.2 엔터프라이즈/레거시 시스템용 인터페이스

DON은 준비금 증명 예시에서와 같이 블록체인 사이에서 가교 역할을 할 수 있지만 또 다른 목적은 블록체인과 레거시 시스템[176] 또는 중앙은행의 디지털 통화와 같은 블록체인 유사 시스템 사이에서 양방향 가교 역할을 제공하는 것입니다[30].

기업은 기존 시스템과 프로세스를 탈중앙 시스템에 연결하는 데 있어 다음과 같은 여러 문제에 직면해 있습니다.

- **블록체인 민첩성:** 블록체인 시스템은 빠르게 변화합니다. 기업은 거래 상대가 트랜잭션을 수행하기를 원하지만, 기존 인프라에서 지원하지 않는 블록체인이 출현하거나 급속히 성장하는 문제에 직면할 수 있습니다. 일반적으로 블록체인의 역동성을 개별 기업이 전체 생태계를 따라가기 어렵게 만듭니다.
- **블록체인 전문 개발 리소스:** 민첩성 문제 측면에서 많은 조직이 최첨단 블록체인 지식을 갖춘 전문 인력을 고용하거나 육성하기는 어렵습니다.
- **비밀키 관리:** 블록체인 또는 디지털 자산의 비밀키를 관리하려면 기존 사이버 보안 관행과 차별화된 높은 전문 지식이 필요한데 이는 많은 기업에서 확보하기 어려운 것들입니다.
- **기밀성:** 기업은 민감한 독점 데이터를 체인에 노출하기를 꺼립니다.

이러한 문제 중 처음의 세 가지 문제를 해결하기 위해 개발자는 DON을 보안 미들웨어 레이어로 사용하여 엔터프라이즈 시스템에 블록체인에서 읽거나 쓸 수 있도록 할 수 있습니다. DON은 개발자와 사용자를 모두를 위해 수수료 억화, 체인 재작성 등과 같은 세부적인 기술적 고려 사항을 추상화할 수 있습니다. DON은 기업 시스템에 간소화된 블록체인 인터페이스를 제공하여 블록체인을 인식하는 기업 애플리케이션의 개발을 상당히 단순화시키고, 기업이 블록체인 관련 개발 리소스를 획득하거나 육성해야 할 부담을 해소할 수 있습니다.

이러한 DON의 사용은 기업 개발자가 대부분 블록체인에 구애받지 않는 스마트 컨트랙트 애플리케이션을 만들 수 있다는 점에서 특히 매력적입니다. 결과적으로 DON이 미들웨어 역할을 하는 블록체인의 집합이 블록체인 엔터프라이즈 사용자가 쉽게 액세스할 수 있는 블록체인의 집합이 커집니다. 개발자는 내부적 스마트 개발된 애플리케이션을 최소한으로 수정하면서 기존 블록체인에서 새로운 블록체인으로 애플리케이션을 포함할 수 있습니다.

추가적인 기밀성 문제를 해결하기 위해 개발자는 본 백서에서 소개하는 도구를 활용해 DON 애플리케이션을 지원하기 위해 배포할 수 있습니다. 여기에는 섹션 3.6.2의 DECO 및 타운크라이어뿐만 아니라, 섹션 7.1.2에서 논의된 기밀 보존 A의 수집과, 섹션 4 나머지 부분에서 다루게 될 여러 애플리케이션별 접근 방식이 포함됩니다. 이러한 DON 시스템은 체인에 있는 민감한 엔터프라이즈 소스 데이터를 공개하지 않고 엔터프라이즈 시스템 상태에 대한 높은 무결성 온체인 증명을 제공할 수 있습니다.

4.3 탈중앙 ID

탈중앙 ID(Decentralized identity)는 사용자가 제3자에 의존하지 않고 자신의 자격 증명을 얻고 관리할 수 있어야 한다는 개념을 가리키는 용어입니다. 탈중앙화된 자격 증명은 소유자의 속성 또는 주장에 대한 증명이며, 종종 **클레임(claims)**이라고도 합니다. 자격 증명은 클레임을 사용자와 신뢰할 수 있는 방식으로 연결할 수 있는, 소위 **발급자(issuers)**라고 하는 엔티티가 디지털 서명합니다. 대부분의 제안된 체계에서, 클레임은 주어진 사용자의 대한 범용 식별자인 탈중앙화 식별자(Decentralized Identifier, DID)와 연결됩니다. 자격 증명은 사용자가 보유한 비밀키의 공개키에 바인딩됩니다. 따라서 사용자는 비밀키를 사용하여 소유권 주장을 증명할 수 있습니다.

기존 및 제안된 체제([14, 92, 129, 216])에는 심각한 세 가지 제약 사항이 존재하는데, 다음과 같습니다.

- **레거시 호환성 부족:** 기존의 탈중앙 ID 시스템은 DID 자격 증명을 생성하기 위해 발급자인 기관 커뮤니티에 의존합니다. 기존 웹 서비스는 일반적으로 데이터에 디지털 서명을 하지 않기 때문에 발급자는 특수 목적 시스템으로 시작해야 합니다. 탈중앙 ID의 생태계가 없다면 이를 수행할 유인책이 없으므로 닭과 달걀의 문제가 발생합니다. 즉, 발급자 생태계를 부트스트랩하는 방법이 불분명해지게 됩니다.
- **작동하지 않는 키 관리:** 탈중앙 ID 시스템은 사용자가 비밀키를 관리해야 하며, 이는 디지털 자산의 위험에 비추어 볼 때, 실행 불가능한 책임인 것으로 보입니다. 키 관리 실패[194]로 인해 약 4,000,000 BTC가 영구적으로 손실된 것으로 추정되며, 결국 많은 사용자는 자신의 암호화 자산을 거래소에 저장[193]하게 되어 탈중앙화를 저해하고 있습니다.
- **개인정보보호/시빌 저항 부족:** 투표, 토큰 판매 중 토큰의 공정한 할당 등과 같은 애플리케이션의 기본 보안 요구 사항은 사용자가 여러 ID를 주장할 수 없게 하는 것입니다. 기존의 탈중앙화된 ID 제안은 사용자가 이러한 시빌 저항(Sybil resistance)을 위해 실제 ID를 공개하도록 요구하므로 중요한 개인정보보호 보장을 훼손합니다.

CanDID[160]라는 시스템에서 볼 수 있듯이 DON 내에서 분산형 연산을 수행하는 노드 위원회와 DECO 또는 타운크라이어와 같은 도구 사용의 조합을 사용하면 이러한 문제를 해결할 수 있습니다.

DECO 또는 타운크라이어는 설계에 따라 기존 웹 서비스를 수정 없이 기밀성 보존 자격 증명 발급자로 전환할 수 있습니다. 이를 통해 DON은 이 목적을 위해 관련 데이터를 자격 증명으로 내보낼 수 있게 되며, 자격 증명에 나타나지 말아야 할 민감 데이터를 숨길 수 있게 됩니다.

또한 사용자의 키 복구를 지원하여 키 관리 문제를 해결하도록, DON은 사용자가 비밀 공유 형식으로 비밀키를 저장할 수 있도록 합니다. 사용자는 DON의 노드에(마찬가지로 타운크라이어 또는 DECO를 사용할 수 있음) 미리 결정된 제공자(예: 트위터, 구글, 페이스북)의 계정에 로그인할 수 있음을 증명함으로써 키를 복구할 수 있습니다. OAuth와 달리 타운크라이어 또는 DECO를 사용할 경우의 이점은 사용자의 개인성을 보호합니다. 이 두 가지 도구를 사용하면 사용자가 웹 제공자 식별자를 DON에게 공개하는 것을 방지할 수 있습니다.

마지막으로, 시빌 저항을 제공하기 위해 [160]에서 볼 수 있는 것과 같이, DON은 사용자 등록 시 해당 사용자의 고유한 실제 식별자(예: 사회 보장 번호(Social Security Numbers, SSN))를 온체인 식별자로 변환하는 개인정보보호 보존 변환을 수행할 수 있습니다. 따라서 시스템은 SSN과 같은 민감한 데이터를 개별 DON 노드에 공개하지 않고 중복 등록을 감지할 수 있습니다.⁷

DON은 무허가 또는 허가된 블록체인(예: 하이퍼레저 인디(Hyperledger Indy)[129])에서 외부 탈중앙 ID 시스템을 대신하여 이러한 서비스를 제공할 수 있습니다.

응용 사례: KYC: 탈중앙 ID는 블록체인에서 금융 애플리케이션에 대한 요구 사항을 간소화하는 동시에 사용자 개인정보를 개선하는 수단으로서 유망합니다. 탈중앙 ID가 도움이 될 수 있는 두 가지 문제는 자금세탁 방지/고객확인제도(anti-money-laundering / know-your-customer, AML/KYC) 규정에 따른 인증 및 규제 준수입니다. 많은 국가의 AML 규정에서는 금융 기관(및 기타 기업)이 트랜잭션을 수행하는 개인 및 기업의 신원을 확인하고 검증할 것을 요구합니다. KYC는 금융기관의 광범위한 AML 정책의 한 요소이며, 일반적으로는 사용자 행동에 대한 모니터링과 자금 흐름에 대한 관찰을 요구합니다.

KYC는 일반적으로 사용자가 어떤 형태로든 신원 자격 증명을 제시할 것을 요구합니다(예: 온라인 웹 양식은 알려하거나 비디오 세션에 경우 사용자의 얼굴과 신원 증명서를 함께 제시, 등). 탈중앙화된 자격 증명 의 안전한 생성 및 표시는 원칙적으로 다음과 같은 여러 측면에서 유익한 대안이 될 수 있습니다.

- (1) 자격 증명을 취득하면 모든 금융 기관에 활화하게 제시할 수 있으므로, 사용자 및 금융 기관을 위한 KYC 프로세스가 더 효율화됩니다.
- (2) 개인 식별 정보(personally identifiable information, PII) 손실 및 영상 확인 중의 스푸핑을 통한 신원 도용의 기회를 줄여 사기를 줄입니다.
- (3) 사용자가 자신의 데이터를 계속 제어할 수 있으므로 금융 기관에 PII 손상 위험이 완화됩니다.

AML 규정 준수 실패에 대해 금융 기관이 지불하게 되는 수십억 달러의 벌금과 KYC를 위해 많은 금융 기관 연간 수백만 달러를 지출한다는 사실을 감안하면, 탈중앙 ID는 전통적인 금융 기관과, 더 나아가 소비자에게 상당히 보장 절감 효과를 제공할 수 있습니다[196]. 전통적인 금융 부문은 규정 준수를 위한 새로운 도구의 채택이 느리지만 디파이 시스템은 점점 더 이를 수용하고 있습니다[43].

응용 사례: 부족 담보 대출: 오늘날 대출을 지원하는 대부분의 디파이 애플리케이션은 완전 담보 대출만 제공합니다. 이는 대출금액을 초과하는 가치의 디지털 자산을 예금하는 차용자에게 제공하는 대출입니다. 디파이 커뮤니티가 일반적으로 부족 담보 대출(undercollateralized loan)이라고 부르는 것에 최근 많은 관심이 쏠리고 있습니다. 이는 해당 담보의 가치가 대출 원금보다 낮은 대출을 의미합니다. 부족 담보 대출은 종종 전통적인 금융 기관에서 수행하는 대출과 유사합니다. 대출 상환을 보증하기 위해 담보물에 의존하는 대신 차용인의 신용 기록을 토대로 대출 결정을 내리기 때문입니다.

부족 담보 대출은 아직 초기 단계이지만 디파이 대출 시장에서 성장하고 있습니다. 부족 담보 대출은 법적 컨트랙트와 같은 전통적인 금융 기관에서 사용하는 메커니즘에 의존합니다[91]. 부족 담보 대출의 성장에 필수적인 요구 사항은 강력한 무결성, 즉 정확한 데이터의 보증을 제공하는 방식으로 디파이 시스템에서 사용자의 신용등급(기존 대출 결정의 핵심 요소)에 대한 데이터를 제공하는 기능이라 할 수 있습니다.

DON이 활성화된 탈중앙화 ID 시스템은 차용인의 민감한 정보의 기밀성을 유지하면서 신용도를 증명하는 높은 보증 자격 증명을 생성할 수 있도록 합니다. 특히, 차용인은 잠재적으로 민감한 다른 데이터를 노출하지 않고 DON이 증명한 데이터만 노출되도록, 신뢰할 수 있는 온/오프체인 출처의 기록을 기반으로 이러한 자격 증명을 생성할 수 있습니다. 예컨대, 차용인은 자신의 기록에 있는 정확한 평가 점수나 기타 데이터를 공개하지 않고도 신용 조사 기관의 신용 점수가 특정 임계값(예: 750)을 초과했음을 나타내는 자격 증명을 생성할 수 있습니다. 원하는 경우 이러한 자격 증명을 익명으로 생성할 수도 있습니다. 즉, 사용자의 이름은 민감 데이터로 취급될 수 있으며, 그 자체가 오라클 노드나 탈중앙화 자격 증명 노드에 노출되지 않게 처리할 수 있습니다. 자격 증명 자체는 애플리케이션에 따라 온체인 또는 오프체인에서 사용될 수 있습니다.

요약하면, 차용인은 신용 기록에 대한 필수 정보를 신용 기록에 대해 강력한 무결성을 유지하면서 불필요하고 민감한 데이터의 노출 위험 없이 제공할 수 있습니다.

차용인은 대출 결정을 내리는 데 도움이 되는 다양한 기타 기밀성 보존 자격 증명을 제공할 수 있습니다. 예를 들어, 자격 증명은 차용인이 (오프체인) 자산을 소유하고 있음을 증명할 수 있습니다.

응용 사례: 인증: 많은 국가에서 미등록 증권을 판매할 수 있는 투자자 유형을 제한합니다. 예컨대, 미국의 SEC 규정 D는 개인이 이러한 투자 기회에 대한 인증을 받으려면 100만 달러의 순자산을 보유하고 특정 수준 소득 요구를 충족하거나 특정 전문 자격 또는 자산을 보유해야한다고 규정합니다[210]. 현재의 인증 프로세스는 번거롭고 비효율적이며, 종종 회계사와 입증되어 있는 유사 증명이 필요합니다.

탈중앙 ID 시스템을 사용하면 사용자는 자신의 인증 규정 준수를 증명하는 기존 온라인 금융 서비스 계정에서 자격 증명을 생성할 수 있으므로, 결과적으로 개인정보를 보호하는 방법론을 보호하는 보조시스템이 추진됩니다. 또한, DECO 및 타운크라이어의 개인정보보호 속성을 통하여 사용자의 재정 상태에 대한 세부 정보를 직접 공개하지 않고도 이러한 자격 증명을 무결성을 지키며 생성할 수 있습니다. 예컨대, 사용자는 재정 상태에 대한 추가 정보를 공개하지 않고 순자산이 100만 달러임을 증명하는 자격 증명을 생성할 수 있습니다.

4.4 우선 채널

우선 채널 (Priority channels)은 DON을 사용하여 쉽게 구축할 수 있는 유용한 새 서비스입니다. 우선 채널의 목표는 네트워크 정체 기간 동안 MAINCHAIN에서 선택적으로 우선순위가 높은 트랜잭션을 적시에 제공하는 것입니다. 우선 채널은 블록 공간에 대한 미래 계약의 한 가지 형태로 볼 수 있으므로, 프로젝트 시카고(Project Chicago)[61, 136]의 일환으로 등장한 용어인 암호화 상품(cryptocommodity)으로도 볼 수 있습니다.

우선 채널은 금융 트랜잭션과 같은 일반적인 사용자 수준 활동이 아닌 채굴자에게 오라클, 컨트랙트에 대한 거버넌스 기능 등과 같은 인프라 서비스를 가능하기 위해 특별히 고안되었습니다. 사실 여기에서 설계된 것처럼 네트워크에서 100% 미만의 채굴력으로 구현된 우선 채널은 전달 시간에 대한 느슨한 경계만 제공할 수 있으므로 선행과 같이 대단히 속도에 의존하는 목적의 사용을 방지할 수 있습니다.

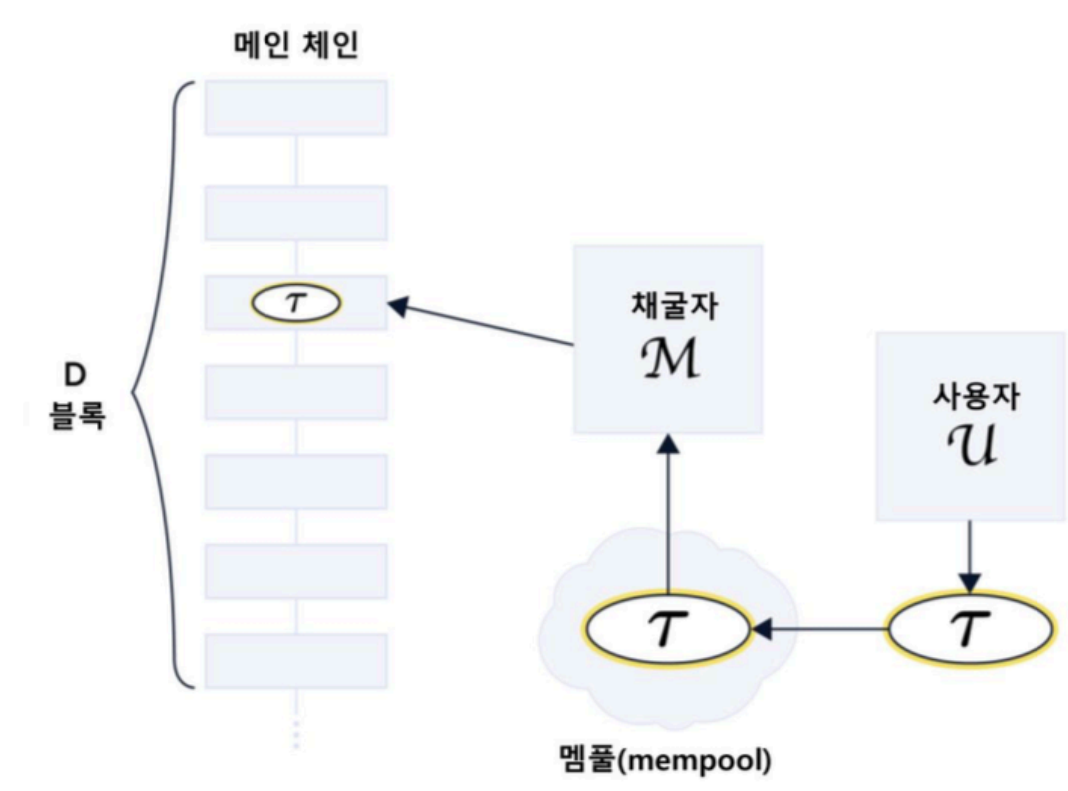


그림 10: 우선 채널은 채굴자 M (또는 보다 일반적으로 채굴자 집합 M)가 사용자 U 에게 제공하게 되는 자신의 트랜잭션 τ 이 메мпool에 포함된 D 블록 내에서 채굴될 것이라는 일종의 보장과 같습니다. 컨트랙트 SC는 DON 모니터링을 사용하여 채널의 서비스 조건을 실행할 수 있습니다.

우선 채널은 채널을 제공하는 채굴자 또는 채굴자 집합(또는 채굴 풀) M , 접근 수수료를 지불하는 사용자 U 간에 체결된 합의의 형식을 취합니다. M 는 U 가 메мпool에 트랜잭션 τ 를 제출할 때(사전에 수수료 한도에 합의한 수수료 가격으로) 이를 다음 D 블록 내에서 체인에 배치한다는 데 동의합니다. 이 아이디어를 도식화하면 그림 10과 같습니다.

우선 채널 컨트랙트 설명: 우선 채널은 대략 다음과 같이 하이브리드 스마트 컨트랙트로 실현될 수 있습니다. SC는 MAINCHAIN의 로직을 나타내고 DON의 로직은 exec으로 표시합니다.

SC는 M 으로부터 보증금/예치 $\$d$ 를 수락하고 U 로부터 선지급 $\$p$ 를 받습니다. DON의 실행파일 `exec`은 메모리를 모니터링하여 U 에 의한 트랜잭션 배치를 트리거합니다. U 가 M 이 제시한 시간에 채굴했다는 트랜잭션을 제출할 경우 SC에 success 메시지를 전송하고 서비스 오류 시 failure 메시지를 전송합니다.

SC는 success 메시지가 제출되면 M 에 지불 $\$p$ 를 전송하고, failure 메시지를 받을 경우 $\$d$ 를 포함한 나머지 모든 자금을 U 에 보냅니다. 성공적으로 완료되면 M 의 보증금 $\$d$ 의 홀딩을 해제합니다.

물론 채굴자 M 은 여러 사용자에게 동시에 우선 채널을 제공할 수 있으며, 사전에 합의한 메시지 수에 맞춰 U 에 대한 우선 채널을 열 수 있습니다.

4.5 기밀성 보존 디파이/믹시클

오늘날 디파이(DeFi) 애플리케이션[1]은 사용자에게 제공하는 기밀성이 거의 또는 전혀 없습니다. 모든 트랜잭션은 체인 상에서 볼 수 있기 때문입니다. 다양한 영지식 기반 접근(예: [149, 217])은 트랜잭션에 대한 개인정보보호를 제공할 수 있으며, TEF는 이를 지원하기에 충분한 방식입니다. 그러나 이러한 접근 방식은 포괄적이지 않으며, 일반적으로 트랜잭션의 기반이 되는 자산을 보호하지 않습니다.

우리가 궁극적으로 DON에서 지원하고자 하는 광범위한 연산 도구 트는 이러한 격차를 메울 수 있는 다양한 방식으로 개인정보를 구현하여 다른 시스템의 개인정보보호를 보완하는 데 도움을 줍니다. 예컨대, 체인링크 랩스(Chainlink Labs) 연구진[135]이 제안한 기밀성 보존 디파이 도구인 믹시클(Mixicles)은 금융 상품을 뒷받침하는 자산 유형을 숨길 수 있으며 DON 프레임워크에 매우 자연스럽게 부합합니다.

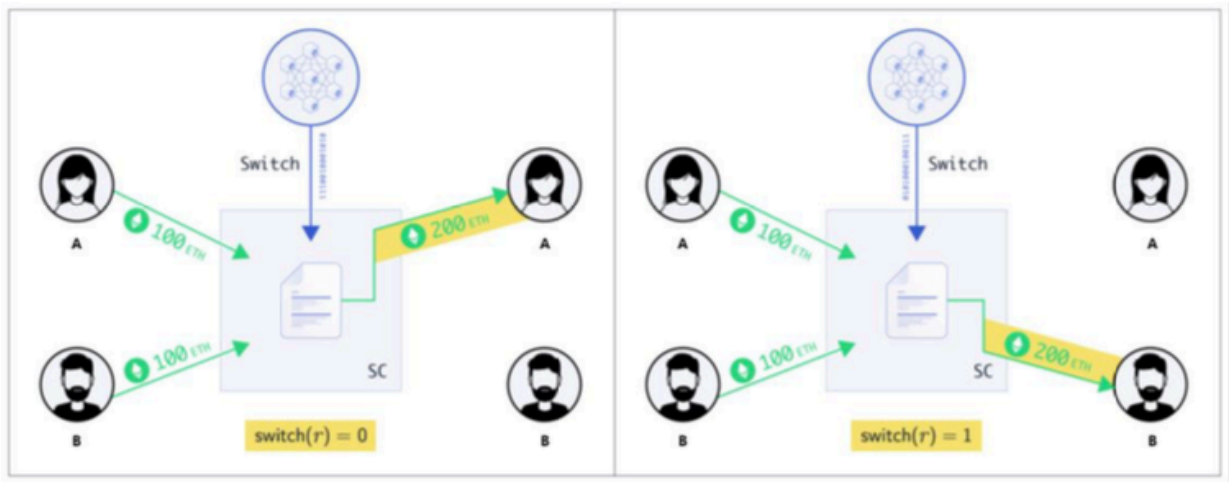
믹시클은 간단한 이진 옵션을 구현하기 위한 용도로 가장 쉽게 설명됩니다. 바이너리 옵션은 하나의 금융상품으로, [135]와의 일관성을 위해 참조할 두 명의 사용자가 두 가지 가능한 결과가 있는 이벤트(예: 자산이 미리 지정된 시간에 목표 가격을 초과하는지 여부)에 플레이어로 베팅하게 됩니다. 다음 예시는 이 아이디어를 보여줍니다.

예시 2. A 씨와 B 씨는 C 씨의 버블 토큰 (Carol's Bubble Token, *CBT*) 이라는 자산 가치를 기반으로 한 바이너리 옵션의 당사자입니다. A 씨는 2025년 6월 21일 시간 T (정오)에 *CBT*의 시장 가격이 최소 250 USD가 될 것으로 베팅합니다. B 씨는 그 반대로 베팅합니다. 각 플레이어는 미리 지정된 마감일까지 100 ETH를 입금합니다. 승리하게 되는 플레이어는 200 ETH를 받게 됩니다 (즉, 100 ETH의 이익을 얻음).

기존 체인링크 오라클 네트워크 O 를 감안할 때 예시 2의 합의를 실현하는 스마트 컨트랙트 SC를 구현하기는 쉽습니다. 두 플레이어는 각각 100 ETH를 SC에 예치합니다. 시간 T 가 지난 후, T 시점에 *CBT*의 가격 r 을 요청하는 쿼리 q 가 O 에 전달됩니다. O 는 해당 가격 보고서 r 을 SC에 보냅니다. SC는 $r \geq 250$ 일 경우 A씨에게 돈을 보내고 그렇지 않으면 B씨에게 보냅니다. 그러나 이 접근 방식은 체인 상의 r 을 드러내므로 관찰자가 바이너리 옵션의 기본 자산을 쉽게 추론할 수 있게 합니다.

믹시클에서는 SC의 결과를 하나의 술어 $\text{switch}(r)$ 로 연산된 어떤 이진값을 전송하는 스위치(Switch)의 관점에서 개념적으로 생각하는 것이 좋습니다. 이 예시에서 $r \geq 250$ 의 조건이라면 $\text{switch}(r) = 0$ 의 식이 성립하고, 이 결과가 따라 A씨가 승리하게 됩니다. 그렇지 않고, $\text{switch}(r) = 1$ 의 조건이면 B씨가 승리합니다.

switch(r)을 연산하고 그 결과를 온체인으로 SC에 보고하는 실행파일 exec를 실행함으로써 기본적인 믹시클을 하이브리드 컨트랙트로 구현할 수 있습니다. 이 구조를 정리하면 그림 11과 같습니다.



- 그림 11: 예시 2에서의 기본 믹시클 구조. 보고서 r에 대한 온체인 비밀과 해당 바이너리 옵션의 기저 자산을 제공하기 위해 오라클은 이진값인 switch(r)만을 스위치(Switch)를 통해 컨트랙트 SC에 보냅니다.

부록 C.3에서는 DON에서 이 목표를 쉽게 달성할 수 있게 해주는 어댑터 ConfSwitch를 설명했습니다. ConfSwitch에 토대가 되는 기본적인 발상은 아주 간단합니다. ConfSwitch는 r의 값을 보고하는 대신 이진 스위치 값 switch(r)만을 보고합니다. SC는 switch(r)에만 근거하여 올바른 지분을 하도록 설계될 수 있으며, switch(r) 그 자체는 그 기저의 자산(본 예시에서 CBT에 해당)에 대한 정보를 공개하지 않습니다. 또한, 어댑터 ConfSwitch는 감사자의 공개키 pk_aud에 의해 암호화된 원장에 (q, r)에 대한 암호문을 배치하여 기밀성을 유지하는 감사 추적을 생성합니다.

본 예시에 대한 설명을 단순화하기 위해 선택한 기본 믹시클은 해당 예시에서 바이너리 옵션 뒤에 자산과 배팅만을 숨깁니다. 완전형 믹시클[135]은 두 가지 형태의 기밀성을 제공할 수 있습니다. 즉, 관찰자에게 (1) 플레이어가 베팅한 이벤트(<q: 및 r> 정보), (2) 해당 베팅에서 승리한 플레이어를 숨깁니다.

믹시클이 MAINCHAIN에서 실행되기 때문에 한 플레이어가 DON에서 MAINCHAIN으로 switch(r)을 중계해야 하거나, ConfSwitch에 의해 결과수 트리거되며 또 다른 어댑터에게 switch(r)을 MAINCHAIN에 전달하도록 요청하는 실행파일 exec이 생성될 수 있습니다.

세 번째로 모호한 형식의 기밀성 또한 고려할 가치가 있습니다. ConfSwitch의 기본 구현에서는 O은 DON에서 어댑터를 실행함으로써, 자산(이 예시의 경우 CBT에 해당)과 바이너리 옵션의 특성을 학습합니다. 그러나 부록 C.3에서 논의한 바와 같이 DECO 또는 타운크라이어를 사용하면 이 정보를 O로부터 추가적으로 숨길 수 있습니다. 이 경우, O은 SC의 공개 관찰자 외에 어떤 정보도 학습하지 않습니다.

믹시클에 대한 상세 정보는 [135]를 참조하시기 바랍니다.

5. 공정한 시퀀싱 서비스

DON의 네트워킹, 연산 및 저장 기능을 활용하여 제공할 것으로 기대하는 중요한 서비스 중 하나는 공정한 시퀀싱 서비스(Fair Sequencing Services, FSS)입니다. FSS는 단순히 DON 프레임워크 내에서 구현된 애플리케이션으로 볼 수 있지만, 블록체인 전반에 걸쳐 수요가 높을 것으로 판단되며 체인링크 네트워크가 적극적으로 지원할 것으로 예상됩니다.

오늘날 많은 디파이 애플리케이션은 공개 블록체인 네트워크에서 실행될 때, 기존 시장에 만연한 일종의 내부자 누출 및 조작 기회와 유사하게 사용자가 자신의 이익을 위해 악용할 수 있는 정보를 노출합니다[64, 155]. 반면, FSS는 공정한 디파이 생태계를 가는 길을 열어줍니다. FSS는 개발자가 정보 유출로 인한 악용 조작에서 보호받는 디파이 컨트랙트를 구축할 수 있도록 지원합니다. 아래에 설명된 문제를 감안할 때 FSS는 레이어-2 서비스에 특히 잘 어울리며 섹션 6에서 논의하게 될 이러한 서비스의 프레임워크에 적합합니다.

과제: 기존의 무허가 시스템에서 트랜잭션은 전적으로 채굴자의 재량에 따라 주문됩니다. 허가된 네트워크에서 검증자 노드는 동일한 권한을 행사할 수 있습니다. 이는 탈중앙화된 시스템에서 크게 인정되지 않는 일시적 중앙 집중화의 한 형태입니다. 채굴자는 자신의 이익을 위해 트랜잭션을 (일시적으로) 결렬하거나[171] 자신의 이익을 극대화하기 위해 재정렬할 수 있으며, 이를 채굴자 추출 가능 가치(miner-extractable value, MEV)[90]라고 합니다. MEV라는 용어는 오해의 소지가 있는데, 이는 채굴자가 포함할 수 있는 가치를 의미하는 것은 오히려 비탈중앙화입니다. 본질적으로 MEV는 일반 사용자가 관찰할 수 있습니다. 그러나 채굴자는 일반 사용자가 보다 많은 수량을 가지고 있으며 MEV는 대체로 적대적 재정렬 및 보안 트랜잭션의 인계전을 통해 모든 엔티티가 얻을 수 있는 가치의 상한을 포함합니다. 채굴자가 단순히 수수료(가스)를 기준으로 트랜잭션을 조작 없이 정렬하더라도 사용자 스스로가 수수료 가격을 조작하여 덜 복잡한 트랜잭션 보다 유리한 트랜잭션을 할 수 있습니다.

Daian et al. [90]에서는 봇(채굴자가 아님)이 오늘날 디파이 시스템 사용자의 해를 끼치는 방식으로 수수료 역학을 활용하는 방식과 MEV가 블록체인의 기본 컨센서스 보장의 안정성을 위협하는 방식을 제시하고 주장합니다. 트랜잭션 순서 조작의 다른 사례는 지주 확인입니다([50, 154]).

롤업과 같은 새로운 트랜잭션 처리 방법은 처리량이 많은 블록체인의 확장 문제에 대한 매우 유망한 접근 방식입니다. 그러나 이러한 방법은 MEV 문제를 완전히 해결하지 않습니다. 대신 해당 트랜잭션 묶음을 생성하는 엔티티인 채굴자에 의한 재정렬이 없다고 가정합니다. 스마트 컨트랙트의 운영자는 유효성 증명이 있는 <tx_i>를 제공하는 사용자라는 해당 엔티티는 트랜잭션을 주문하고 실행할 권한을 갖습니다. 즉, 동일한 MEV를 통한 추출 가능 값(Rollup-extractable Value, REV)으로 남깁니다.

MEV는 맴볼에 제출되었지만 아직 체인에 커밋되지 않은 예정된 트랜잭션에 영향을 미칩니다. 이러한 트랜잭션에 관한 정보는 네트워크에서 광범위하게 사용할 수 있습니다. 따라서 채굴자, 검증자, 일반 네트워크 참여자가 이 지식을 활용하고 숨겨진 재정렬을 생성할 수 있습니다. 또한 재정렬은 채굴자가 자신의 이익을 위해 트랜잭션 순서에 영향을 계속 줄 수 있고 이들 자신에게 유리하게 이용할 수 있습니다.

컨센서스 프로토콜은 트랜잭션 순서에 대한 리더의 과도한 영향을 막고자 하는 1990년대의 분산처리 기술과 유사합니다[71, 109]. 이는 많은 사용자가 다른 사용자와 다양한 합의와 같은 일관적 속성에 의존할 수 있도록 보장합니다. 전통적 합의는 프로토콜 외 정확성(예: 아주 최근이거나) 속성과 경쟁할 때까지 비밀로 유지되는 트랜잭션 내용에 의존합니다.

공정 시퀀싱 서비스(Fair Sequencing Services, FSS) 개요: DON은 신뢰되는 체인링크 네트워크를 통해, 트랜잭션 순서를 무작위로 결정하고 트랜잭션 묶음을 생성, 수정, 합의할 수 있는 프레임워크를 제공하여 온체인 컨트랙트 트랜잭션의 순서를 공정하게 할 수 있습니다. FSS는 기존의 블록체인 네트워크 위에서 실행되는 시스템에 대한 보호 계층으로 작용할 수 있습니다. 예를 들어, FSS는 특정 트랜잭션이 실행되는 순서의 무작위성을 추가하여 공격자가 예상할 수 없도록 합니다.

FSS는 체인링크 네트워크의 노드들이 협력하여 온체인 트랜잭션을 무작위화하고 집계하는 프로세스를 사용합니다. FSS는 노드들이 MAINCHAIN에 전달하는 모든 트랜잭션의 순서를 무작위화하고, 동시에 집계된 트랜잭션 묶음을 MAINCHAIN에 보냅니다. 이 과정은 공정성을 보장하고 특정 사용자가 유리하게 트랜잭션 순서를 조작할 수 없도록 합니다.

FSS의 잠재적 이점은 다음과 같습니다.

- **순서의 공정성:** FSS는 개발자가 특정 컨트랙트에 대한 트랜잭션 투입이 풍부하고 기술적으로 정당한 사용자에게 불공평한 이점을 제공하지 않는 방식으로 공정하게 실행될 것을 보장할 수 있는 도구를 제공합니다. 순서와 정확성은 이러한 목적에 따라 지정할 수 있습니다.
- **정보 유출 감소 또는 제거:** FSS는 네트워크 참가자가 다가오는 트랜잭션에 대한 지식을 악용할 수 없도록 하여 트랜잭션이 커밋되기 전에 네트워크에서 사용할 수 있는 정보에 기반한 선행과 같은 공격을 줄이거나 제거할 수 있습니다. 이러한 누출의 악용을 방지하면 앞선 블록 전 트랜잭션에 의한 문제 및 위협이 줄어듭니다.
- **트랜잭션 처리 비용 감소:** FSS는 스마트 컨트랙트에 트랜잭션을 제출하는 순서를 공정하게 조정함으로써 플레이어의 요구를 제거함으로써 트랜잭션 처리 비용을 크게 줄일 수 있습니다.
- **우선순위화:** FSS는 중요한 트랜잭션에 자동으로 특정 순서 또는 특별한 우선순위를 부여할 수 있습니다. 예를 들어 [79]와 같이 오라클 보고서에 대한 선행 공격을 방지하기 위해 FSS는 오라클 보고서를 소급하여 트랜잭션 스트림에 삽입할 수 있습니다.

DON에서 FSS의 가장 중요한 목표는 디파이 생성자가 공정한 금융 시스템을 실현할 수 있도록 권한을 부여하는 것입니다. 여기서 공정한 시스템이란 속도, 내부 지식, 기술적 조작을 수행하는 능력에 있어 특정 사용자(또는 채굴자)를 유리하게 처리하지 않는 시스템을 의미합니다. 공정성에 대한 명확하고 일반적인 개념은 이해하기 어렵고 합리적으로 완벽한 공정성은 달성할 수 없지만, FSS는 개발자가 디파이에 강력한 도구 트를 제공하여 설계 목표를 달성하는 데 도움이 되는 정책을 시행할 수 있도록 지원하는 것을 목표로 합니다.

FSS의 주요 목표는 DON이 대상으로 하는 MAINCHAIN에 대해 공정한 시퀀싱 서비스로 작동하는 것이지만 FSS가 보장하는 동일한 공정성 요구 사항 중 일부는 DON 당사자 간에 실행되는 (탈중앙화) 프로토콜에도 적합할 수 있습니다. 따라서 FSS는 MAINCHAIN 사용자가 보낸 트랜잭션뿐만 아니라 다른 DON 노드 간에 공유되는 트랜잭션(즉, 메시지)을 공정하게 순서화하기 위해 DON 노드의 하위 집합이 제공하는 서비스라고 폭넓게 이해할 수 있습니다. 이 선에서는 주로 MAINCHAIN 트랜잭션을 시퀀싱하는 목표에 초점을 맞출 것입니다.

섹션 구성 : 5.1절에서 우리는 FSS의 설계에 동기를 부여하는 두 가지 고차원적 애플리케이션인 오라클 보고서의 선행 방지 및 사용자 트랜잭션의 선행 방지에 대해 설명합니다. 이후 섹션 5.2에서 FSS 설계에 대한 자세한 내용을 제공합니다. 섹션 5.3에서는 공정한 주문 보장 및 이를 달성하기 위한 수단을 제시합니다. 마지막으로 섹션 5.4 및 섹션 5.5에서는 이러한 정책에 대한 네트워크 수준의 위협인 네트워크 플러딩 (flooding) 및 시빌 공격과 해결 방안에 관해 설명합니다.

5.1 선행 문제

FSS의 목표와 설계를 설명하기 위해 두 가지 일반적인 형태의 선행(front-running) 공격과 기존 솔루션의 한계에 대해 설명합니다. 선행은 트랜잭션 순서 공격의 한 종류를 가리킵니다. 여기서 다루지는 않지만, FSS가 도움을 줄 수 있는 후행 및 샌드위치(선행과 후행의 조합)[237]와 같은 여러 관련 공격도 존재합니다.

5.1.1 오라클의 선행(front-running)

블록체인 애플리케이션에 오프체인 데이터를 제공하는 전통적인 그 역할에 있어 오라클은 자연스럽게 선행 공격의 대상이 됩니다.

오라클을 사용하여 온체인 거래소에 다양한 가격 피드를 제공하는 일반적인 설계 패턴을 보면, 오라클은 주기적으로(예: 매시간) 다양한 자산에 대한 가격 데이터를 수집하고 이를 교환 컨트랙트로 보냅니다. 이러한 가격 데이터 트랜잭션은 명백한 차익 거래의 기회를 제공합니다. 예컨대, 최신 오라클 보고서가 일부 자산에 대해 훨씬 더 높은 가격을 나열하는 경우 공격자는 오라클 보고서를 선행하여 자산을 구매하고 오라클 보고서가 처리되는 데로 즉시 재판매할 수 있게 됩니다.

속도 제한(speed bump) 및 가격 소급: 오라클 선행 매매에 대한 자연스러운 해결책은 오라클 보고서에 다른 트랜잭션보다 특별한 우선순위를 부여하는 것입니다. 예컨대, 오라클 보고서는 채굴자들이 먼저 접하도록 높은 수수료를 매겨 전송할 수 있습니다. 그러나 차익 거래 기회가 높을 경우에는 이 방법으로도 선행 매매를 막을 수는 없으며, 채굴자들이 자발적으로 하는 차익 거래도 방지할 수 없습니다.

따라서 일부 거래소는 처리 전 여러 블록에 사용자 트랜잭션을 대기시키거나, 새로운 오라클 보고서가 도착할 때 가격을 소급하여 조정하는 것과 같이 더 엄한 속도 제한 조치인 이른바 ‘스피드 범프(speed bump)’를 구현하는 데 노력하고 있습니다. 이 방법의 단점은 자산 교환이 상당한 시간이 지난 후에야 확인되기 때문에 교환을 구현하는 것이 더 복잡해지고 스토리지 요건과 이에 따른 트랜잭션 비용이 증가하며 사용자 경험을 해치게 된다는 것입니다.

피기백킹(Piggybacking): FSS로 넘어가기 전에 오라클의 선행 문제에 대한 아주 간단하고 우아한 솔루션인 피기백킹에 대해 논의하겠습니다. 그러나 이 방법은 다른 시나리오에서의 선행 처리에는 적용할 수 없습니다.

간단히 말하면, 오라클은 주기적으로 온체인 컨트랙트에 보고서를 보내는 대신 사용자가 온체인 자산을 매매할 때 트랜잭션에 추가하는 서명된 보고서를 게시합니다. 거래소는 해당 보고서가 유효하고 최신 상태인지 확인하고(예: 오라클은 보고서가 유효한 블록 범위에 서명할 수 있음) 관련 가격 피드를 추출합니다.

이 간단한 접근 방식은 위의 ‘스피드 범프’ 접근 방식에 비해 여러 가지 이점이 있습니다. (1) 교환 컨트랙트는 가격 피드의 상태를 유지할 필요가 없으므로 트랜잭션 비용이 낮아집니다. (2) 오라클 보고서가 필요에 따라 체인에 게시됨에 따라 오라클은 더 빈번한 업데이트(예: 매 분)를 생성할 수 있으므로 보고서를 미리 실행하는 차익 거래 기회를 최소화할 수 있습니다. (3) 트랜잭션은 항상 새로운 가격 피드를 포함하므로 즉시 검증할 수 있습니다.

그러나 이 접근 방식은 완벽하지 않습니다. 첫째, 이 방법은 최신 오라클 보고서를 가져와 자신의 트랜잭션에 첨부할 책임을 거래소 사용자에게 전가합니다. 둘째, 이 방법은 차익 거래 기회를 최소화하지만, 온체인 컨트랙트의 활성에 영향을 미치지 않고는 이를 완전히 방지할 수 없습니다. 실제로 오라클 보고서가 일부 블록 번호 n 까지 유효할 경우, 블록 $n + 1$ 에 제출된 트랜잭션에는 유효한 새 보고서가 필요하게 됩니다. 오라클에서 사용자에게 보고서를 전파하는 과정에서 고유한 지연에 따라 블록 $n + 1$ 에 유효한 새 보고서는 블록 $n + 1$ 의 채굴 전, 즉 블록 $n - k$ 에서 일정 기간 공개되어야 하며 이를 통해 단기 차익 거래 기회가 존재하는 블록 k 의 시퀀스를 생성하게 됩니다. 아래에서는 FSS를 통해 이러한 제약을 해소하는 방식을 설명합니다.

FSS로 오라클 보고서 우선순위 지정: FSS는 앞서 살펴본 피기백킹 솔루션에 기반하여 오라클의 선행 문제를 해결할 수 있지만, 오라클 보고서를 사용하여 트랜잭션을 증대하는 추가 작업을 탈중앙화 오라클 네트워크에 푸시합니다.

높은 수준에서, 오라클 노드는 온체인 거래소로 향하는 트랜잭션을 수집하고 실시간 가격 피드에 동의하며 수집된 트랜잭션과 함께 가격 피드를 메인 체인 컨트랙트에 게시하게 됩니다. 개념적으로 보면 이 접근 방식은 ‘데이터 증강형 트랜잭션 배치 작업’이라 생각할 수도 있습니다. 여기서 오라클은 최신 가격 피드가 항상 트랜잭션에 추가되도록 합니다.

FSS 솔루션은 거래소 사용자에게 투명하게 구현될 수 있으며 컨트랙트의 로직에 최소한의 변경만을 가합니다. 이에 대해서는 섹션 5.2에서 자세히 설명합니다. 새 오라클 보고서가 항상 사용자 트랜잭션보다 높은 우선순위를 갖는지 확인하는 것은, FSS가 채택하고 시행할 수 있는 주문 정책의 일례에 지나지 않습니다. 주문 공정성을 보장하기 위한 FSS의 정책에 대해서는 섹션 5.3에서 보다 일반적으로 설명하겠습니다.

5.1.2 선행 매매 사용자 트랜잭션

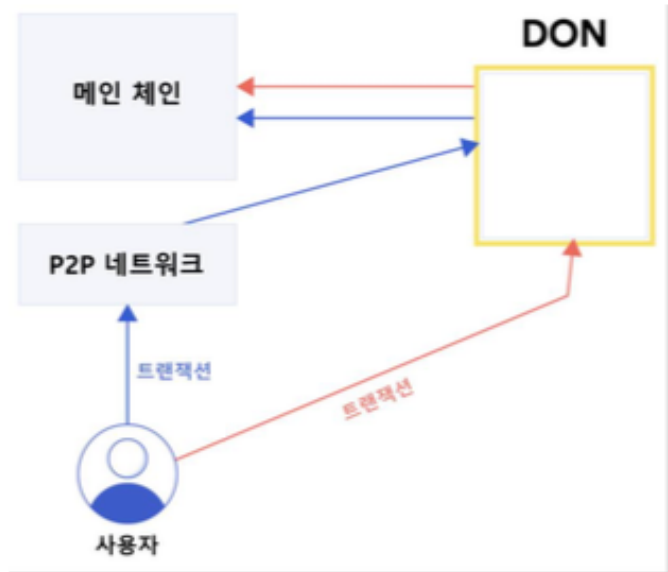
이제 위의 방어 방법이 작동하지 않는 일반 애플리케이션에서의 선행 매매를 다루겠습니다. 다음 시나리오는 이 문제를 광범위하게 보여줍니다. 공격자는 P2P 네트워크로 전송된 일부 사용자 트랜잭션 $tx1$ 을 보고 자신의 적대적 트랜잭션 $tx2$ 를 인젝션하여 $tx2$ 가 $tx1$ 보다 먼저 처리되도록 합니다(예: 더 높은 트랜잭션 수수료를 지불하게 하는 방식 사용). 예컨대, 이러한 종류의 선행 매매는 디파이 시스템[90]에서 차익 거래 기회를 악용하고 다양한 디앱[101]의 사용자에게 영향을 미치는 봇 사이에서 흔하게 확인됩니다. 블록체인에서 처리되는 트랜잭션 사이에 공정한 순서를 부여하면 이 문제가 해결됩니다.

보다 근본적으로, $tx1$ 의 부 사항을 보는 것은 때때로 필요하지도 않으며, 단순히 존재한다는 사실만 알 수 있어도 적은 자신의 $tx2$ 를 통해 $tx1$ 에 선행 공격을 하고 $tx1$ 을 생성한 무고한 사용자를 속일 수 있습니다. 예컨대, 이 사용자는 정기적으로 특정 자산을 거래하는 것으로 알려져 있을 수 있습니다. 이러한 공격을 방지하려면 메타데이터의 유출까지 방지하는 완화 조치가 필요합니다[62]. 이 문제에 대한 몇 가지 솔루션이 있지만 지연 및 사용성 문제가 수반됩니다.

FSS를 네트워크 주문에서 최종 주문까지 사용: 기존 시스템에는 트랜잭션이 체인상에 나타나는 순서에서 이벤트의 순서와 네트워크 외부의 정보 흐름을 존중하도록 보장하는 메커니즘이 부재하기 때문에 선행의 기회가 존재합니다. 이는 블록체인에서 애플리케이션(예: 거래 플랫폼) 구현상의 결함에 따라 발생하는 문제를 의미합니다. 이상적으로는 트랜잭션이 생성되어 블록체인의 P2P 네트워크로 전송된 것과 동일한 순서로 블록체인에서 커밋되도록 해야 합니다. 그러나 블록체인 네트워크가 탈중앙화되어 있기 때문에 그러한 순서는 포착 불가능합니다. 따라서 FSS는 블록체인 네트워크의 분산 특성에만 기인하여 발생하게 되는 공정성

위반에 대한 보호를 제공하는 메커니즘을 도입합니다.

5.2 FSS 상세 정보



- 그림 12: 두 가지 서로 다른 트랜잭션 경로(직접 및 멤풀 기반)가 있는 공정 순서 멤풀.
그림 12는 FSS의 일반적인 구성을 보여줍니다. 공정성을 보장하기 위해 FSS를 제공하는 DON은 MAINCHAIN에 들어갈 때 트랜잭션의 흐름을 방해해야 합니다. 클라이언트, MAINCHAIN의 스마트 컨트랙트 또는 둘 다에 대한 조정이 필요할 수 있습니다. 높은 수준에서 볼 때, FSS에 의한 트랜잭션 처리는 아래에 설명된 (1) 트랜잭션 모니터링 (2) 트랜잭션 순서 (3) 트랜잭션 게시의 단계로 구분할 수 있습니다. 트랜잭션 시퀀싱에 사용되는 주문 방법에 따라 다음 섹션에서 설명하는 것처럼 추가 프로토콜 단계가 필요합니다.

5.2.1 트랜잭션 처리

트랜잭션 모니터링: 우리는 FSS가 특정 스마트 컨트랙트를 대상으로 하는 사용자 트랜잭션을 모니터링하기 위해 두 가지 다른 접근 방식을 제시합니다.

- 직접 방식:** 직접 접근 방식은 개념적으로 가장 간단하지만, 트랜잭션이 메인 체인의 노드가 아닌 탈중앙화 오라클 네트워크 노드로 직접 전송되도록 사용자 클라이언트를 변경해야 합니다. DON은 특정 스마트 컨트랙트 SC로 향하는 사용자 트랜잭션을 수집하고 일부 순서화 정책에 따라 정렬합니다. 그런 다음 DON은 정렬된 트랜잭션을 메인 체인의 스마트 컨트랙트로 보냅니다. 일부 순서화 메커니즘의 경우 트랜잭션을 생성하는 사용자가 FSS로 전송하기 전에 암호화를 통해 보호해야 하므로 직접 접근 방식이 요구됩니다.
- 멤풀 기반 방식:** FSS와 레거시 클라이언트의 통합을 용이하게 하기 위해 DON은 멤풀 서비스(MS)를 사용하여 메인 체인의 멤풀을 모니터링하고 트랜잭션을 수집할 수 있습니다.

직접 전송 방식은 많은 컨트랙트에서 선호될 가능성이 크며, 많은 경우 상당히 실용적이어야 할 것으로 판단됩니다.

양호한 사용자 경험을 유지하면서 직접 전송을 지원하도록 기존 디앱을 최소한으로 수정하는 방법에 대해 간략하게 논의하겠습니다. 우리는 오늘날 가장 인기 있는 선택이라는 점에서 ETH와 메타마스크 (MetaMask)[6]를 사용하는 접근 방식을 설명하곤 하는데, 사실 언급된 기술은 다른 체인과 지갑으로 확장되어야 합니다. 최근의 ETH 개선안인 **EIP-3085(지갑에 ETH 체인 RPC 추가 방법)**[100]을 사용하면 메타마스크 및 기타 브라우저 기반 지갑에서 사용자 지정 ETH 체인(리플레이 공격을 방지하기 위해 MAINCHAIN과 다른 것을 사용)을 쉽게 대상으로 지정할 수 있습니다. 이 제안을 구현한 후 DON을 사용하려는 디앱은 ETH 호환 API를 노출하는 모든 DON에 트랜잭션을 직접 전송할 수 있도록 프론트 엔드에 단일 메서드 호출을 추가하기만 하면 됩니다.

한편, **EIP-712(ETH 유형의 구조화 데이터 해싱 및 서명)**[49]에서는 디앱 사용자가 메타마스크를 사용하여 DON 트랜잭션을 지정하는 구조화된 데이터에 서명할 수 있도록 약간 더 복잡하지만 이미 널리 배포된 대안을 제공합니다. 이 디앱은 이 서명된 구조화된 데이터를 DON으로 보낼 수 있습니다.

마지막으로 하이브리드 접근 방식도 가능합니다. 예를 들어 레거시 클라이언트는 계속해서 메인 체인의 멤풀로 트랜잭션을 보낼 수 있지만 중요한 트랜잭션(예: 오라클 보고서)은 DON 노드(특히 가격 피드 업데이트 및 오라클 보고서와 같은 오라클 보고서를 제공하는 노드 집합으로 FSS를 제공하는 노드 집합은 중복되거나 동일할 수 있음)로 직접 전송됩니다.

트랜잭션 시퀀싱: FSS의 주요 목적은 사용자 트랜잭션이 미리 정의된 정책에 따라 정렬되도록 보장하는 것입니다. 이 정책의 성격은 애플리케이션의 요구 사항과 방지하려는 불공정한 트랜잭션 순서화 유형에 따라 다릅니다.

DON의 FSS는 데이터를 처리하고 로컬 상태를 유지할 수 있으므로 오라클에서 사용할 수 있는 정보를 기반으로 임의의 시퀀싱 정책을 부과할 수 있습니다. 특정 주문 정책과 그 구현 방안에 대해서는 섰 5.3에서 논의하겠습니다.

트랜잭션 게시: 사용자로부터 직접 받거나 멤풀에서 수집한 사용자 트랜잭션을 수집 및 정렬한 후 DON은 이러한 트랜잭션을 메인 체인으로 보냅니다. 따라서 메인 체인과 DON의 상호 작용은 메인 체인의 채굴자가 통제하는 (불공정할 가능성이 있는) 트랜잭션 순서화에 따른 영향을 받습니다. 탈중앙화된 트랜잭션 정렬의 이점을 활용하려면 대상 스마트 컨트랙트 SC는 DON을 ‘일등급(first-class)’ 시민으로 취급하도록 설계되어야 합니다. 두 가지 접근 방식이 확인됩니다.

- **DON 전용 컨트랙트 :** 가장 간단한 설계 옵션은 메인 체인 스마트 컨트랙트 SC가 DON에 의해 처리된 트랜잭션만 수락하도록 하는 것입니다. 이를 통해 스마트 컨트랙트는 DON이 제안한 순서대로 트랜잭션을 처리하지만 사실상 스마트 컨트랙트가 위원회 기반 시스템에서 작동하도록 제한합니다(즉, DON 위원회는 이제 트랜잭션의 순서와 포함 여부를 결정할 수 있는 지속적인 권한을 갖게 됨).
- **이중 클래스 컨트랙트 :** 선호되는 더 밀한 설계 방안에서는 메인 체인 스마트 컨트랙트 SC가 DON과 레거시 사용자 모두에서 발생하는 트랜잭션을 수락하되10 DON에서 처리되지 않은 트랜잭션에 전통적인 ‘스피드 댐프’를 설정합니다. 예컨대, DON의 트랜잭션은 즉시 처리될 수 있지만 레거시 트랜잭션은 고정된 기간 동안 스마트 컨트랙트에 의해 ‘버퍼링’됩니다. 커밋 표시 방식이나 VDF[53]와 같은 선행을 방지하기 위한 다른 표준 메커니즘도 레거시 트랜잭션에 적용할 수 있습니다. 이는 해당

DON에서 트랜잭션을 검열할 원치 않는 권한을 부여하지 않고 DON이 주문한 트랜잭션이 협의된 순서에 따라 처리되도록 합니다.

FSS에서 트랜잭션 순서를 부과하려면 트랜잭션이 ‘오프체인’으로 집계되어야 하므로, 이 솔루션은 온체인 처리 비용을 줄이는 것을 목표로 하는 다른 집계 기술과 자연스럽게 결합될 수 있습니다. 예컨대, DON은 트랜잭션을 수집 및 정렬한 다음 이러한 트랜잭션을 하나의 ‘배치 트랜잭션’(예: 롤업)으로 메인 체인에 보낼 수 있으므로 총 트랜잭션 수수료를 줄일 수 있습니다.

트랜잭션 주문의 집행: DON 전용 디자인이든 이중 클래스 디자인이든 DON의 트랜잭션 순서가 유지되도록 하기 위해서는 메인 체인 스마트 컨트랙트 SC와 DON을 공동으로 설계해야 합니다. 우리는 이와 관련해서도 다양한 설계 옵션을 구상하고 있습니다.

- **시퀀스 번호 :** DON은 각 트랜잭션에 시퀀스 번호를 추가하고 이러한 트랜잭션을 메인 체인의 맴폴로 보낼 수 있습니다. 메인 체인 스마트 컨트랙트 SC는 ‘순서에 벗어난’ 트랜잭션을 무시합니다. 우리는 이 설정에서 메인 체인 채굴자가 DON의 트랜잭션 순서를 무시하기로 할 수 있으므로 트랜잭션이 실패할 수 있다는 점에 주목합니다. 이는 TCP가 누락된 패킷이 수신될 때까지 순서가 잘못된 패킷을 버퍼링하는 방식과 다소 유사하게, SC가 올바른 트랜잭션 순서를 적용하도록 (비싼) 상태를 유지함으로써 실현할 수 있습니다.
- **트랜잭션 논스 (nonce):** 많은 블록체인, 특히 ETH의 경우 위의 시퀀스 번호 지정 방식은 내장된 트랜잭션 논스를 활용하여 메인 체인의 스마트 컨트랙트 SC가 트랜잭션을 순서대로 처리하도록 할 수 있습니다. 여기서 DON 노드는 DON 노드 간에 공유되는 키로 보호되는 단일 메인체인 계정을 통해 메인 체인으로 트랜잭션을 보냅니다. 해당 계정의 트랜잭션 논스는 트랜잭션이 올바른 순서로 채굴되고 처리되도록 합니다.
- **트랜잭션 집계 :** DON은 롤업(또는 롤업과 유사한 번들)에서 여러 트랜잭션을 집계할 수 있습니다. 메인체인 스마트 컨트랙트는 이러한 집계 트랜잭션을 처리하도록 설계되어야 합니다.
- **메인 체인 프록시로 트랜잭션 집계 :** 여기에서 DON은 유사하게 트랜잭션을 메인 체인에 대한 하나의 ‘메타 트랜잭션(meta-transaction)’으로 묶지만 트랜잭션의 압축을 풀고 대상 컨트랙트 SC에 중계하기 위해 사용자 지정 프록시 스마트 컨트랙트에 의존하게 됩니다. 이 기술은 레거시 호환성 차원에서 유용할 수 있습니다. 메타 트랜잭션은 롤업처럼 작동하지만 메인 체인에 한 번 게시된 압축되지 않은 트랜잭션 목록으로 구성된다는 점에서 다릅니다.

마지막 설계 옵션은 DON의 목표 컨트랙트 SC에 도달하기 전에 메인 체인 컨트랙트를 통해 프록시되는 사용자 트랜잭션을 원활하게 지원하는 이점을 제공합니다. 예컨대, 일부 지갑 컨트랙트에 트랜잭션을 보내고 SC에 내부 트랜잭션을 보내는 사용자를 가정해 보겠습니다. 사용자의 지갑 컨트랙트가 모든 내부 트랜잭션의 시퀀스 번호를 SC로 전달하도록 특별히 설계되지 않는 한, 이러한 트랜잭션에 시퀀스 번호를 할당하기란 까다로울 것입니다. 마찬가지로 이러한 내부 트랜잭션은 SC로 직접 전송되는 메타 트랜잭션으로 쉽게 집계될 수 없습니다. 아래에서는 이러한 프록시 트랜잭션에 대한 설계 고려 사항에 대해 추가적으로 논의하겠습니다.

5.2.2 트랜잭션의 원자성

지금까지의 논의에서 우리는 트랜잭션이 단일 온체인 스마트 컨트랙트와 상호 작용(예: 사용자가 거래소에 구매 요청을 보내는 경우 등)한다고 암묵적으로 가정했습니다. 그러나 ETH와 같은 시스템에서 단일 트랜잭션은 여러 내부 트랜잭션으로 구성될 수 있습니다(예: 하나의 스마트 컨트랙트가 다른 컨트랙트의 기능을 호출). 아래에서 우리는 트랜잭션의 원자성(atomicity)을 유지하면서 ‘다중 컨트랙트’ 트랜잭션을 시퀀싱(즉, 트랜잭션에 의해 규정된 일련의 작업이 모두 올바른 순서로 실행되거나 전혀 실행되지 않도록)하기 위한

두 가지 높은 수준의 전략에 대하여 설명합니다.

강력한 원자성: 가장 간단한 해결책은 위에서 설명한 FSS를 전체 ‘다중 컨트랙트’ 트랜잭션에 직접 적용하는 것입니다. 즉, 사용자는 자신의 트랜잭션을 네트워크로 보내고 FSS는 이러한 트랜잭션을 모니터링 및 시퀀싱 처리하여 메인 체인에 게시합니다.

이 접근 방식은 기술적으로 간단하지만 한 가지 잠재적인 제약이 존재합니다. 사용자 트랜잭션이 SC1과 SC2의 두 컨트랙트와 상호 작용하고 둘 다 공정한 시퀀싱 서비스를 활용하려는 경우 이 두 컨트랙트의 시퀀싱 정책은 일관되어야 합니다. 즉, SC1과 SC2 모두와 각각 상호 작용하는 두 개의 서로 다른 트랜잭션 tx_1 과 tx_2 가 있을 경우, 이는 SC1의 정책이 tx_2 보다 먼저 tx_1 을 주문하고 SC2의 정책은 그 반대의 순서를 규정하는 것이 아니어야 합니다.

대부분 관심 시나리오에 대해 우리는 서로 다른 컨트랙트에서 채택한 시퀀싱 정책이 일관될 것이라고 간주합니다. 예컨대, SC1과 SC2 둘 다 멤풀에서의 대략적인 도착 시간까지 해당 트랜잭션이 정렬되기를 원할 수 있으며, SC1의 경우 특정 오라클 보고서가 항상 먼저 전달되기를 추가로 기대할 수도 있습니다. 후자의 오라클 보고서 트랜잭션은 SC2와 상호 작용하지 않으므로 정책은 일관됩니다.

약한 원자성: 일반적으로 FSS는 개별 내부 트랜잭션 수준에서 적용될 수 있습니다.

일부 초기 트랜잭션 $\tilde{tx}_{pre}$ 로 구성된 $tx = \{\tilde{tx}_{pre}, \tilde{tx}_{SC}, \tilde{tx}_{post}\}$ 형식의 트랜잭션을 고려해 보겠습니다. 이 트랜잭션은 SC에 대한 내부 트랜잭션 $\tilde{tx}_{SC}$ 를 도출하여 내부 트랜잭션(들) $\tilde{tx}_{post}$ 를 발행하게 됩니다. SC의 시퀀싱 정책은 SC로 전송된 다른 트랜잭션과 관련하여 내부 트랜잭션 $\tilde{tx}_{SC}$ 가 어떻게 정렬되어야 하는지 결정할 수 있지만 $\tilde{tx}_{pre}$ 및 $\tilde{tx}_{post}$ 에 대한 시퀀싱 순서는 결정되지 않은 상태로 둡니다.

ETH와 같은 시스템에서의 트랜잭션 처리의 본질을 고려할 때 특정 내부 트랜잭션을 대상으로 하는 시퀀싱 서비스 개발은 간단한 문제가 아닙니다. 특수 설계된 컨트랙트 SC를 사용하면 다음과 같이 실현할 수 있습니다.

1. 트랜잭션 tx 는 네트워크로 전송되고 채굴됩니다(FSS에서 수행하는 시퀀싱 없음). 초기 $\tilde{tx}_{pre}$ 가 실행되고 $\tilde{tx}_{SC}$ 를 호출합니다.
2. SC는 $\tilde{tx}_{SC}$ 를 실행하지 않고 반환합니다.
3. FSS는 SC에 대한 내부 트랜잭션을 모니터링하고 순서를 지정하고 SC에 다시 게시합니다(즉, 트랜잭션 $\tilde{tx}_{SC}$ 를 SC에 직접 전송).
4. SC는 FSS로부터 접수된 트랜잭션 $\tilde{tx}_{SC}$ 를 처리하고, $\tilde{tx}_{SC}$ 에서 도출된 내부 트랜잭션 $\tilde{tx}_{post}$ 를 발행합니다.

이 접근 방식을 사용하면 트랜잭션이 완전히 원자적으로 실행되지 않지만(즉, 원래의 트랜잭션 tx 가 여러 온체인 트랜잭션으로 분할됨) 내부 트랜잭션의 순서는 유지됩니다.

이 솔루션에는 여러 설계 제약이 따릅니다. 예컨대, $\tilde{tx}_{pre}$ 는 $\tilde{tx}_{SC}$ 와 $\tilde{tx}_{post}$ 가 실행될 것이라고 가정할 수 없습니다. 또한 SC는 $\tilde{tx}_{SC}$ 와 $\tilde{tx}_{post}$ 가 FSS에서 보낸 트랜잭션임에도 불구하고 특정 사용자를 대신하여 트랜잭션을 수행할 수 있도록 설계해야 합니다. 이러한 이유로 앞서 설명한 바와 같은 덜 조밀한 ‘강력한 원자성’ 솔루션이

실제로 더 바람직할 수 있습니다.

여러 트랜잭션 및 각각의 내부 트랜잭션을 포함하는 보다 복잡한 종속성을 고려하기 위해 FSS의 트랜잭션 스케줄러에는 관계형 데이터베이스 관리자의 트랜잭션 관리자에서 볼 수 있는 것과 유사한 정교한 함수가 포함될 수 있습니다.

여러 트랜잭션 및 각각의 내부 트랜잭션을 포함하는 보다 복잡한 종속성을 고려하기 위해 FSS의 트랜잭션 스케줄러에는 관계형 데이터베이스 관리자의 트랜잭션 관리자에서 볼 수 있는 것과 유사한 정교한 함수가 포함될 수 있습니다.

5.3 공정한 트랜잭션 시퀀싱

여기에서 우리는 FSS에 의해 실현될 수 있는 트랜잭션 순서 지정 및 해당 구현에 대한 두 가지 개념인 순서 공정성(order-fairness)과 안전한 인과 보존(secure causality preservation)에 대해 논의합니다. 전자는 FSS에 의해 부과되며 후자는 FSS에 추가 암호화 방법론을 요구합니다.

순서 공정성: 순서 공정성은 Kelkar et al.[144]에 의해 공식적으로 처음 도입된 컨센서스 프로토콜의 시간적 공정성의 개념입니다.

Kelkar et al.의 경우, DON(또는 맴폴 기반 FSS의 경우 P2P 네트워크)이 처음 수신한 시간을 기준으로 트랜잭션이 정렬되는 일종의 자연 정책을 달성하는 것을 목표로 합니다. 그러나 탈중앙 시스템에서는 다른 노드에서 트랜잭션이 다른 순서로 도착하는 것을 볼 수 있습니다. 모든 트랜잭션에 대한 전체 순서를 설정하는 것은 MAINCHAIN의 기본 컨센서스 프로토콜이 해결해야 할 바로 그 문제입니다. Kelkar et al.[144]에서는 탈중앙화 오라클 네트워크의 도움으로 달성할 수 있는 ‘블록 순서 공정성(block-order fairness)’이라는 더 소극적인 개념을 도입합니다. 이는 일정 시간 동안 DON이 수신한 트랜잭션을 ‘블록’으로 그룹화하고 블록의 모든 트랜잭션을 동시에 동일한 위치(즉, 높이)에 있는 MAINCHAIN에 삽입합니다. 따라서 트랜잭션은 함께 정렬되고 서로 충돌 없이 병렬로 실행 가능해야 합니다. 넓은 의미에서 순서 공정성은 많은 노드의 비중이 트랜잭션 τ_1 를 τ_2 전에 본 다음, τ_1 의 시퀀스가 τ_2 보다 앞서거나 그와 같은 블록 내에 위치하도록 처리하는 것을 의미합니다. 트랜잭션 순서에 이러한 덜 조밀한 특성을 반영하면 선행 공격 및 기타 순서와 관련된 공격 기회가 크게 줄어듭니다.

Kelkar et al.에서는 동기, 부분 동기, 비동기 네트워크 설정을 포함하여 다양한 배포 모델을 처리하는 이른바 ‘애퀴타스(Aequitas)[144]’라는 프로토콜 계열을 제안합니다. 애퀴타스 프로토콜은 기본 BFT 컨센서스에 비해 상당한 통신 오버헤드를 부과하므로 실제 사용에는 이상적이지 않습니다. 그러나 우리는 FSS 및 기타 애플리케이션에서 트랜잭션 시퀀싱에 사용할 수 있는 애퀴타스의 실용적인 변형이 등장할 것이라고 믿습니다. 덜 형식적이고 속성이 약하지만, 실용성은 더 나은 일부 관련 체계(예: [36, 151, 236]과 같이)가 이미 제안된 바 있습니다. 이러한 체계는 FSS에서도 지원될 수 있습니다.

‘공정성’이라는 용어는 다른 블록체인 문헌에서 다른 의미로 사용된다는 점도 주목해야 합니다. 채굴자에게 있어서는 커밋된 자원에 비례하는 기회라는 의미에서의 공정성[106, 181]을, 검증자에게 있어서는 평등한 기회의 관점에서의 공정성[153]을 의미하기도 합니다.

안전한 인과 보존: 탈중앙 플랫폼에서 선행 및 기타 순서 위반을 방지하는 방법 중 가장 널리 알려진 접근 방식은 암호화 기술에 의존합니다. 이들은 공통적으로 트랜잭션 데이터 자체를 숨기고 컨센서스 레이어의 순서가 설정될 때까지 기다렸다가 나중에 처리를 위해 트랜잭션 데이터를 공개합니다. 이로써 블록체인에 의해 실행되는 트랜잭션 간의 인과적 순서를 유지합니다. 관련 보안 개념과 암호화 프로토콜은 블록체인이 도래하기 전에 상당 수준 개발된 바 있습니다[71, 190].

‘입력 인과 관계’[190] 및 ‘안전한 인과 관계 보존’[71, 97]의 보안 조건은 글로벌 질서 차원에서 이 트랜잭션의 위치가 결정되기 전에 트랜잭션에 대한 정보가 알려지지 않을 것을 공식적으로 요구합니다. 공격자는 그 때까지 암호학적으로 절대 정보를 추론할 수 없어야 합니다.

인과 관계를 유지하기 위한 암호화 기술로는 네 가지 기술을 꼽을 수 있습니다.

- **커밋-공개 (commit - reveal) 프로토콜 [29, 142, 145]:** 트랜잭션을 명시적으로 선언하는 대신 해당 트랜잭션에 대한 암호화 약정(Commitment)만 브로드캐스트합니다. 커밋되었지만 숨겨진 모든 트랜잭션이 정렬된 후(초기 블록체인 시스템의 경우에는 MAINCHAIN 자체 상에, 여기에서는 FSS에 의해 정렬됨), 발신자는 약정을 열고 미리 결정된 시간 간격 내에 트랜잭션 데이터를 공개해야 합니다. 그런 다음 네트워크는 이 개방이 이전 약정을 충족하는지 확인합니다. 이 방법의 기원은 블록체인이 출현하기 이전으로 거슬러 올라갑니다. 아주 간단한 방법이지만 상당한 단점을 갖고 있으며, 두 가지 이유로 사용하기가 쉽지 않습니다. 첫째, 주문 프로토콜 수준에서는 약정만 존재하므로 컨센서스 시 트랜잭션의 의미를 확인할 수 없습니다. 클라이언트까지 추가 왕복이 필요합니다. 더 심각한 것은 서비스 거부 공격(DoS)에 해당할 수 있는 개방이 전혀 이루어지지 않을 경우의 가능성입니다. 또한, 개방이 유효한지를 일관되고 탈중앙화된 방식으로 결정하는 것은 어렵습니다. 모든 참가자가 개방이 제 시간에 도착했는지 여부에 대해 동의해야만 하기 때문입니다.
- **복구 지연 커밋-공개 프로토콜 [145]:** 커밋-공개 접근 방식의 한 가지 문제는 클라이언트가 트랜잭션에 투기적으로 커밋하고 후속 트랜잭션이 수익성이 있는 경우에만 나중에 공개 가능하다는 점입니다. 최근 제안된 커밋-공개 접근 방식의 변형은 이러한 종류의 오작동에 대한 복원력을 향상시킵니다. 특히 TEX 프로토콜[145]은 암호화된 트랜잭션에 검증 가능한 지연 함수(Verifiable Delay Function, VDF)를 연산하여 얻을 수 있는 암호 해독 키를 포함하는 영리한 접근 방식을 통해 이 문제를 해결합니다[53, 221]. 클라이언트가 적시에 트랜잭션을 해독하지 못하면 시스템의 다른 사람들이 적당히 어려운 암호화 퍼즐을 풀어서 클라이언트를 대신해 해독합니다.
- **임계값 암호화 [71, 190]:** 이 방법은 DON이 임계값 암호화 작업을 수행할 수 있다는 점을 이용합니다. FSS가 암호화 공개키 pk_O 를 유지하고 오라클이 해당 비밀키를 공유한다고 가정합니다. 그런 다음 클라이언트는 pk 로 트랜잭션을 암호화하여 FSS로 보냅니다. FSS는 DON에서 트랜잭션을 주문한 다음 암호를 해독하고 최종적으로 고정된 순서로 MAINCHAIN에 인젝션합니다. 따라서 암호화는 정렬이 트랜잭션 내용에 기반하지 않아도 필요할 때 데이터 자체를 사용할 수 있도록 합니다. 이 방법은 원래 Reiter and Birman[190]에서 제안되었으며 후에 Cachin et al.[71]이 개량하여 허가된 컨센서스 프로토콜과 통합시킨 바 있습니다. 최근의 관련 작업에서는 일반 메시지[33, 97] 및 공유 데이터를 사용한 일반 연산[41]에 대한 컨센서스 수준 메커니즘으로 임계값 암호화의 사용을 탐구한 바 있습니다.
- **커밋된 비밀 공유 [97]:** DON이 보유한 키로 트랜잭션 데이터를 암호화하는 대신 클라이언트는 O 의 노드에 대해 비밀을 공유할 수도 있습니다. 연산 차원에서 안전한, 하이브리드 비밀 공유 체계를 사용

하는 이 트랜잭션은 먼저 무작위 키가 있는 대칭 암호를 사용하여 암호화됩니다. 해당 대칭키만 공유되고 암호문은 DON에 제출됩니다. 클라이언트는 별도로 암호화된 메시지를 사용하여 O 의 각 노드에 하나의 키 공유를 보내야 합니다. 나머지 프로토콜 단계는 트랜잭션 데이터가 공유에서 트랜잭션별 키를 재구성한 후 대칭 알고리즘으로 해독된다는 점을 제외하고 임계값 암호화와 동일합니다. 이 방법은 DON과 관련된 공개키 암호 시스템을 설정하거나 관리할 필요가 없습니다. 물론 클라이언트는 O 의 노드를 인식하고, 각 노드와 안전한 컨텍스트로 통신해야 하므로 클라이언트에는 추가적인 부담이 발생하게 됩니다.

이 암호화 방법은 제출된 트랜잭션에서 네트워크로 정보가 누출되는 것을 완벽하게 방지하지만, 메타데이터를 숨기지는 않습니다. 예컨대, 발신자의 IP 주소 또는 ETH 주소는 여전히 공격자가 선행 및 기타 공격을 수행하는 데 사용할 수 있습니다. 이러한 목표를 달성하려면 네트워크 레이어(예: [52, 95, 107]) 또는 트랜잭션 레이어(예: [13, 65])에 배치된 다양한 개인정보보호 강화 기술이 필요합니다. 특정 메타데이터의 특정 부분이 갖는 파급력, 즉 어떤 트랜잭션이 어떤 컨트랙트로 전송되는지의 문제는 동일한 DON에서 많은 컨트랙트를 다중화함으로써 (부분적으로) 숨길 수 있습니다. 트랜잭션을 암호학적으로 숨기는 행위 자체 또한 트랜잭션 발신자와 공모하여 손상된 DON 노드에 의한 트랜잭션의 우선순위를 지정하는 행위를 방지하지 않습니다.

암호화 프로토콜에 의해 보장되는 안전한 인과 관계는 모든 정책의 순서 공정성 보장을 보완하며 가능한 경우 두 가지 방법의 조합을 모색하게 될 것입니다. 공격자가 메타데이터를 관찰하여 상당한 이점을 얻을 수 없는 경우 안전한 인과 관계 보존 프로토콜을 단편적인 정렬 접근 방식과 병용할 수도 있을 것입니다. 예컨대, 오라클 노드는 트랜잭션을 받는 즉시 중복 없이 L 에 쓸 수 있습니다. 이때, 트랜잭션은 L 에 등장한 바에 순서가 지정되고 이후에 해독됩니다.

공정한 순서를 시행하는 데 도움이 되도록 하기 위해 TEE 사용 또한 고려할 계획입니다. 예컨대, Tesseract[44]는 일종의 인과적 순서를 달성하는 것으로 볼 수 있지만 기밀성을 유지하면서 트랜잭션을 명시적인 형식으로 처리하는 TEE 기능을 활용하여 강화된 방법입니다.

5.4 네트워크 레이어 고려 사항

지금까지 FSS에 대한 설명은 최종 트랜잭션 순서가 네트워크에서 관찰된 순서와 일치하도록 강제하는 문제에 중점을 두었습니다. 이하에서는 네트워크 레이어 자체에서 발생할 수 있는 공정성 문제에 대해 고려합니다.

기존 전자 시장의 고빈도 거래자는 우수한 네트워크 속도를 얻기 위해 상당한 자원을 투자하고[64], 디지털 자산 거래소 거래자도 이와 유사한 행태를 보입니다[90]. 네트워크 속도는 다른 당사자 트랜잭션을 관찰하고 경쟁 트랜잭션을 제출하는 데 이점을 제공합니다. 플래시 보이스(Flash Boys)[155]라는 책을 통해 알려진, 실제로 배포된 한 가지 해결 방안은 IEX 거래소에서 처음 도입[128]하고 후에 다른 거래소에도 (복잡한 결과[19])와 함께) 도입된[179] ‘스피드 범프’입니다. 이 메커니즘은 속도의 이점을 무력화할 목적으로 시장 액세스 지연(IEX의 경우 350ms)시킵니다. 경험적 증거(예: [128])에 따르면 이 조치는 일반 투자자의 특정 거래 비용을 줄이는 효과를 제공합니다. FSS는 들어오는 트랜잭션을 지연시키는 비대칭형 스피드 범프를 구현하는 데 간단히 사용할 수 있습니다.

Budish, Cramton, Shim[64]은 연속 시간 시장에서 속도의 장점을 활용하는 것이 불가피하다는 주장과 함께 일괄 경매 기반 시장 형식의 구조적 개선안을 제안합니다. 그러나 이 접근 방식은 기존 거래 플랫폼에서 광범위하게 채택되지 않았습니다.

기존 거래 시스템은 중앙화되어 있으며, 일반적으로 단일 네트워크 연결을 통해 트랜잭션을 수신합니다. 이와 대조적으로 탈중앙 시스템에서는 여러 유리한 지점에서 트랜잭션의 전파를 관찰할 수 있습니다. P2P 네트워크에서 네트워크 플러딩과 같은 동작을 관찰하는 것 또한 가능합니다. 우리는 개발자가 이러한 바람직하지 않은 네트워크 동작을 금지하는 정책을 지정할 수 있도록 지원하는 FSS에 대한 네트워크 레이어 접근 방식을 탐색하고자 합니다.

5.5 엔티티 수준의 공정 정책

순서 공정성과 안전한 인과 관계는 트랜잭션이 생성되어 네트워크에 처음 제출된 시간을 존중하는 트랜잭션 순서화를 구현하는 것을 목표로 합니다. 이러한 공정성 개념의 한계는 공격자가 많은 트랜잭션으로 시스템을 플러딩하여 이점을 얻게 되는 공격을 방지하지 못한다는 것입니다. 이는 토큰 판매[159]에서 효과적인 트랜잭션 저격을 수행하고 혼잡을 유발하여 담보부채 포지션(CDP)을 청산하는 방법으로 수행되는 전략입니다 [48]. 즉, 순서 공정성은 플레이어가 아닌 트랜잭션의 공정성을 강화합니다.

CanDID 시스템[160]에서 볼 수 있듯이, 개인정보를 보호하면서 다양한 형태의 시빌 저항을 달성하기 위해 노드 위원회(예: DON)와 함께 DECO 오라클 도구 또는 타운크라이어를 사용할 수 있습니다. 사용자는 자신의 신원을 공개하지 않고 신원을 등록하고 고유성에 대한 증거를 제공할 수 있습니다. 시빌 저항 자격 증명은 플러딩 공격의 기회를 제한하는 방식으로 트랜잭션 순서 지정 정책을 강화할 수 있는 접근 방식을 제공합니다. 예컨대, 토큰 판매에 등록된 사용자당 하나의 트랜잭션만 허용할 수 있습니다. 등록에는 사회 보장 번호와 같은 국가 식별자의 고유성 증명이 필요합니다. 이러한 접근 방식은 완벽하지는 않지만, 트랜잭션 플러딩 공격을 완화하는 데 유용한 정책이 될 수 있습니다.

6. DON 트랜잭션 실행 프레임워크(DON-TEF)

DON은 우리가 탈중앙화 오라클 네트워크 트랜잭션-실행 프레임워크(Decentralized Oracle Network Transaction-Execution Framework, DONTEF 또는 축약하여 TEF)라고 부르는 레이어-2 솔루션에 대한 오라클 및 탈중앙화 리소스 지원을 제공할 것입니다.

오늘날 디파이 컨트랙트에 대한 업데이트 빈도는 메인 체인 지연 시간(예: ETH의 10~15초 평균 블록 간격 [104])과 체인 및 제한된 연산/tx에 많은 양의 데이터를 푸시하는 비용에 의해 제한되며 이는 샤딩[148, 158, 232] 및 레이어-2 실행[5, 12, 121, 141, 169, 186, 187]과 같은 확장성 접근을 유발하고 있습니다. 훨씬 빠른 트랜잭션 시간을 가진 블록체인(예: [120])조차도 오프체인 연산을 포함하는 확장 전략을 제안한 바 있습니다[168]. TEF는 이러한 모든 레이어 1/MAINCHAIN 시스템에 대해 레이어-2 리소스 역할을 제공합니다.

TEF를 사용하여 DON은 메인 체인에서 제공하는 주요 신뢰 보증을 유지하면서 MAINCHAIN 컨트랙트에서 더 빠른 업데이트를 지원할 수 있습니다. TEF는 롤업, 낙관적 롤업(optimistic rollups)¹, 발리디움(Validium) 등을 비롯한 여러 레이어-2 실행 기법 및 패러다임과 DON 노드가 트랜잭션을 실행하는 임계값 신뢰 모델을 지원할 수 있습니다.

TEF는 FSS를 보완하며 지원하기 위한 것입니다. 즉, TEF에서 실행되는 모든 애플리케이션은 FSS를 사용할 수 있습니다.

6.1 TEF 개요

TEF는 고성능 하이브리드 스마트 컨트랙트 SC의 구성 및 실행을 위한 설계 패턴입니다.

하이브리드 스마트 컨트랙트 기본 아이디어에 따라 TEF는 SC를 두 부분으로 분해합니다.

1. TEF 컨텍스트에서 앵커 컨트랙트라 부르는 MAINCHAIN의 `$SC_a$`
2. TEF 실행파일이라고 부르는 DON 로직 `$exec_t$`

여기에서는 SC를 사용하여 `$SC_a$` 및 `$exec_t$`의 조합으로 구현된 논리 컨트랙트를 나타내겠습니다. (앞서 언급한 바와 같이 컨트랙트 SC를 이러한 구성요소로 자동 분해하는 컴파일러 도구를 개발할 예정입니다.)

TEF 실행파일 `$exec_t$`는 SC에서 사용자의 트랜잭션을 처리하는 엔진입니다. 이 엔진은 DON에서 실행되므로 성능에 적합한 방식으로 실행할 수 있으며 다음과 같은 여러 기능을 제공합니다.

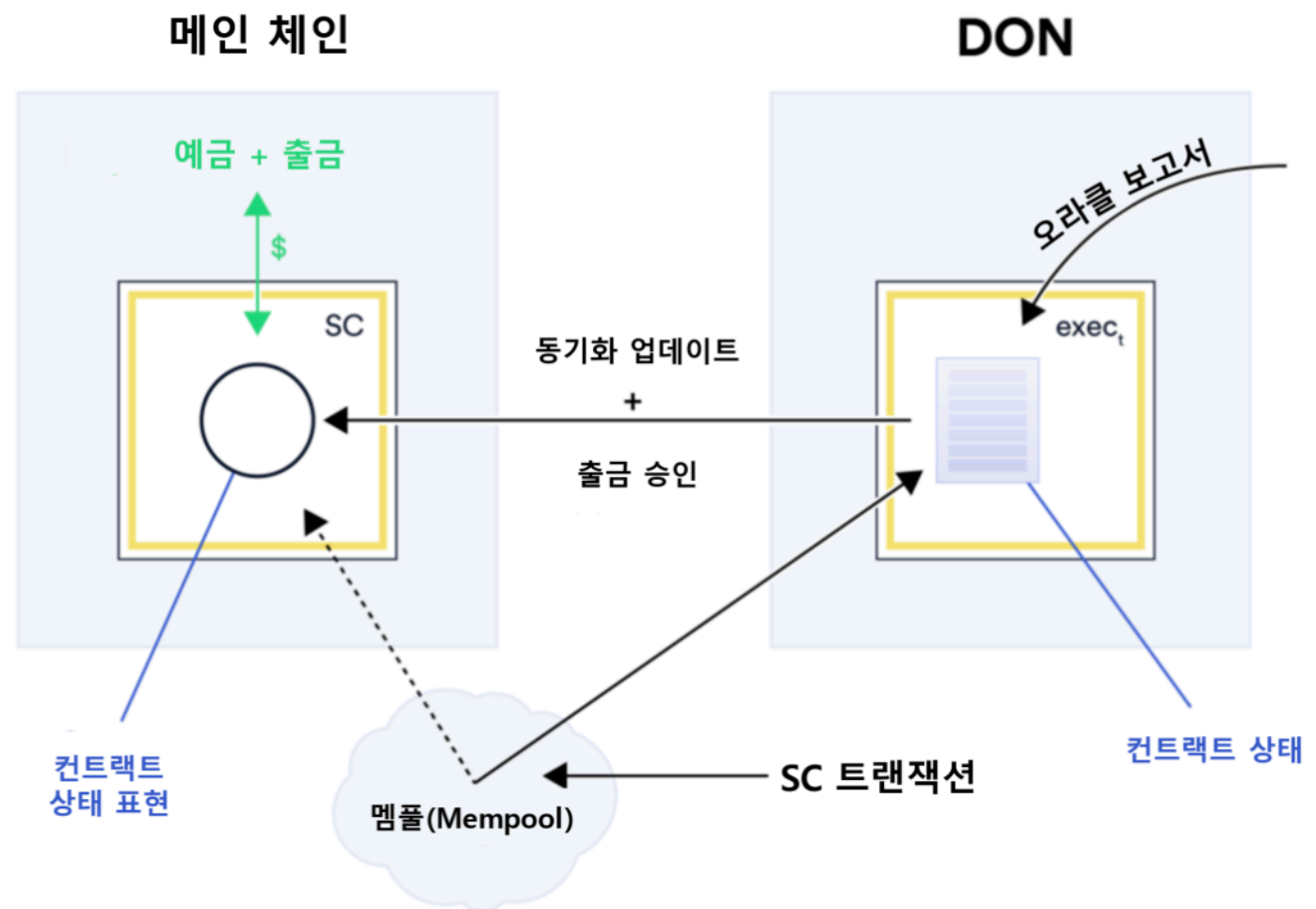
- **트랜잭션 수집:** `$exec_t$`는 사용자의 트랜잭션을 받거나 가져옵니다. 즉, DON에서 트랜잭션 제출이나 MS를 사용하는 MAINCHAIN 메모를 통해 직접 수행할 수 있습니다.
- **빠른 트랜잭션 실행:** `$exec_t$`는 SC 내 자산과 관련된 트랜잭션을 처리합니다. 이는 로컬, 즉 DON에서 수행됩니다.
- **빠르고 저렴한 오라클 / 어댑터 액세스:** `$exec_t$`는 오라클 보고서 및 기타 어댑터 데이터에 대한 네이티브 액세스를 하고 있어 MAINCHAIN 실행보다 더 빠르고 저렴하며 정확한 자산 가격 책정이 가능합니다. 더불어, 오프체인 오라클 액세스는 높은 비용의 온체인 스토리지를 회피하여 오라클 운영 비용 및 시스템 사용 비용을 절감합니다.
- **동기화:** `$exec_t$`는 주기적으로 DON에서 MAINCHAIN으로 업데이트를 푸시하여 `$SC_a$`를 업데이트합니다.

앵커 컨트랙트는 SC의 MAINCHAIN 프론트 엔드입니다. SC의 더 높은 신뢰 구성요소로서 다음과 같은 여러 용도로 사용됩니다.

- **자산 보관:** 사용자 자금은 `$SC_a$`에서 입금, 보관 및 출금됩니다.
- **동기화 확인:** `$SC_a$`는 `$exec_t$`가 동기화할 때, 상태 업데이트의 정확성을 확인할 수 있습니다 (예: 롤업에 연결된 SNARK).
- **가드 레일:** `$SC_a$`는 `$exec_t$`의 손상 또는 실패를 방지하기 위한 조항을 포함할 수 있습니다. (자세한 내용은 섹션 7을 참조합니다.)

TEF에서 사용자 자금은 MAINCHAIN에서 보관되며, 이는 DON 자체가 비수탁형(non-custodial)임을 의미합니다. 동기화 메커니즘의 선택(아래 참조)에 따라 사용자는 정확한 오라클 보고서와 MAINCHAIN과의 시기적절한 동기화를 위해서만 DON을 신뢰해야 할 수 있습니다. 결과적인 신뢰 모델은 오늘날 일반적으로 주문 매칭을 위한 오프체인 구성요소와 청산 및 결제를 위한 온체인 구성요소를 포함하는 주문서 기반 DEX(예: [2])와 매우 유사합니다.

지불 시스템 용어를 사용할 경우에는 $\$SC_a\$$ 가 정산을 담당하는 개념이라면, $\$exec_t\$$ 가 청산을 담당하는 SC 구성요소라고 간주할 수 있습니다. TEF를 도식화하면 그림 13과 같습니다.



• 그림 13: TEF 개념도에서 트랜잭션은 MS를 통해 MAINCHAIN 멤풀을 통해 DON으로 전달됩니다.

TEF의 이점

TEF는 다음과 같은 가지 이점을 제공합니다.

- **높은 성능:** SC는 트랜잭션 및 오라클 보고서 모두에 대해 MAINCHAIN보다 훨씬 높은 DON 처리량을 상속 받습니다. 또한 $\$exec_t\$$ 는 MAINCHAIN 단독 구현보다 더 빠르게 트랜잭션을 처리하고 오라클 보고서에 더 적시에 응답할 수 있습니다.
- **낮은 수수료:** 동기화 프로세스는 트랜잭션 처리보다 시간에 덜 민감하며 트랜잭션은 DON에서 MAINCHAIN으로 일괄 전송될 수 있습니다. 결과적으로 이 접근 방식을 사용하는 트랜잭션당 온체인 수수료(예: 가스 비용)는 MAINCHAIN에서만 실행되는 컨트랙트보다 훨씬 낮습니다.

- **기밀성:** DON 기밀성 보존 메커니즘은 SC에 영향을 미칠 수 있습니다.

TEF의 제약: TEF의 한 가지 제약은 MAINCHAIN에서만 발생하기 때문에 즉각적인 인출을 지원하지 않는다는 것입니다. $\$SC_a\$$ 에 출금 요청을 보낼 때, 사용자는 승인 전 출금 트랜잭션을 포함하는 상태 업데이트를 수행할 때까지 $\$exec_t\$$ 를 대기해야 할 수 있습니다. 섹션 6.2에서는 이와 관련한 부분적 해결 방법 일부에 대해 논의합니다.

TEF의 또 다른 한계는 MAINCHAIN에서 디파이 컨트랙트 원자적 구성, 특히 단일 트랜잭션에서 여러 디파이 컨트랙트를 통해 자산을 라우팅하는 기능을 지원하지 않는다는 것입니다. 그러나 TEF는 동일한 DON에서 실행되는 디파이 컨트랙트 간에 이러한 원자성을 지원할 수 있습니다. 섹션 6.2에서는 이 문제를 해결하는 몇 가지 방법을 추가적으로 논의합니다.

6.2 트랜잭션 라우팅

SC에 대한 트랜잭션은 사용자가 DON으로 직접 보내거나 MAINCHAIN 멤풀을 통해 라우팅할 수 있습니다(FSS를 거쳐). 여기에는 네 가지 고유한 트랜잭션 유형이 존재하며 각기 다른 처리를 해야 합니다.

컨트랙트 내 트랜잭션: TEF는 수수료 역학의 복잡성을 회피할 수 있어 SC에게 레이어-1 컨트랙트에서 사용할 수 있는 것보다 트랜잭션 처리에 더 많은 유연성을 제공할 수 있습니다. 예컨대, ETH 멤풀 트랜잭션은 수수료 가격이 더 높은 새로운 트랜잭션으로 덮어쓸 수 있지만, SC는 SC 내 자산에서 작동하는 트랜잭션을 멤풀에서 보이는 즉시 권위 있는 트랜잭션으로 처리할 수 있습니다. 결과적으로 SC는 블록 내에서 트랜잭션이 확인될 때까지 기다릴 필요가 없으므로 대기 시간이 상당히 줄어들게 됩니다.

프록시: 사용자는 지갑 컨트랙트 또는 MAINCHAIN의 기타 컨트랙트를 통해 SC에 트랜잭션 $\$\tau\$$ 를 보내고자 할 수 있습니다. DON은 MAINCHAIN에서 $\$\tau\$$ 의 실행을 시뮬레이션하여 SC에 대한 후속 트랜잭션이 발생하는지를 결정할 수 있습니다. 따라서 $\$\tau\$$ 는 SC에 대한 다른 트랜잭션과 순서를 지정할 수 있습니다. DON이 이러한 트랜잭션을 식별하는 방법으로는 몇 가지 가능성이 존재합니다. (1) DON은 멤풀의 모든 트랜잭션을 시뮬레이션할 수 있습니다(고비용의 접근 방식). (2) 특정 컨트랙트 또는 컨트랙트 유형(예: 지갑)은 DON 모니터링을 위해 나열될 수 있습니다. 또는 (3) 사용자는 DON 검사를 위해 트랜잭션에 주석을 달 수 있습니다.

공정한 시퀀싱 서비스를 사용하고 주문 정책이 서로 호환되지 않는 두 개 컨트랙트 $\$SC_1\$$ 및 $\$SC_2\$$ 와 단일 트랜잭션이 상호 작용할 때 문제가 더 복잡해집니다. 예를 들어 DON은 두 가지 모두와 호환되는 최근 시간에 $\$\tau\$$ 를 시퀀싱할 수 있습니다.

예금: SC에 MAINCHAIN 자산을 예금하는 트랜잭션은 SC가 유효한 것으로 처리하기 전에 블록에서 확인되어야 합니다. $\$exec_t\$$ 는 자산(예: ETH)을 $\$SC_a\$$ 로 보내는 트랜잭션의 채굴을 감지할 경우, 즉시 입금을 확인할 수 있습니다. 예컨대, DON에 대한 현재 오라클 보고 가격을 자산에 적용할 수 있습니다.

인출: 앞서 언급했듯이 TEF의 한 가지 제약은 인출이 항상 즉시 실행될 수 없다는 것입니다. 롤업형 실행 모델에서 출금 요청은 안전한 처리를 위해 다른 트랜잭션과의 순서가 지정되어야 합니다. 즉, 롤업이 필요합니다. 그러나 이 제약에 대한 일부 부분적인 해결 방안이 존재합니다.

DON이 인출 트랜잭션 τ 까지의 롤업 유효성 증명을 신속하게 연산할 수 있어 멤풀에서 사용자 트랜잭션 τ 에 대한 관찰이 가능해지면, $\text{\$exec_t}$ 는 τ 에 대한 상태 업데이트 트랜잭션 τ' 를 더 높은 수수료 가격으로 보낼 수 있습니다. 이는 일종의 유익한 선행 전략이라 할 수 있습니다. τ' 가 멤풀에 도달하기 전에 τ 가 채굴되지 않을 경우, τ' 는 τ 에 앞서게 되며 τ 는 앞서 승인된 출금에 영향을 미치게 됩니다.

DON이 상태 업데이트를 연산하는 데 의존하는 TEF 변형(아래 임계값 서명 변형 내용 참조)에서 DON은 대안적으로 τ 의 실행 시 SC의 상태를 감안하여 그 승인 여부를 오프체인에서 결정할 수 있습니다. 그런 다음 DON은 전체 상태 업데이트에 영향을 주지 않고 인출 τ 를 승인하는 트랜잭션 τ' 를 보낼 수 있습니다.

이 접근 방식이 불가능하거나 성공하지 못한 경우 DON이 개시한 트랜잭션 τ' 는 τ 에 대한 응답으로 사용자에게 자금을 보낼 수 있으므로 사용자는 추가 트랜잭션을 시작할 필요가 없습니다.

6.3 동기화

TEF 실행파일 $\text{\$exec_t}$ 는 주기적으로 DON에서 MAINCHAIN으로 업데이트를 푸시하여 프로스의 $\text{\$SC_a}$ 상태 업데이트를 하는데, 이를 동기화(syncing)라고 합니다. 동기화는 레이어-2 트랜잭션을 레이어-1로 전파하는 것으로 생각할 수 있으므로 TEF는 이러한 목적을 위해 롤업[5, 12, 16, 69], 낙관적 롤업[10, 11, 141], 발리디움[201] 또는 기본 임계값 서명(예: 임계값 BLS, Schnorr 또는 ECDSA[24, 54, 116, 202])을 포함한 다수의 기존 기법 중 하나를 사용할 수 있습니다. 원칙상 신뢰된 실행 환경은 상태 변경의 정확성을 증명할 수도 있으며, 따라서 롤업에 대한 훨씬 더 좋은 성능의 대안을 제공합니다(단, 하드웨어 종속적인 신뢰 모델을 사용해야 함). (예: [80] 참조)

아래에서는 TEF의 가지 주요 속성과 관련하여 동기화 옵션을 비교합니다.

- **데이터 가용성:** SC 상태가 어디에 저장되는지의 문제입니다. TEF에서는 MAINCHAIN, DON 또는 일부 타사 스토리지 공급자(IPFS 등)라는 가지 옵션을 사용할 수 있습니다. 이들 옵션은 서로 다른 보안 보장, 가용성 수준 및 성능 프로필을 달성합니다. 간단히 말해서, MAINCHAIN에 상태를 저장하면 온체인 감사 기능이 활성화되고 상태 가용성에 대해 당사자에 대한 의존도가 제거됩니다. 반면에 상태를 오프체인에 저장하면 데이터 가용성을 위해 스토리지 공급자(DON 또는 타사)를 신뢰하는 비용으로 스토리지 비용을 줄이고 처리량을 향상시킬 수 있습니다. 물론 이러한 옵션을 결합한 유연한 모델도 가능합니다. 필요시 되는 데이터 가용성의 형식은 표 1과 같습니다.
- **정확성 보장:** $\text{\$SC_a}$ 가 $\text{\$exec_t}$ 가 푸시한 업데이트 정확성을 어떻게 확인하는지 문제입니다. 이는 $\text{\$exec_t}$ 와 $\text{\$SC_a}$ 에서의 연산 부하와 동기화 지연 시간에 영향을 미칩니다(아래 내용 참조).
- **지연 시간:** 동기화 지연에는 가지 요인이 기여합니다. (1) $\text{\$exec_t}$ 가 동기화 트랜잭션 τ_{sync} 를 생성하는 데 걸리는 시간 (2) τ_{sync} 가 MAINCHAIN에서 확인되는 데 걸리는 시간 (3) τ_{sync} 가 $\text{\$SC_a}$ 에 발효되는 시간입니다. TEF에서 지연 시간은 출금에 있어 특히 중요합니다(컨트랙트 내 트랜잭션의 경우 덜 중요). 이는 출금을 위해서는 (최소한 부분적) 상태 동기화가 반드시 필요하기 때문입니다.

동기화 옵션	데이터 가용성	정확성 보장	지연 시간
롤업 [5, 12, 16, 69]	온체인	유효성 증명	생성 소요 시간 유효성 증명(예: 현재 시스템의 회의록)
발리디움 [201]	오프체인	유효성 증명	위와 같음
낙관적 롤업 [10, 11, 141]	온체인	사기 증명	챌린지 기간(예: 일 또는 주)
임계값 서명 [24, 54, 116, 202]	유연	DON의 임계값 서명	동시적
신뢰할 수 있는 실행 환경 [80]	유연	하드웨어 기반 입증	동시적

표 1: TEF 및 해당 속성의 다양한 동기화 옵션

- 표 1에서는 TEF의 5가지 주요 동기화 옵션에 대한 속성을 요약합니다. (여기에서는 이들 기술을 독립형 레이어-2 확장성 솔루션으로 비교하지 않음에 유의. 해당 비교는 [121] 참조.)

아래에서는 각 동기화 옵션에 관해 설명합니다.

롤업: 롤업[69]은 하나의 프로토콜로, 트랜잭션 일괄 처리에 의해 영향을 받는 상태 전환을 오프체인에서 연산하고 상태 변경은 이후 MAINCHAIN으로 전파됩니다.

롤업을 구현하기 위해 앵커 스마트 컨트랙트 SC_a 는 실제 상태에 대한 압축 표현 $R\{state\}$ (예: 머클(Merkle) 루트)을 저장합니다. 동기화를 위해, $exec_t$ 는 $\tau\{sync\} = (T, R'\{state\})$ 를 SC_a 로 전송하는데 여기서 T 는 마지막 동기화 이후에 처리한 트랜잭션 집합이며, $R'\{state\}$ 는 이전 상태 $R\{state_0\}$ 에 T 의 트랜잭션을 적용하여 연산된 새 상태의 압축 표현을 의미합니다.

두 가지 인기 있는 변형이 존재하는데 이들은 SC_a 가 T 집합의 상태 업데이트를 확인하는 방법에 있어 서로 다릅니다. 첫 번째는 ‘tx-롤업’이며 전환 로그, 새로운 $R'\{state\}$ 의 정확성에 대한 간략한 증명(때때로 유효성 증명이라고도 함)을 첨부합니다. 이 변형을 구현한 예시에서 $exec_t$ 는 $\tau\{sync\}$ 와 더불어 유효성 증명(예: 영지식 스나크(zk-SNARKs) 증명)을 연산 후 제출하여, SC_a 가 현재 상태에 T 를 적용한 결과임을 증명합니다.

앵커 컨트랙트는 이 증명을 검증한 후에만 상태 업데이트를 수락하게 됩니다.

낙관적 롤업에는 정확성에 대한 주장이 포함되지 않지만, 상태 전환의 탈중앙화된 검증을 용이하게 하는 예치 및 도전(challenge) 절차가 있습니다. 이 롤업 변형에 있어, SC_a 는 그것이 정확하다고 가정하는 $\tau\{sync\}$ 를 잠정적으로 수락하지만(따라서 낙관적임), $\tau\{sync\}$ 는 MAINCHAIN을 모니터링하는 모든 당사자가 잘못된 상태 업데이트를 식별하고 필요한 조치(예: 상태를 롤백하고 $exec_t$ 에 페

널티를 부과)를 취하도록 \$SC_a\$에게 알릴 수 있는 도전 기간이 지난 후에야 적용됩니다.

두 롤업 변형 모두 전체 상태를 구성할 수 있는 온체인에 트랜잭션을 게시하게 되므로 온체인 데이터 가용성을 달성합니다. zk-롤업의 대기 시간은 유효성 증명을 생성하는 데 필요한 시간에 따라 다른데, 기존 시스템의 경우 일반적으로 몇 분 정도가 되며[16] 시간이 지남에 따라 개선될 가능성이 있습니다. 반면에 낙관적 롤업은 챌린지 기간이 사기에 대한 증거가 작동하기에 충분히 길어야 하므로, 대기 시간(예: 며칠 또는 몇 주)이 더 길어집니다. 확인이 느리다는 것의 의미는 모호하고 때로는 체계에 따라 다르게 볼 수 때문에 철저한 분석은 어렵습니다. 예컨대, 일반 사용자가 MAINCHAIN보다 훨씬 빠르게 롤업을 확인할 수 있기 때문에 특정 체계는 상태 업데이트가 확인되기 전에 지불 트랜잭션을 ‘신뢰 없는 종결(trustless final)’[109]로 간주합니다.

발리디움: 발리디움은 데이터를 오프체인에서만 사용할 수 있도록 하고 MAINCHAIN의 모든 데이터를 유지 관리하지 않는 (zk-)롤업의 한 형태입니다. 특히, \$exec_t\$는 \$SC_a\$에 새로운 상태와 증명만을 보내며 트랜잭션은 보내지 않습니다. 발리디움 방식의 동기화를 사용할 경우, \$exec_t\$와 이를 실행하는 DON은 완전한 상태를 저장하고 트랜잭션을 실행하는 유일한 당사자가 됩니다. zk-롤업과 마찬가지로 동기화 대기 시간은 유효성 증명 생성 시간에 따라 다릅니다. 그러나 zk-롤업과 달리 발리디움 형식의 동기화는 스토리지 비용을 줄이고 처리량을 늘립니다.

DON의 임계값 서명: DON 노드 임계값이 정직하다고 가정할 경우, 가장 간단하고 빠른 동기화 옵션은 DON 노드가 새 상태에 집합적으로 서명하도록 하는 것입니다. 이 접근 방식은 온체인 및 오프체인 데이터 가용성을 모두 지원할 수 있습니다. 사용자가 오라클 업데이트에 대해 DON을 신뢰한다면, 사용자는 이미 임계값 신뢰 모델에 있으므로 상태 업데이트를 수락하기 위해 DON을 더 신뢰할 필요가 없습니다. 임계값 서명의 또 다른 이점은 짧은 대기 시간입니다. EIP-2938 [70]에서 제안된 계정 추상화(account abstraction)라는 새로운 트랜잭션 서명 형식에 대한 지원은 임계값 서명의 구현을 상당히 쉽게 할 것입니다. 임계값 Schnorr [202] 또는 BLS [55] 서명과 같은 대안보다 훨씬 더 복잡한 프로토콜(예: [116, 117, 118])을 포함하는 임계값 ECDSA의 필요성을 제거하기 때문입니다.

신뢰할 수 있는 실행 환경(TEE): 42TEE는 내부에서 실행되는 프로그램에 대한 강력한 보안 제공을 목표로 하는 격리된 실행 환경(일반적으로 하드웨어로 구현됨)입니다. 일부 TEE(예: 인텔 SGX [84])는 출력이 특정 입력에 대한 특정 프로그램에 의해 올바르게 연산되었다는 증명(attestations)을 생성할 수 있습니다.

동기화의 TEE 기반 변형은 [80]의 기술을 사용하여 (zk-)롤업 또는 발리디움의 증명을 TEE 증명으로 대체하여 구현할 수 있습니다. 롤업 및 발리디움에 사용되는 영지식 증명과 비교할 때 TEE는 훨씬 더 뛰어난 성능을 제공합니다. TEE는 임계값 서명과 비교할 때 원칙적으로 하나의 TEE만 관련되어야 하므로, 임계값 ECDSA 서명을 생성해야 하는 복잡성을 해소합니다. 그러나 TEE를 사용하면 하드웨어 종속적 신뢰의 가정이 추가로 도입됩니다. TEE를 임계값 서명과 결합하여 일부 TEE 인스턴스의 손상에 대한 복원력을 생성할 수도 있지만 이러한 보호 조치는 임계값 ECDSA 서명을 생성하는 복잡성을 다시 발생시키게 될 것입니다.

추가 유연성: 이러한 동기화 옵션은 다음과 같은 방식으로 더 유연하게 조정할 수 있습니다.

- **유연한 트리거링:** TEF 애플리케이션은 동기화가 트리거되는 조건을 결정할 수 있습니다. 예를 들어 동기화는 배치 기반일 수 있습니다. 예를 들면, 모든 \$N\$ 트랜잭션마다 발생하거나, 시간 기반(예: 10블록마다)으로 또는 이벤트 기반(예: 대상 자산 가격의 큰 변동 시마다) 발생할 수 있습니다.

- **부분 동기화:** \$exec_t\$가 소량의 상태를 빠르게 동기화하여 전체 동기화를 주기적으로만 수행하도록 할 수 있으며, 일부 경우에는 바람직한 것으로 간주됩니다(예: 롤업의 경우 부분 동기화는 대기 시간을 줄일 수 있음). 예컨대, \$exec_t\$는 MAINCHAIN 상태를 달리 업데이트하지 않고 사용자의 \$SC_a\$ 잔액을 업데이트하여 출금 요청을 승인할 수 있습니다.

6.4 재편성

네트워크 불안정성 또는 51% 공격으로 인한 블록체인의 재편성은 메인 체인의 무결성에 위협이 될 수 있습니다. 실제로 공격자들이 이를 사용하여 이중 지불 공격을 시작한 바 있습니다[34]. 주요 체인에 대한 이러한 공격은 실행하기 어렵지만, 일부 체인에서는 여전히 가능합니다[88].

DON은 MAINCHAIN과 독립적으로 작동하기 때문에 공격과 관련된 재편성(reorg)을 관찰하고 보호할 흥미로운 가능성을 제공합니다.

예를 들어, DON은 일부 임계값 길이 τ 의 경쟁 포크에 관한 존재 여부를 MAINCHAIN 의존 컨트랙트 SC에 보고할 수 있습니다. DON은 작업증명(PoW) 또는 지분증명(PoS) 설정에서 그러한 포크의 존재에 대한 증거를 추가로 제공할 수 있습니다. 컨트랙트 SC는 일정 기간 추가 트랜잭션 실행을 중단하는 것과 같은 적절한 방어 조치를 구현할 수 있습니다(예: 거래소에서 이중 사용 자산을 블랙리스트에 올리도록 허용). 51%의 공격을 가하는 공격자가 DON의 보고서를 검열하려 할 수도 있지만 SC의 대응책은 트랜잭션(즉, 하트비트)을 처리하기 위해 DON으로부터 정기적인 보고를 요구하거나 높은 가치의 트랜잭션을 검증하기 위해 새로운 보고서를 요구합니다.

이러한 포크 경보는 원칙적으로 DON이 다양한 목적을 위해 제공할 수 있는 일반 서비스에 해당하지만 TEF와 통합될 계획입니다.

7. 신뢰 최소화

이질적인 개체 집합의 참여를 통한 탈중앙 시스템인 체인링크 네트워크는 활성화(가용성)와 안전성(보고서의 무결성) 차원 모두에 있어 오류에 대한 강력한 보호 기능을 제공합니다. 그러나 대부분의 탈중앙 시스템은 구성요소 자체의 탈중앙화 수준이 다르게 구성됩니다. 이는 채굴자[32]와 중개자[51] 사이에 탈중앙의 제한이 오랫동안 존재해 온 대규모 시스템에서도 마찬가지입니다.

모든 탈중앙화 노력의 목표는 신뢰를 최소화하는 것입니다. 우리는 악의적인 DON으로 인한 경우에도 체인링크 네트워크 내에서 시스템 손상 또는 실패의 부작용을 줄이기 위해 노력합니다. 우리의 주된 원칙은 최소 권한의 원칙(Principle of Least Privilege)[197]입니다. 시스템 내의 시스템 구성요소와 행위자는 할당된 역할을 성공적으로 완료할 수 있도록 엄격하게 지정된 범위의 권한을 가져야 합니다.

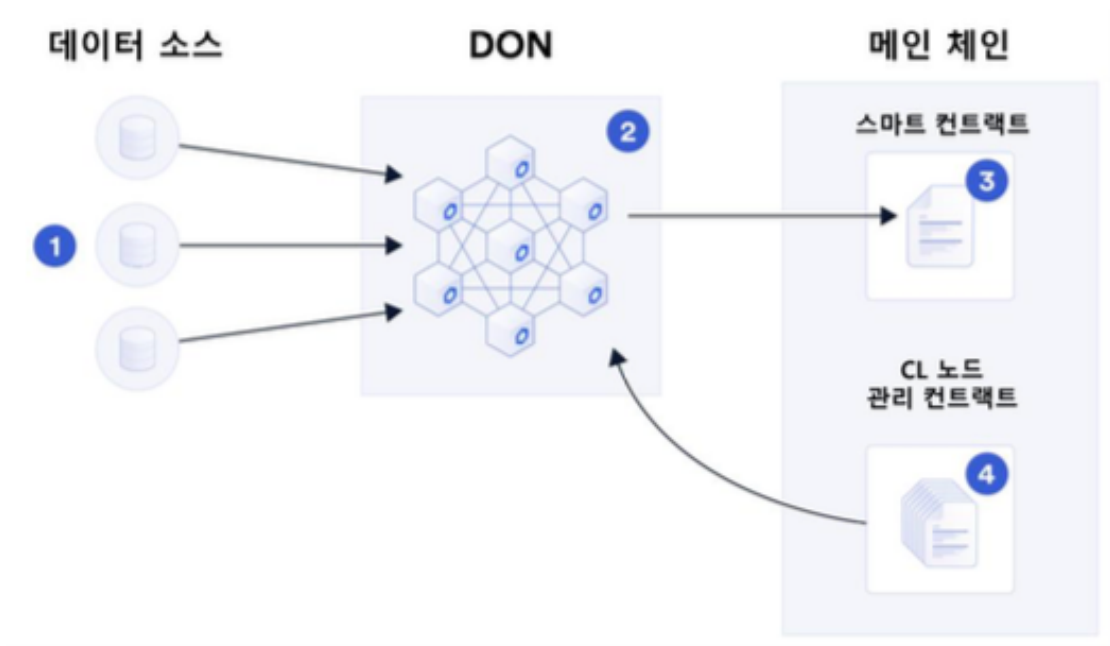
여기에서 우리는 체인링크가 점점 더 큰 신뢰 최소화를 위해 채택할 몇 가지 구체적인 메커니즘을 제시합니다. 우리는 이러한 메커니즘의 특성을 그림 14와 같이 그 근원이 존재하는 위치, 즉 시스템 구성요소의 관점에서 파악합니다. 아래 각 하위 섹션에서는 이러한 각각의 위치에 관해 설명합니다.

7.1 데이터 소스 인증

오라클에 대한 현재 운영 모델은 한 가지 사실에 의해 제약을 받습니다. 이는 TLS가 기본적으로 데이터에 서명하지 않기 때문에 생략된 데이터에 디지털 서명하는 데이터 소스가 거의 없다는 것입니다. TLS는 자체 ‘핸드셰이크(handshake)’ 프로토콜에서 (서버와 클라이언트 간에 공유키를 설정하기 위해) 디지털 서명을 사용합니다. 따라서 HTTPS 지원 서버에는 원칙적으로 데이터 서명에 사용할 수 있는 공개키에 대한 인증서가 존재하지만, 일반적으로는 이러한 인증서를 사용하여 데이터 서명을 지원하지는 않습니다. 이에 따라 DON의 보안은 현재 오라클 네트워크처럼, 데이터 소스에서 컨트랙트로 데이터를 충실하게 중계하는 오라클 노드에 의존하게 됩니다.

체인링크 신뢰 최소화에 대한 우리 비전에 있어 장기적으로 중요한 한 가지는 데이터 서명을 위한 도구 및 표준 지원을 통한 더 강력한 데이터 소스 인증을 포괄합니다. 데이터 서명은 종단 간(end-to-end) 무결성 보장을 시행하는 데 도움이 될 수 있습니다. 원칙상 어떤 컨트랙트가 데이터 소스가 직접 서명한 데이터 D의 조각을 입력으로 수락할 경우, 오라클 네트워크는 D를 실제로 변조할 수 없습니다.

이러한 데이터 서명을 가능하게 하는 다양한 고무적인 활동이 이루어지고 있습니다. 대표적으로는 주로 사용자 인증을 위해 설계된 OpenID Connect[9], TLS 인증서를 용도 변경하여 TLS[191]를 확장하는 것을 목표로 하는 학술 프로젝트 TLS-N, TLS Evidence Extensions[63] 등이 있습니다. OpenID Connect는 일부 채택되었으나, TLS Evidence Extensions 및 TLS-N은 아직 채택된 바가 없습니다.



- 그림 14: 이 선에서 논의된 신뢰 최소화 메커니즘의 위치. ① 데이터 소스는 ② DON에 데이터를 제공하고, DON은 데이터의 함수를 종속된 ③ 스마트 컨트랙트에 중계합니다. 또한, DON 또는 오라클 네트워크에는 보상 노드, 가드레일 등을 위한 ④ MAINCHAIN 상의 노드 관리 스마트 컨트랙트가 포

함됩니다.

데이터 소스 인증의 또 다른 잠재적인 방법으로는 게시자 고유의 SXG(Signed HTTP Exchanges) [230]를 사용하는 방법이 있습니다. 이는 가속 모바일 페이지(Accelerated Mobile Pages, AMP) 프로토콜[225]의 일부로 콘텐츠 전송 네트워크에 캐시를 처리할 수 있게 해줍니다. 크롬 모바일 브라우저는 AMP 캐시 SXG의 콘텐츠를 캐시 서버 도메인이 아닌 게시자의 자체 네트워크 도메인에서 제공한 것으로 표시합니다. 이러한 클라우드플레어(CloudFlare) Real URL[83] 및 구글 amppackager[124]와 같은 브랜드 서비스를 사용하면 비교적 쉽게, 캐시된 뉴스 콘텐츠에 SXG를 광범위하게 채택할 수 있습니다. 즉, 체인링크 오라클이 유효한 SXG에서 보고된 주목할 만한 이벤트를 트리거할 수 있는 간단하면서도 변조 방식이 가능한 방법이라 할 수 있겠습니다. 뉴스 게시자의 AMP 캐시 SXG는 거래 데이터에 대한 보고서와 같은 고속 애플리케이션에는 유용하지 않지만, 극단적인 날씨 변화 또는 선거 결과와 같은 실제 이벤트와 관련된 맞춤형 컨트랙트에 안전한 소스를 제공할 수 있습니다.

우리는 데이터 소스 서명을 가속화하는 데 간단한 배포, 성숙한 도구 및 유연성이 필수적이라고 믿습니다. 데이터 제공자가 체인링크 노드를 인증된 API 프런트 엔드로 사용할 수 있도록 하는 것은 유망한 접근 방식으로 보입니다. 우리는 완전한 오라클로 네트워크에 참여하거나 참여하지 않을 때도 모든 노드가 이 모드에서 작동할 수 있는 옵션을 제공하고자 합니다. 이러한 기능은 데이터 출처 인증(authenticated data origination, ADO)이라고 합니다. ADO와 함께 체인링크 노드를 사용하면, 데이터 소스는 기존의 오픈체인 API 제품군에 디지털 서명 기능을 추가할 때 체인링크 커뮤니티에서 개발한 경험과 도구의 이점을 누릴 수 있습니다. 노드를 오라클로 운영하기로 선택할 경우, 기존 데이터 제공자(예: 크라켄(Kraken)[28], 카이코(Kaiko)[140]) 및 체인에서 API 데이터를 판매하기 위해 체인링크 노드를 실행하는 기타 주체처럼 동일한 모델로 새로운 잠재 수익 흐름을 창출할 수 있습니다.

7.1.1 데이터 출처 인증(ADO)의 한계

데이터 소스에 의한 디지털 서명은 인증을 강화하는 데 도움이 될 수 있지만, 그 자체로는 오라클 네트워크의 모든 자연적 보안 또는 운영 목표를 달성하기에 충분하지 않습니다.

우선, 주어진 데이터 $\$D\$$ 조각은 여전히 데이터 소스에서 스마트 컨트랙트 또는 다른 데이터 소비자로 강력하고 시기적절한 방식으로 중계되어야 합니다. 즉, 종속 컨트랙트에 미리 프로그래밍된 키를 사용하여 모든 데이터에 서명하는 이상적인 환경에서도 소스에서 컨트랙트로 안정적으로 데이터를 전달하기 위해서는 DON이 여전히 필요합니다.

또한 컨트랙트 또는 기타 오라클 데이터 소비자가 소스 데이터에 대해 연산된 다양한 기능의 인증된 출력값에 대한 액세스를 원하는 경우가 많습니다. 그 주된 이유는 두 가지로 다음과 같습니다.

- **기밀성:** 데이터 소스 API는 체인에서 공개적으로 표시되기 전에 수정이나 삭제해야 하는 민감한 독점 데이터를 제공할 수 있습니다. 그러나 서명된 데이터를 수정하면 서명은 무효가 됩니다. 즉, 단편적인 ADO와 데이터 위생(data sanitization) 전략은 양립할 수 없습니다. 예시 3에서는 향상된 ADO를 통해 이 둘을 조정하는 방법을 보여줍니다.
- **데이터 소스 오류:** 오류와 실패는 모두 데이터 소스에 영향을 줄 수 있으며 디지털 서명은 이 문제 중 어떤 것도 해결할 수 없습니다. 체인링크는 처음부터 [98] 이러한 결함을 수정하는 메커니즘인 이중화(redundancy) 전략을 이미 포함시킨 바 있습니다. 오라클 네트워크에서 발행한 보고서는 일반적으

로 여러 소스의 결합된 데이터를 나타냅니다.

아래에서는 원본 데이터의 기밀성을 강화하고, 여러 원본의 데이터를 안전하게 결합하기 위해 ADO 설정에서 탐색 중인 체계에 대해 논의하겠습니다.

7.1.2 기밀성

데이터 소스는 사용자가 원하는 API의 전체 영역을 예상하거나 제공하지 못할 수 있습니다. 특히 사용자는 기밀성을 보장하기 위해 사전 처리된 데이터에 액세스하기를 원할 수 있습니다. 다음의 예시에서는 이 문제를 다룹니다.

예시 3. Alice는 자신이 18 이상(예를 들어 대출받을 조건)임을 나타내는 탈중앙 ID(DID) 자격 증명을 얻기를 원합니다. 이를 위해 Alice는 자신의 나이에 관한 이 사실을 DID 자격 증명 발급자에게 증명해야 합니다.

Alice는 이를 위해 그녀가 거주하는 지역의 자동차국(Department of Motor Vehicles, DMV) 웹사이트의 데이터를 사용하기를 희망합니다. DMV는 Alice의 생년월일 기록을 갖고 있으며, 다음과 같은 형식으로 디지털 서명된 증명 $\$A\$$ 를 발행합니다.

$\$A\$ = \{\text{\texttt{Name: Alice}}, \text{\texttt{DoB: 02/16/1999}}\}.\$$

위의 예시에서 증명 $\$A\$$ 는 Alice가 18 이상임을 DID 자격 발급자에게 증명하기에 충분할 수 있습니다. 그러나 이는 Alice의 정확한 DoB, 즉 불필요하게 민감한 정보를 유출하게 됩니다. 사실 Alice가 DMV에 바라는 것은 ‘A씨는 18 이상이다’라는 간단한 진술에 서명하는 것뿐입니다. 즉, Alice는 자신의 생년월일 $\$X\$$ 에 대한 함수 $\$G\$$ 의 출력을 바랍니다. 여기서 (비공식적이지만) $\$\text{\texttt{CurrentDate}} - X \geq 18\text{\texttt{년}}\$$ 의 경우에는 $\$A' = G(X) = \text{\texttt{True}}\$$ 가 되며, 그렇지 않을 때는 $\$G(X) = \text{\texttt{False}}\$$ 가 됩니다.

일반화를 위해 Alice는 데이터 소스에서 다음과 같은 형식으로 서명된 증명 $\$A'\$$ 를 요청할 수 있기를 바랄 것입니다.

$\$A' = \{\text{\texttt{Name: Alice}}, \text{\texttt{Func: }} G(X), \text{\texttt{Result: True}}\}.\$$

여기서, $\$G(X)\$$ 는 함수 $\$G\$$ 와 그 입력 $\$X\$$ 의 사양을 나타냅니다. 우리는 사용자가 해당 증명 $\$A'\$$ 에 대한 요청과 함께 원하는 $\$G(X)\$$ 를 입력으로 제공할 수 있어야 한다고 생각합니다.

데이터 소스 증명 $\$A'\$$ 는 $\$A'\$$ 의 올바른 해석을 위해 사양 $\$G(X)\$$ 를 포함해야 합니다. 위의 예시에서 $\$G(X)\$$ 는 $\$A'\$$ 에서의 불리언 값 의미를 정의하므로 True는 해당 증명의 대상이 18 이상임을 나타냅니다.

우리는 사용자가 $\$G(X)\$$ 를 기능 쿼리로 지정할 수 있게 하는 유연한 쿼리를 참조합니다. 우리는 예시 3과 같은 사용 사례와 컨트랙트에서 직접 쿼리를 포함하는 사용 사례를 지원하고자 ADO의 일환으로 간단한 함수 $\$G\$$ 를 포함하는 기능 쿼리를 지원하고자 합니다.

7.1.3 소스 데이터 결합

온체인 비용을 줄이기 위해 컨트랙트는 일반적으로 다음 예와 같이 여러 소스에서 결합된 데이터를 사용하도록 설계됩니다.

예시 4(가격 데이터의 중간화). 가격 피드, 즉 어떤 자산(예: USD)에 대한 특정 자산(예: ETH)의 상대적 가치를 제공하기 위해 오라클 네트워크는 일반적으로 거래소와 같은 여러 출처에서 현재 가격을 얻습니다. 오라클 네트워크는 일반적으로 이러한 값의 중앙값을 종속 컨트랙트 $\$SC\$$ 로 보냅니다.

데이터 서명이 있는 환경에서 올바르게 작동하는 오라클 네트워크는 데이터 소스 $\$S=\{S_1, \dots, S_{n_S}\}$ 로부터 n_S 소스의 일련의 $V=\{v_1, v_2, \dots, v_{n_S}\}$ 값을 수반된 소스별 서명 $\Sigma=\{\sigma_1, \sigma_2, \dots, \sigma_{n_S}\}$ 와 함께 얻습니다. 이 서명을 확인하면 네트워크는 해당 가격 $v=\mathrm{median}(V)$ 를 $\$SC\$$ 에 전송합니다.

안타깝게도, 오라클 네트워크가 서명된 입력에 대해 v 가 올바르게 계산되었다는 간결한 증거 Σ^* 와 함께 예시 4의 중앙값 v 를 $\$SC\$$ 로 전송할 수 있는 간단한 방법은 없습니다.

한 가지 단편적인 접근 방식은 모든 n_S 데이터 소스의 공개키를 $\$SC\$$ 로 인코딩하는 것입니다. 그런 다음 오라클 네트워크는 (V, Σ) 를 중계하고 $\$SC\$$ 가 V 의 중앙값을 연산하도록 허용합니다. 그러나 이 방법은 크기가 $O(n_S)$ 인 증거 Σ 가 될 것입니다. 즉, Σ^* 는 간결하지 않게 될 것입니다. 또한 Σ 의 모든 서명을 확인해야 하는 $\$SC\$$ 에 높은 수수료 비용이 발생하게 됩니다.

이와 대조적으로 SNARK를 사용하면 올바르게 결합된 인증 소스값에 대한 간결한 증명이 가능합니다. 이는 실제로 실행 가능하지만, 증명자에게 상당히 높은 연산 비용을 부과하고 체인에 다소 높은 수수료 비용을 부과합니다. 타운크라이어를 사용할 수도 있지만, 이 경우 모든 사용자의 신뢰 모델에 맞지 않는 TEE를 사용해야 합니다.

기능 서명(functional signatures)이라 알려진 암호화 도구는 소스로부터 결합된 데이터에 서명하는 일반적인 문제를 해결하는 실마리를 제공하는 한 가지 유용한 개념입니다[59, 132]. 간단히 말하면, 서명자는 기능 서명을 통해 서명 기능을 위임할 수 있으며, 따라서 대리인은 서명자가 선택한 기능 F 범위의 메시지에만 서명할 수 있습니다. 부록 D에서는 이러한 기능적 제약이 데이터 소스에 의해 서명된 값의 함수로 DON이 내보낸 보고서 값의 범위를 어떻게 제한할 수 있는지 설명합니다. 또한 이산화 기능 서명(discretized functional signature)라는 새로운 프리미티브를 도입합니다. 이는 정확도에 대한 완화된 요구사항을 포함하지만, 잠재적으로 SNARK와 같은 접근 방식보다 훨씬 더 나은 성능을 제공합니다.

출력의 소스 인증을 포함하는 방식으로 데이터 소스를 결합하는 문제는 어떤 경우에는 공개하고 어떤 경우에는 독점적인 방법론을 사용하여 볼륨을 기준으로 가중치를 부여하는 다양한 거래소로부터 데이터를 얻는 데이터 수집기(예: CoinCap, CoinMarketCap, CoinGecko, CryptoCompare 등)에도 적용됩니다. 소스 인증으로 값을 게시하려는 집계자는 소스 데이터를 집계하는 노드 컬렉션과 동일한 문제에 직면하게 됩니다.

7.1.4 소스 데이터 처리

정교한 스마트 컨트랙트는 많은 자산에 대한 최근 가격 기록의 변동성 또는 관련 이벤트에 대한 뉴스의 텍스트 및 사진과 같은 기본 데이터 소스의 사용자 지정(custom) 집계 통계에 의존할 가능성이 큼니다. DON에서는 연산과 대역폭이 상대적으로 저렴하므로, 블록체인을 대상으로 하는 출력값이 아주 간결하기만 하면 이러한 통계를(입력이 많은 복잡한 머신러닝 모델일지라도) 경제적으로 처리할 수 있습니다.

DON 참가자가 복잡한 입력에 대해 서로 다른 견해를 가질 수 있는 연산 집약적 작업의 경우, 결과를 연산하기 전에 입력에 대한 컨센서스를 설정하기 위해 DON 참가자 간의 추가 의사소통이 필요할 수 있습니다. 최종값이 입력으로 완전히 결정되는 한, 입력 컨센서스가 설정되면 각 참가자는 단순히 값을 연산하고 부분 서명을 사용하여 다른 참가자에게 브로드캐스트하거나 집계자에게 보낼 수 있습니다.

7.2 DON 신뢰 최소화

우리는 DON 구성요소에 대한 신뢰를 최소화하는 두 가지 주요 방법인 페일오버 클라이언트(failover clients)와 마이너리티 리포트(minority reports)를 구성합니다.

7.2.1 페일오버(failover) 클라이언트

암호화 및 탈중앙 시스템 관련 문헌상 제시된 적대적 모델은 일반적으로 노드의 하위 집합(많은 BFT 프로토콜의 경우 1/3 미만의 노드)을 손상시킬 수 있는 적대자로 인식합니다. 그러나 모든 노드가 동일한 소프트웨어를 실행할 경우, 치명적 악용을 식별할 수 있는 공격자는 일반적으로 이론상 모든 노드를 어느 정도 동시에 손상시킬 수 있다고 여겨집니다. 이러한 설정을 ‘소프트웨어 모노컬처(software monoculture)’라고 표현합니다[47].

이러한 문제를 해결하기 위해 소프트웨어 및 소프트웨어 구성을 자동으로 다양화하기 위한 다양한 제안이 이루어진 바 있습니다(예: [47, 113]). 그러나 [47]에서 지적한 바와 같이, 소프트웨어 다양성은 복잡한 문제이며 신중한 고려가 필요합니다. 예컨대, 소프트웨어 다양화는 시스템의 공격 표면을 증가시키는데, 시스템이 제공하는 보안상의 편익보다 공격 벡터 수가 크다면 보안은 모노컬처보다 더 악화될 수 있습니다.

강력한 페일오버(failover) 클라이언트(즉, 치명적 이벤트 시 노드를 전환할 수 있는 클라이언트)에 대한 지원은 특히 매력적인 소프트웨어 다양화 전략으로 판단됩니다. 페일오버 클라이언트는 메인 소프트웨어로 배포되지 않기 때문에 잠재적인 공격 벡터의 수를 늘리지 않습니다. 또한 2차 방어선으로서의 분명한 이점을 제공합니다. 우리는 단일 클라이언트에 대한 보안 의존도를 줄이는 핵심 수단으로 DON에서 페일오버 클라이언트를 지원하고자 합니다.

체인링크는 이미 강력한 페일오버 클라이언트 시스템을 갖추고 있습니다. 이 접근 방식에는 실전 테스트를 완료한 이전 클라이언트 버전의 유지 관리까지 포함됩니다. 예를 들어 오늘날 OCR(Off-Chain Reporting)을 기본 클라이언트로 사용하는 체인링크 노드에는 필요한 경우 체인링크의 이전 FluxMonitor 시스템에 대한 지원이 포함됩니다. FluxMonitor는 실제로 도입되어 얼마간 사용되었으며 보안 감사 및 현장 테스트를 완료했습니다. 이는 OCR과 동일한 기능을 제공하며, 더 많은 비용이 들지만 사용한만큼 비용이 발생합니다.

7.2.2 마이너리티 리포트

충분히 큰 소수 집합(다수의 부정행위를 관찰하는 정직한 노드의 일부)이 주어진다면 마이너리티 보고서를 생성하는 데 도움이 될 수 있습니다. 이 보고서는 병렬 보고서 또는 플래그이며 $SO_{\{minority\}}$ 에 의해 온체인 종속 컨트랙트 SC 에 전달됩니다. SC 는 자체적인 컨트랙트별 정책에 따라 이 플래그를 사용할 수 있습니다. 예컨대, 안전이 활성화나 반응성보다 더 중시되는 컨트랙트에서는 마이너리티 리포트는 다른 DON에 추가 보고서를 요청하는 컨트랙트나 서킷 브레이커를 유발할 수 있습니다(다음 섹션 참조).

마이너리티 리포트는 대다수가 정직한 때도 중요한 역할을 할 수 있습니다. 모든 보고서 집계 체계는 기능 서명을 사용하더라도 오라클 또는 데이터 오류에 대한 복원력을 보장하기 위해 임계값 방식으로 작동해야 하기 때문입니다. 다시 말해, 일부 임계값 k_S 의 경우, $k_S < n_S$ 오라클의 입력을 토대로 하나의 유효한 보고서를 생성할 수 있어야 합니다. 이는 손상된 DON의 경우, 모든 출처가 정직하더라도 전체 오라클트에 의해 V 에서 보고된 n_S 중에서 선호하는 k_S 값을 선택하여 보고된 값을 조작하는 데 어느 정도의 자유도가 존재한다는 사실을 의미합니다.

예컨대, 기능 서명을 사용하여 ETH의 USD 가격에 대한 V 의 중간값 연산을 인증하는 시스템에서 $n_S=10$ 이고 $k_S=7$ 이라고 가정해 보겠습니다. 5개의 소스에서 $\$500$ 의 가격을 보고했지만, 다른 5개의 출처에서는 $\$1000$ 를 보고한다고 가정하겠습니다. 그런 다음 가장 낮은 7개의 보고서의 중간값을 취하면, DON이 유효한 값 $v=\$500$ 를 출력할 수 있고, 가장 높은 보고서를 중간값으로 취하면 $v=\$1000$ 를 출력할 수 있습니다.

모든 노드가 어떤 데이터가 사용 가능한지, 어떤 데이터가 보고서 구성에 사용되었는지 알 수 있도록 DON 프로토콜을 향상하는 방법으로, 노드는 통계적으로 유의한 경향성을 탐지하고 플래그를 지정하여 하나의 보고서트를 다른 것보다 선호하게 할 수 있으며, 그 결과로 마이너리티 리포트를 생성할 수 있습니다.

7.3 가드 레일

DON에 대한 우리의 신뢰 모델은 MAINCHAIN을 DON보다 더 높은 보안, 더 높은 권한 시스템으로 취급합니다. (이 신뢰 모델이 항상 옳은 것은 아니지만 설사 DON이 더 높은 보안 플랫폼이더라도 결과 메커니즘을 그에 맞게 적용시키는 것이 그 반대의 경우보다 더 쉽습니다.)

따라서 자연스러운 신뢰 최소화 전략에는 DON에 대한 MAINCHAIN 프론트 엔드 또는 종속 컨트랙트 SC 에서 직접 스마트 컨트랙트의 모니터링 및 안전장치 메커니즘 구현 등이 포함됩니다. 우리는 이러한 메커니즘을 가드 레일(guard rails)이라고 부르는데, 가장 중요한 몇 가지를 들자면 다음과 같습니다.

- **서킷 브레이커 (Circuit breakers):** SC 는 상태 업데이트 자체의 특성(예: 순차적 보고서 간의 큰 편차) 또는 다른 입력을 기반으로 한 기능으로 상태 업데이트를 일시 중지하거나 중단할 수 있습니다. 예컨대, 오라클 보고서가 시간이 지남에 따라 비합리적으로 다양화될 경우 서킷 브레이커가 가동될 수 있습니다. 서킷 브레이커는 마이너리티 리포트에 의해서도 가동될 수도 있습니다. 이로써 DON이 크게 잘못된 보고를 하지 않도록 방지할 수 있습니다.
서킷 브레이커는 추가 개입을 고려하거나 실행할 시간을 제공할 수 있습니다. 탈출 해치(escape hatch)는 그러한 개입에 해당합니다.
- **탈출 해치 (escape hatch):** 일련의 관리인, 커뮤니티 토큰 보유자 또는 기타 수탁자 단체에 의해 확인된 불리한 상황에서 컨트랙트는 때때로 이른바 탈출 해치(escape hatch)라는 비상 기능을 호출할 수 있습니다[163]. 탈출 해치는 SC 가 어떤 방식으로든 종료되도록 하거나 보류 중인 트랜잭션이

나 미래의 트랜잭션을 종료합니다. 예컨대, 탈출 해치는 보관된 자금을 사용자에게 반환시키고[17], 컨트랙트 조건을 종료할 수 있으며[162], 보류 중인 트랜잭션이나 미래의 트랜잭션을 취소할 수 있습니다[173]. 탈출 해치는 DON에 의존하는 컨트랙트뿐만 아니라 모든 유형의 컨트랙트에 배포될 수 있어 DON 불법 행위에 대한 잠재적 완충 장치로서 많은 관심을 받고 있습니다.

- **페일오버 (failover):** \$SC\$가 필수 서비스를 위해 DON에 의존하는 시스템에서, \$SC\$는 DON 실패 또는 오작동의 경우에도 서비스 지속을 보장하는 페일오버 메커니즘을 제공할 수 있습니다. 예를 들어 TEF(선 6)에서 앵커 컨트랙트 \$SC_a\$는 특정 중요 작업(예: 인출)이나 일반 트랜잭션에 대해 온체인 및 오프체인 실행 인터페이스가 모두 지원되는 이중 인터페이스를 적절한 지연으로 제공할 수 있으며 이를 통해 DON 트랜잭션의 선행을 방지합니다. 데이터 소스가 데이터에 서명할 경우, 사용자는 DON이 작업에 실패하더라도 \$SC_a\$에 보고서를 제공할 수도 있습니다.

다양한 형태의 낙관적 롤업(선 6.3 참조)에 대해 제안된 사기 증명은 이와 유사하며 위에서 열거한 메커니즘을 보완합니다. 또한 사기 증명은 오프체인 시스템 구성요소의 잠재적인 오류에 대한 일종의 온체인 모니터링과 보호 기능도 제공합니다.

7.4 최소 신뢰의 거버넌스

모든 탈중앙 시스템과 마찬가지로 체인링크 네트워크도 시간이 지남에 따라 매개변수를 조정하고 비상 상황에 대응하며 진화를 유도하는 거버넌스 메커니즘이 필요합니다.

이러한 메커니즘 중 일부는 현재 MAINCHAIN에 있으며 DON을 배포하더라도 동일하게 유지할 수 있습니다. 하나의 예시로 오라클 노드 공급자(DON 노드)에 대한 지불 메커니즘을 볼 수 있는데, MAINCHAIN의 DON 프론트 엔드 컨트랙트에는 주기적으로 수정될 수 있는 가드레일과 같은 추가 메커니즘이 포함되어 있습니다.

우리는 **진화적(evolutionary) 거버넌스**와 **비상(emergency) 거버넌스**가 거버넌스 메커니즘의 주요한 유형이 될 것으로 예상합니다.

- **진화적 거버넌스:** 체인링크 생태계에 대한 많은 수정에서 구현 자체는 긴급한 문제가 아니며, 중요한 것은 성능 향상, 기능 향상, (긴급하지 않은) 보안 업그레이드 등입니다. 체인링크의 거버넌스에 더 많은 참가자가 참여하면서, 이러한 변경 사항의 대부분은 변경 사항의 영향을 받는 특정 DON의 커뮤니티에서 의해 비준될 것으로 예상합니다. 그동안 아마도 궁극적으로 병렬 메커니즘으로써, 우리는 일시적인 최소 권한이 진화적 거버넌스를 구현하는 유용한 수단이 될 수 있다고 믿습니다. 간단하게 말하면, 이 아이디어는 변경 사항을 점진적으로 배포하여 커뮤니티가 이에 대응할 기회를 보장하는 것입니다. 예를 들어 새 MAINCHAIN 컨트랙트로의 마이그레이션이 활성화되기 최소 30일 전에 새 컨트랙트를 배포하는 조건으로 제한할 수 있습니다.
- **비상 거버넌스:** 악용할 수 있거나, 악용되는 MAINCHAIN 컨트랙트의 취약점이나 활성 형태, 안전 관련 실패는 치명적인 결과를 예방하기 위해 즉각적인 개입이 필요할 수 있습니다. 이러한 특정 조직의 불법 행위를 방지하기 위해 우리는 서명자가 조직 전체에 분산되도록 하는 다중 서명 개입 메커니즘을 지원합니다. 서명자의 일관된 가용성과 비상 변경 승인이 필요할 때 적합한 명령 체인의 적시 액스를 보장하려면, 신중한 운영 계획과 정기적인 검토가 분명 필요합니다. 이러한 과제는 경계 감소[223]와 같은 일반적인 문제에 대처할 때와 비슷한데, 다른 사이버 보안 사고 대응 기능[134]을 테스트할 때와 같은 양상을 보입니다.

DON 거버넌스는 잠재적 이질성 정도에 있어 많은 탈중앙시스템의 거버넌스와 차별적입니다. 각 DON에는 고유한 데이터 소스, 실행파일, 가동 시간과 같은 서비스 수준 요구 사항 및 사용자가 있을 수 있습니다. 체인링크 네트워크 거버넌스 메커니즘은 운영 목표 및 매개변수 변동을 수용할 수 있을 만큼 충분히 유연해야 합니다. 우리는 설계 아이디어를 적극적으로 탐색 중이며 향후 관련 주제에 관한 연구를 발표할 계획입니다.

7.5 공개키 인프라

점진적인 탈중앙화에는 DON 노드를 포함한 네트워크 참여자에 대한 확실한 식별이 요구됩니다. 특히 체인링크는 강력한 공개키 인프라(PKI, Public-Key Infrastructure)가 필요한데, 여기서 PKI는 키를 ID에 바인딩하는 체계를 의미합니다. 예컨대, PKI는 인터넷의 보안 연결(TLS) 시스템을 뒷받침하며 HTTPS(예: <https://www.chainlinklabs.com>)를 통해 웹사이트에 연결하고 브라우저에 잠금이 표시되면 도메인 소유자의 공개키가 인가 받아(구체적으로는 소위 인증서(certificate)의 디지털 서명으로) 해당 소유자에게 바인딩되었음을 의미합니다. 최상위 루트 권한이 널리 사용되는 브라우저에 고정되어 있는 인증 기관(CA)의 계층적 시스템은 인증서가 합법적인 도메인 소유자에게만 발급되도록 합니다.

우리는 체인링크가 궁극적으로 PKI에 기반하여 초기에는 탈중앙화 네임 서비스인 이더리움 네임 서비스(ETH Name Service, ENS)[22]를 사용하게 될 것으로 예상합니다. 그 이름에서 알 수 있듯, ENS는 DNS와 유사합니다. DNS는 (사람이 읽을 수 있는) 도메인 이름을 인터넷의 IP 주소에 매핑하는 도메인 네임 시스템입니다. 대신 ENS는 사람이 읽을 수 있는 ETH 이름을 블록체인 주소에 매핑합니다. ENS는 ETH 블록체인에서 작동하기 때문에 주요 손상을 제외하면 네임스페이스를 변조하는 것은 원칙적으로 이를 관리하는 컨트랙트나 기본 블록체인을 변조하는 것만큼 어렵습니다. (반면 DNS는 그동안 스푸핑, 하이재킹 및 기타 공격에 취약한 모습을 보여왔습니다.)

우리는 ETH 메인넷 상에 ENS로 `data.eth`를 등록했으며 오라클 데이터 서비스 및 기타 체인링크 네트워크 엔티티 ID가 상주하는 루트 네임스페이스로 설정하려고 합니다.

ENS의 도메인은 계층적입니다. 즉, 각 도메인은 그 아래에 있는 다른 이름에 대한 참조를 포함할 수 있습니다. ENS 하위 도메인은 신뢰를 구성하고 위임하는 방법으로 사용될 수 있습니다.

`data.eth`의 주요 역할은 데이터 피드를 위한 온체인 디렉토리 서비스입니다. 전통적으로 오라클 개발자와 사용자는 오프체인 소스(예: `docs.chain.link` 또는 `data.chain.link`와 같은 웹사이트 또는 Twitter와 같은 소셜 네트워크)를 사용하여 오라클 데이터 피드 주소(예: ETH-USD 가격 피드)를 발행 및 획득했습니다. `data.eth`와 같은 매우 신뢰할 수 있는 루트 네임스페이스를 사용하면, 예컨대 ETH-USD 가격 피드 온체인 오라클 네트워크 수집기의 스마트 컨트랙트 주소 `eth-usd.data.eth` 매핑을 대신 설정할 수 있습니다. 이는 누구나 블록체인을 해당 가격/이름 쌍(ETH-USD)의 데이터 피드에 대한 증거(source of truth)로 참조할 수 있게 하는 안전한 경로를 제공할 것입니다. 결과적으로 이러한 ENS의 사용은 오프체인 데이터 소스에서는 불가능한 두 가지 이점을 실현합니다.

- **강력한 보안:** 도메인의 모든 변경 및 업데이트는 두 가지 보안 속성을 모두 사용하지 않는 웹사이트의 텍스트 주소와 달리 변경 불가능하게 기록되고 암호로 보호됩니다.
- **자동화된 온체인 전파:** 데이터 피드 스마트 컨트랙트의 기본 주소 업데이트는 종속 스마트 컨트랙트로 전파되는 알림을 작동시킬 수 있습니다. 예를 들면 종속 컨트랙트를 새 주소로 자동 업데이트할 수 있습니다.¹³

그러나 ENS와 같은 네임스페이스는 주장된 이름의 합법적 소유권을 자동으로 확인하지 않습니다. 따라서 예를 들어 네임스페이스에 해당 항목이 포함된 경우,

('Acme Oracle Node Co.' , addr)

사용자는 addr이 Acme Oracle Node Co.라는 이름의 주체에 속한다는 보증을 얻습니다. 그러나 네임스페이스 관리에 대한 추가 메커니즘이 없다면, 실제 차원에서 어떤 이름이 Acme Oracle Node Co.라는 합법적인 주체에 속한다는 보증을 얻지 못합니다.

이름 검증에 대한 우리의 접근 방식, 즉 합법적인 실제 주체에 의한 소유권을 보장하기 위한 접근 방식은 여러 구성요소에 의존합니다. 현재 체인링크 랩스는 체인링크 네트워크의 CA 역할을 효과적으로 수행하고 있습니다. 체인링크 랩스는 계속해서 이름을 검증할 것이며, 우리의 PKI는 다음 두 가지 방식의 보다 탈중앙화된 모델로 발전할 것입니다.

- **신뢰 웹(Web-of-trust) 모델:** 계층적 PKI를 탈중앙화한 결과는 종종 신뢰 웹(Web-of-Trust)¹⁴라고 일컬어집니다. 1990년대부터 다양한 변형이 제안되었으며[98], 블록체인이 전 계적으로 일관된 원장에 인증서를 기록하여 이러한 아이디어의 실현을 용이하게 할 수 있음을 많은 연구자가 확인한 바 있습니다[227]. 우리는 이 모델의 변형을 탐색하여 보다 탈중앙화된 방식으로 체인링크 네트워크에 있는 엔티티의 ID를 검증하고 있습니다.
- **검증 데이터의 연결:** 오늘날 상당한 양의 오라클 노드 성능 데이터가 온체인에서 가시화되고 있으며 이에 따라 노드 주소에 아카이브로 바인딩되고 있습니다. 이러한 데이터는 네트워크에 대한 (신뢰할 수 있는) 참여 관련 기록 증거를 제공하므로, PKI의 ID를 강화할 가능성이 있습니다. 또한 DECO 및 타운크라이어[160]를 기반으로 하는 탈중앙 ID용 도구를 사용하면 노드가 실제 데이터에서 파생된 자격 증명을 추적할 수 있습니다. 한 가지 예로, 노드 운영자는 던 앤 브래드 스트리트(Dun and Bradstreet) 평가 등급을 받은 자격 증명을 PKI ID에 첨부할 수 있습니다. 이러한 추가적인 유효성 검사는 예치를 보완해 네트워크 보안 보장을 꺾을 수 있습니다. 실제에서 확립된 신원을 가진 오라클 노드는 그 평판으로 미루어 시스템에 예치가 있는 것으로 간주할 수 있습니다. (선 4.3 및 선 9.6.3 참조)

체인링크 PKI에서 마지막으로 요구되는 바는 **보안 부트스트래핑**입니다. 이는 현재 `data.eth`인 체인링크 네트워크의 루트 이름을 안전하게 게시하는 것을 의미합니다(브라우저에서 최상위 도메인의 직접 연결(hardwiring)과 유사). 이는 `data.eth`가 실제로 체인링크 프로젝트와 관련된 최상위 도메인인지 체인링크 사용자가 어떻게 결정할 수 있는지의 문제를 의미합니다. 체인링크 네트워크와 관련한 이 문제에 대한 솔루션은 다각적이며 다음이 포함될 수 있습니다.

- 체인링크 생태계의 루트 도메인으로 `data.eth`를 지정하는 `chain.link` 도메인 레코드에 TXT 기록을 추가하는 방법[224] (따라서, 체인링크는 인터넷 도메인용 PKI를 암묵적으로 활용하여 루트 ENS 도메인의 유효성을 검사하게 됨)
- 체인링크의 기존 웹사이트(예: <https://docs.chain.link>)에서 `data.eth`로 연결하는 방법 (인터넷 도메인에 대한 PKI의 또 다른 암묵적 사용을 의미함)
- 이 백서를 포함하여 다양한 문서를 통해 알려진 `data.eth`를 사용하는 방법
- Twitter 및 체인링크 블로그 등 소셜 미디어 채널에 `data.eth`를 공개적으로 게시하는 방법[18]
- `data.eth`와 동일한 등록자 주소의 제어 하에 대량의 LINK를 배치하는 방법

8. DON 배포 관련 고려 사항

이 장은 핵심 설계의 일부는 아니지만 DON을 실현하는 데 몇 가지 중요한 기술적 고려 사항을 다룹니다.

8.1 롤아웃(Rollout) 접근 방식

이 백서에서는 과정상 발생하는 많은 문제의 솔루션을 요하는 고급 체인링크 기능과 관련하여 야심 찬 비전을 제시했습니다. 이 백서에서는 몇 가지 문제에 대해 논의할 것이며, 예상치 못한 문제가 발생할 수도 있습니다.

우리는 이러한 비전에 따라 관련 요소를 장기간에 걸쳐 점진적인 방식으로 구현할 계획이며, DON이 초기에 체인링크 커뮤니티 팀이 공동 작성한 특정 사전 빌드 구성요소(pre-built components)를 지원하는 형태로 출시될 것이라 예상합니다. 이는 DON의 광범위한 사용(예: 임의의 실행파일을 시작하는 기능)을 이후에도 지원하기 위함입니다.

이렇게 주의를 기울여야 하는 이유는 최근의 플래시 대출 기반 공격에서 볼 수 있듯, 스마트 컨트랙트 구성이 복잡하고 의도치 않은 위험한 부작용을 초래할 수도 있기 때문입니다[127, 189]. 따라서 스마트 컨트랙트, 어댑터 및 실행파일 구성 시 각별한 주의가 필요합니다.

DON 초기 배포에는 사전에 빌드된 템플릿형 실행파일 및 어댑터 집합만이 포함될 계획입니다. 이는 공식적인 방법[46, 170] 및 기타 접근 방식을 통해 이러한 기능의 구성 보안에 관한 연구를 가능하게 할 것이며 가격 책정도 단순화시킬 것입니다. 기능성 가격 정책(Functionality pricing)은 예컨대 [156]에서 채택된 접근 방식인 일반화된 측정을 통하지 않고 기능성(perfunctionality)을 기준으로 DON 노드에 의해 설정될 수 있습니다. 또한 체인링크 커뮤니티가 추가 템플릿 생성에 참여하여 다양한 어댑터와 실행파일을 점점 더 유용한 탈중앙화 서비스로 결합하여 수천 명의 개별 DON까진 아니어도 수백 명 정도는 실행할 수 있게 될 것으로 기대합니다.

또한, 이 접근 방식은 상태 팽창(state bloat), 즉 DON 노드가 작업 메모리 상 작업 불가한 양의 상태를 유지하는 것을 방지하는 데 도움이 될 수 있습니다. 이는 ‘무상태 클라이언트(stateless client)’와 같은 접근 방식을 유발하는 무허가 방식의 블록체인에서 이미 발생하고 있는 문제입니다(예: [206] 참조). DON이 상태 크기에 최적화된 실행파일만 배포하는 접근 방식을 유도하는 더 높은 처리량의 시스템이라면 이러한 문제는 더 심각할 수 있습니다.

DON이 발전과 성숙을 거듭하며, 섰 7에서 논의된 강력한 가드레일, 섰 9에서 논의된 암호화 경제 및 평판 기반 보안 메커니즘, DON 사용자에게 높은 수준의 보증을 제공하는 기타 기능을 포함하게 되면, 커뮤니티가 스스로 프레임워크 및 도구를 개발해 DON의 광범위한 출시 및 사용을 촉진할 것입니다. 이상적으로는 노드 운영자 집합이 오라클 네트워크에 모여 이러한 도구를 사용해 무허가 또는 셀프서비스 방식으로 자체 DON을 (단독으로) 시작하리라 기대할 수 있습니다.

8.2 동적인 DON 멤버십

특정 DON을 실행하는 노드 집합은 시간 경과에 따라 변경될 수 있습니다. O 에서 동적 멤버십이 주어질 경우, skL 키 관리는 두 가지 접근 방식으로 가능합니다.

첫 번째 pkL 는 변경하지 않고 구성원이 변경되면 노드가 보유한 skL 지분을 업데이트하는 것입니다. 이 방식의 이점은 [41, 161, 198]에서 살펴본 바와 같이, 의존 당사자에게 pkL 업데이트를 수행하도록 요구할 필요가 없다는 것입니다.

[122]에서 소개된 고전적인 지분 재공유 (share resharing) 기법은 이러한 지분 업데이트를 실현하는 간단하고 효율적인 방법입니다. 이는 하나의 노드 집합 $O(1)$ 과 두 번째 노드 집합 $O(2)$, 아마도 하나의 $O(2)$ 를 교차하는 노드 사이에 비밀을 전송할 수 있게 해줍니다. 이 접근 방식에서 각 노드 $O_i(1)$ 은 $n(2) = |O(2)|$ 및 원하는 (아마도 새로운) 임계값 $k(2)$ 를 위해 $O(2)$ 노드 사이에 그 비밀 중 $(k(2), n(2))$ 에 대한 비밀 공유를 수행합니다. 다양하게 검증 가능한 비밀 공유(Various verifiable secret sharing, VSS) 체제[108]에서는 노드를 적극적으로 손상시키는 공격자, 즉 프로토콜에 악의적인 행동을 도입하는 공격자에 대항하는 보안을 제공할 수 있습니다. [161]이 제시한 기법도 이를 목표로 삼는데, 통신의 복잡성을 줄이면서도 암호학적 견고성 가정(cryptographic hardness assumptions)의 실패에 대한 회복성을 제공합니다.

두 번째 접근 방식은 원장키 pkL 을 업데이트하는 것인데, 이는 순방향 보안(forward security) 이점을 제공합니다. 따라서 pkL 이전 지분(즉, 이전 위원회 노드) 손상은 현재 키의 손상으로 이어지지 않습니다. 그러나 pkL 업데이트에는 두 가지 단점이 존재합니다. (1) pk 로 암호화된 데이터는 키 새로 고침 중 다시 암호화되어야 하고 (2) 키 업데이트는 신뢰 당사자에게 전파되어야 합니다.

우리는 이 두 가지 접근 방식과 둘을 혼합한 방식을 모두 탐색하고자 합니다.

8.3 DON 책임성

기존 체인링크 오라클 네트워크와 마찬가지로 DON에는 올바른 노드 동작을 기록, 모니터링 및 시행하는 등의 책임성(accountability)을 지원하는 메커니즘이 포함됩니다. DON은 특히 외부의 탈중앙화된 스토리지와의 연결 기능을 고려할 때, 기존의 많은 무허가 방식의 블록체인보다 훨씬 더 많은 데이터 용량을 제공할 수 있을 것입니다. 이에 따라 노드의 실적 내역을 자히 기록할 수 있게 될 것이며, 보다 밀한 책임성 메커니즘을 구현하게 될 것입니다. 예컨대, 자산 가격의 오프체인 연산에는 중앙값 결과가 체인으로 전송되기 전에 폐기되는 입력이 포함될 수 있습니다. DON에서는 이러한 중간 결과를 기록할 수 있어, DON의 개별 노드에 의한 오작동 또는 성능 저하가 DON에서 분화된 방식으로 수정되거나 처벌될 수 있습니다. 우리는 앞서 7.3절에서 시스템 장애의 컨트랙트별 영향을 처리하는 가드레일 구축에 대한 접근 방식을 추가로 논의한 바 있습니다.

그러나 지금 설명하는 것처럼 DON 자체에 대한 안전 장치(failsafe) 메커니즘, 특히 분기/모호화 및 SLA(서비스 수준 계약) 오류와 같은 시스템 상의 잠재적으로 치명적인 DON 오류에 대한 보호 기능을 보유하는 것도 중요합니다.

포크/모호화: 결함이 있는 노드가 충분히 많이 주어진다면, DON은 분기하거나 모호화가 가능하며, 이 경우 L 내에 두 개의 일관성 없는 별개 블록 또는 블록 시퀀스가 생성될 수 있습니다. 그러나 DON은 L 의 콘텐츠에 디지털 서명을 하기 때문에 메인 체인 MAINCHAIN을 활용하여 모호성을 방지하고 처벌할 수 있습니다.

DON은 MAINCHAIN 감사 컨트랙트 내 L 로 주기적인 상태 확인이 가능합니다. 미래 상태가 확인했던 상태에서 벗어나면 사용자/감사자는 오작동 증거를 감사 컨트랙트에 제출할 수 있습니다. 이 증거는 경보를 생성하거나 컨트랙트에서 삭감하여 DON 노드에 불이익을 주는 데 사용할 수 있습니다. 후자의 접근 방식은 특정 오라클 피드와 유사한 인센티브 설계 문제를 도입되며 섹션 9에 설명된 작업을 기반으로 구성될 수 있습니다.

서비스 수준 계약의 시행: DON이 반드시 무한정 실행될 필요는 없지만, 사용자와의 서비스 수준 계약(service level agreements, SLA)을 준수해야 할 것입니다. 기본 SLA의 시행은 메인 체인에서 가능합니다. 예컨대, DON 노드는 특정 날짜까지 DON을 유지하거나 서비스 종료에 대한 사전 통지(예: 3개월 통지)를 제공하기로 약정할 수 있습니다. MAINCHAIN 상에서의 컨트랙트에서는 기본적인 암호화 경제 차원의 SLA 시행을 제공할 수 있습니다.

예컨대, SLA 컨트랙트는 체크포인트가 필요한 간격으로 제공되지 않을 때, DON에 예치된 자금을 삭감할 수 있습니다. 사용자는 자금을 예치하고 체크포인트가 유효 블록 시퀀스를 올바르게 나타내는지 증명하도록 DON에게 요구할 수 있습니다(예: [141]과 유사한 방식을 사용). 물론 블록 생산은 트랜잭션 처리와 동일하지 않지만, SLA 컨트랙트는 후자를 시행하는 역할도 할 수 있습니다. 예컨대, 댄풀(mempool)에서 트랜잭션을 가져오는(섹션 5.2 참조) FSS의 레거시 호환 버전에서 트랜잭션은 언젠가 채굴되어 체인에 배치될 것입니다. 사용자는 적절한 시간 간격 내 대상 컨트랙트에서 처리하기 위해 채굴되었지만, DON에 의해 전송되지 않은 트랜잭션을 SLA 컨트랙트에 제공함으로써 DON의 불법 행위를 입증할 수 있습니다.

또한, 실행파일을 사용한 연산 오류나(예: 섹션 6.3에 설명된 올바른 오프체인 상태 트랜잭션을 증명하는 메커니즘) DON상에서 보이는 개시자 기반 실행파일 실행 실패, DON의 데이터를 적시에 MAINCHAIN으로 전달하지 못한 실패 등 보다 분화된 SLA 실패를 입증하고 처벌할 수 있습니다.

9. 경제학 및 암호경제학

체인링크 네트워크가 탈중앙화된 신뢰 모델 내에서 강력한 보안을 달성하기 위해 노드는 반드시 집합적으로 올바른 동작을 나타내야 합니다. 즉, 대부분의 시간동안 DON 프로토콜을 정확히 준수해야 합니다. 이 섹션에서는 경제적 인센티브(이른바 암호경제학적 인센티브)를 통해 이러한 행동을 시행하는 데 도움이 되는 접근 방식을 논의합니다. 이러한 인센티브는 명시적 및 암묵적 인센티브라는 두 가지 범주로 나뉩니다. 이는 각각 예치 및 미래 수수료 기회(future fee opportunity, FFO)를 통해 실현됩니다.

예치: 다른 블록체인 시스템과 마찬가지로 체인링크에서의 예치는 해당 네트워크 참여자(즉, 오라클 노드)가 LINK 토큰의 형태로 고정 자금을 맡기는 과정을 수반합니다. 예치(stake) 또는 명시적 예치(explicit stake)라고도 하는 자금은 명시적인 인센티브(explicit incentive)라고 할 수 있습니다. 이들 자금은 노드 장애 또는 불법 행위가 발각되면 몰수될 수 있습니다. 블록체인에서는 이러한 절차를 종종 삭감(slash)이라고 표현합니다. 그러나 체인링크 오라클 노드에 의한 예치는 무허가 블록체인 검증자에 의한 예치와 근본적으로 다릅니다. 검증자는 트랜잭션을 모호화하거나 적대적 순서화 등 불법 행위를 저지할 수 있지만, 무허가 블록체인의 기본 컨센서스 프로토콜은 검증자가 유효하지 않은 블록을 생성하지 못하도록 강력하고 빠른 블록 검증 규칙과 암호화 프리미티브를 사용합니다. 대조적으로, 프로그래밍 방식의 보호는 부정을 저지르는 오라클 네트워크가 잘못된 보고서를 생성하는 것을 방지할 수 없습니다. 그 이유는 두 가지 유형

의 시스템 간 주요한 차이에 있습니다. 블록체인 트랜잭션 유효성 검사는 내부 일관성 속성이지만, 블록체인에 대한 오라클 보고서의 정확성은 외부 속성, 즉 오프체인 데이터이기 때문입니다.

우리는 외부 데이터를 사용할 수 있는 오라클 노드 간 대화형 프로토콜을 기반으로 체인링크 네트워크에 대한 예비 예치 메커니즘을 설계했습니다. 이 메커니즘은 명시적 보상과 벌칙(삭감)을 사용하여 올바른 행동에 재정적 인센티브를 부여합니다. 이 메커니즘은 경제 구조에 기반하여 금전적 자원, 즉 뇌물(bribery)을 통해 노드를 손상시키는 공격자를 방지하도록 설계됩니다. (이러한 공격자 유형은 매우 일반적이며 널리 퍼지기도 합니다. 예를 들어, 집단적 잘못된 행동으로 이익을 얻기 위해 서로 협력하는 노드들의 경우가 있습니다.) 우리가 설계한 체인링크 예치 메커니즘에는 강력하고 새로운 기능이 있습니다. 이 기능은 초선형적 예치 효과(super-linear staking impact) (특히, 제곱)입니다. 공격자가 메커니즘을 전복시키기 위해서는 노드 예치금을 크게 초과하는 리소스를 보유해야 합니다. 하지만 우리 예치 메커니즘은 이전에 유사한 시스템에서 고려한 것보다 더 강력한 적, 즉 노드의 미래 행동에 대한 뇌물 조건을 만들 수 있는 적에 대한 보호를 추가로 제공합니다. 또한 DECO와 같은 체인링크 도구가 결함이 있는 노드 동작 사례에서 올바른 판단을 촉진하여 예치 메커니즘을 강화하는 데 어떻게 도움이 되는지 논의합니다.

미래 수수료 기회(FFO): PoW와 PoS가 모두 존재하는 무허가 블록체인은 오늘날 우리가 암묵적 인센티브(implicit incentives)라고 부르는 것에 크게 의존합니다. 이는 명백한 보상에서 촉발된 것이 아닌, 플랫폼 참여 자체에서 파생되는 정직한 행동에 대한 경제적 인센티브입니다. 예컨대, BTC 채굴자 커뮤니티는 BTC에 대한 신뢰를 약화시키고 가치를 떨어뜨려 결과적으로 채굴 기반 시설에 대한 집합적 자본 투자의 가치를 잠식할 위험이 있는 51% 공격에 대항하는 것에 대해 인센티브를 받습니다[150]. 체인링크 네트워크는 이와 유사하게 미래 수수료 기회(FFO)라고 부르는 암묵적 인센티브를 제공합니다. 강력한 성능 이력이나 평판을 가진 오라클 노드는 사용자로부터 수수료를 받습니다. 오라클 노드의 잘못된 행동은 향후 수수료 지불을 위태롭게 하고 네트워크 참여를 통해 얻을 수 있는 잠재적 수익 측면에서 기회비용으로 노드에 불이익을 주게 됩니다. 명시적 이익에서 유추해 볼 때 FFO는 암묵적 이익의 한 형태로 볼 수 있으며, 노드 운영자의 비즈니스가 의존하는 플랫폼에 대한 신뢰를 유지하는 공유 이익, 즉 네트워크의 긍정적 성과와 평판에서 파생되는 정직한 행동에 대한 인센티브입니다. 이 인센티브는 체인링크 네트워크 프로토콜에 내재되어 있지만 명시적으로 표현되지는 않습니다. BTC에서 위에서 언급한 채굴 작업의 가치를 유지하는 것도 일종의 암묵적 이익으로 볼 수 있습니다.

우리는 FFO가 이미 체인링크에 존재하며 오늘날 네트워크를 보호하는 데 도움이 된다는 점을 강조합니다. 체인링크의 추가 개발에 있어 우리가 주요하게 기여하는 바는 소위 암묵적 인센티브 프레임워크(Implicit-Incentive Framework, IIF)를 통한 FFO와 같은 암묵적 인센티브를 평가하는 원칙적이고 경험적으로 주도된 접근 방식이 될 것입니다. 노드의 미래 수수료 기회와 같은 정량적 추정을 위해 IIF는 체인링크 네트워크에서 수집한 포괄적인 성능 및 지불 데이터를 지속해서 도출할 것입니다. 이러한 추정은 현재의 휴리스틱 모델이나 정적 모델보다 더 정확하게 노드 인센티브를 반영하는 스테이킹 시스템의 IIF 기반 매개변수화를 가능하게 합니다.

요약하면, 현재 개발 중인 체인링크 네트워크에서 제공하는 올바른 오라클 노드 동작에 대한 두 가지 주요 경제적 인센티브는 다음과 같습니다.

- 예치(예치금) → 명시적 인센티브
- 미래 수수료 기회(FFO) → 암묵적 인센티브

이 두 가지 형태의 인센티브는 상호 보완적입니다. 오라클 노드는 체인링크 예치 프로토콜에 참여하여 사용자로부터 지속적인 수익을 받고, 꾸준한 선의의 행동을 통해 집단적으로도 이익을 얻을 수 있습니다. 따라서 이 두 인센티브는 모두 오라클 네트워크가 제공하는 암호경제학적 보안에 이바지하게 됩니다. 또한, 서로를 강화 및 상쇄할 수 있습니다. 예컨대, 실적 내역과 수익원이 없는 새로운 오라클 운영자는 정직한 행동을 보장하기 위해 많은 양의 LINK를 예치하여 사용자와 수수료를 유지할 수 있습니다. 반대로, 비교적 결함이 없는 장기간 실적 이력을 가진 기존 오라클 운영자는 대규모 사용자로부터 상당한 수수료를 받을 수 있으므로 암묵적 인센티브의 한 형태인 FFO에 더 많이 의존할 것입니다.

일반적으로 여기에서 고려하는 접근 방식은 일정량의 오라클 네트워크 리소스를 목표로 하며 이를 통해 합리적인 에이전트(즉, 재정적 유용성을 극대화하는 노드)가 정직하게 행동하도록 체인링크에서 가능한 최대의 경제적 인센티브를 생성하고자 합니다. 다르게 말하자면, 이 목표는 공격자가 네트워크를 성공적으로 공격하는 데 필요한 재정적 자원 기준을 최대화하는 것입니다. 수학적으로 잘 정의된 경제적 보안 체계를 가진 예치 프로토콜을 공식화하고 IIF를 사용함으로써, 우리는 체인링크의 인센티브 강도를 가능한 한 정확하게 측정하고자 합니다. 이를 통해 해당되는 컨트랙트의 생성자는 오라클 네트워크가 필요한 수준의 암호경제학적 보안을 충족하는지 여부를 확신하여 결정할 수 있습니다.

경제적 보안의 선순환: 이 선에서 논의하는 인센티브인 예치 및 FFO는 DON의 보안 강화를 초월하는 영향을 미칩니다. 이는 이른바 경제적 보안의 선순환(The virtuous cycle of economic security)을 유도합니다. 초선형적 예치 효과(및 기타 규모의 경제)는 DON의 보안이 성장함에 따라 운영 비용을 낮추게 됩니다. 더 낮은 비용은 DON에 추가 사용자를 끌어들이고 수수료 지분을 증가시킵니다. 수수료 지분의 증가는 네트워크의 성장을 계속 장려하여 결과적으로 선순환을 영구화하게 됩니다. 우리는 경제적 보안의 선순환이 이 선 뒷부분에서 논의할 규모의 경제와 네트워크 효과의 한 가지 예에 지나지 않는다고 믿습니다.

섹션 구성 (Section organization): 예치는 주목할 만한 기술적/개념적 문제를 제시합니다. 이러한 문제에 대해 우리는 새로운 기능을 가진 메커니즘을 이미 고안한 바 있습니다. 따라서 예치는 이 선의 주요한 초점이 될 것입니다. 섹션 9.1에서는 이 백서에서 소개하는 예치 접근 방식에 대한 개요를 제공하고 섹션 9.2 내지 9.5에서는 관련된 상한 논의를 제공할 것입니다. 섹션 9.6에서는 IFF를 소개하며 섹션 9.7에서는 체인링크 네트워크 인센티브를 요약할 것입니다. 섹션 9.8에서는 우리가 제안한 예치 접근 방식이 오라클 네트워크에 가져올 수 있는 경제적 보안의 선순환 효과를 논의하겠습니다. 마지막으로 섹션 9.9에서는 체인링크 네트워크의 성장을 촉진하는 다른 잠재적 효과에 대해 간략하게 설명하겠습니다.

9.1 예치 개요

앞서 언급한 바와 같이, 여기서 소개하는 예치 메커니즘 설계는 외부 데이터 보고 불일치를 해결할 수 있는 오라클 노드 간의 대화식(interactive) 프로토콜이 포함됩니다. 예치는 합리적인 오라클 노드의 정직한 행동을 보장하는 것을 목표로 합니다. 따라서 우리는 예치 프로토콜을 공격하는 공격자를 하나의 뇌물 제공자(briber)로 가정합니다. 공격자의 전략은 재정적 인센티브를 사용하여 오라클 노드를 손상시키는 것입니다. 예를 들어, 공격자는 창출된 이익을 손상된 노드와 공유하겠다는 제안으로 오라클 보고서의 변조를 성공시켜 재정적 리소스를 얻을 수 있습니다.

우리는 두 가지 야심찬 목표를 동시에 달성하는 예치 메커니즘 설계를 목표로 합니다.

1. 강력한 공격자에 저항 : 예치 메커니즘은 사후에 신원이 확인된 오라클에 뇌물을 제공하는(우선순위가 높은 경보를 위해 무작위 선택된 오라클에 뇌물을 제공하는) 잠재적 뇌물(prospective bribery)을 포함하여 복잡한 조건부 뇌물 제공 전략을 수행 가능한 광범위한 공격자로부터 오라클 네트워크를 보호하도록 설계되었습니다. 다른 오라클 설계에서는 실제 공격자의 완전한 능력을 고려하지 않는 좁은 범위의 공격 트만을 고려해왔으나, 우리가 아는 한 여기에서 소개할 적대적 메커니즘은 해당 모델에서 최초로 광범위한 뇌물 전략을 명시적으로 다루고 저항을 제시했습니다. 우리의 모델은 공격자 외의 노드가 경제적으로 합리적이라고 가정하고(정직한 것과는 대조적으로), 일반적으로 사용하기엔 너무 고비용이지만 불일치에 사용할 수 있는 증거(source of truth)의 존재를 가정합니다(이에 대해서는 아래에서 추가로 논의함).
2. 초선형 예치 효과 달성 : 우리의 목표는 합리적인 에이전트로 구성된 오라클 네트워크가 전체 네트워크에 의한 총예치금에 있어 초선형적인 예산을 가진 공격자가 있는 경우에도 진실하게 보고하도록 하는 것입니다. 기존 예치 시스템에서 n 의 노드 각각이 $\$d$ 을 예치할 경우, 공격자는 약 $\$dn$ 에 달하는 총예산을 사용하여 각 노드에 $\$d$ 보다 약간 더 많은 금액을 지불하는 대가로 신뢰할 수 있는 뇌물을 발행하여 해당 노드에게 부정직하게 행동하도록 요청할 수 있습니다. 공격자는 네트워크에 있는 모든 예치자의 합산 예치금 순서에 따라 유동적으로 대응할 예산을 가져야 하므로 이 자체로 이미 높은 기준이 될 것입니다. 우리의 목표는 이미 설정된 상당한 수준의 장애물보다 훨씬 더 강력한 수준의 경제적 보안을 제공하는 것입니다. 우리의 목표는 n 에서 초선형적 예산으로 일반 공격자의 보안을 달성할 수 있는 최초의 예치 시스템을 설계하는 것입니다. 현실적 고려 사항이 반영되면, 영향력은 더 낮아질 수 있지만, 아래에서 논의하게 될 것처럼 우리의 예비 설계는 $\$dn^2/2$ 보다 더 큰 차원의(즉, n 의 제곱(quadratic) 확장) 적대적 예산 요건을 달성하여, 노드가 적당한 수준의 예치금만 보유해도 뇌물을 상당 수준 무력화시킬 수 있습니다.

이 두 목표의 달성을 위해서는 인센티브 설계와 암호화의 혁신적인 조합이 필요합니다.

핵심 아이디어: 우리의 예치 접근 방식은 우리가 감시자 우선순위 (watchdog priority)라고 부르는 아이디어에 의존합니다. 체인링크 오라클 네트워크에 의해 생성되어 의존 컨트랙트로 전송된 보고서(예: 자산 가격)는 참여 노드가 기여한 개별 보고서(예: 중앙값을 취하여)에서 집계됩니다. 일반적으로 서비스 수준 계약(SLA)은 보고서에 대한 허용 가능한 편차 범위, 즉 노드의 보고서가 집계 보고서에서 얼마나 벗어날 수 있는지, 집계가 올바른 것으로 간주되기 위해 실제 값에서 얼마나 벗어나도록 허용해야 하는지를 지정합니다.

우리의 예치 시스템상 각 오라클 노드는 주어진 보고 라운드 동안 집계 보고서가 올바르지 않다고 판단되면 경보를 발생시키는 감시자 역할을 할 수 있습니다. 각 보고 라운드에서 각 오라클 노드에는 경보(존재 시)가 처리되는 순서를 결정하는 공개 우선순위가 할당됩니다. 우리의 메커니즘은 보상 집중에 목표를 합니다. 즉, 경보를 발생시키는 가장 높은 우선순위 감시자가 결함이 있는 노드의 예금을 몰수하여 얻은 전체 보상을 얻게 됩니다.

우리의 예치 시스템 설계에는 1단계인 기본 계층(default tier)과 2단계인 백스톱 계층(backstop tier)의 두 가지 계층(tier)이 포함됩니다. 1단계는 n 개의 노드 집합인 오라클 네트워크 그 자체입니다. (단순화를 위해 n 을 홀수로 가정함) 대다수의 노드가 잘못된 값을 보고하면 1단계의 감시자가 경보를 발생하도록 강력한 인센티브가 주어집니다. 경보가 발생하면 네트워크의 보고 결정이 2단계, 즉 네트워크 서비스 수준 계약(SLA)에서 사용자가 지정할 수 있는 고비용의 최대 신뢰도 시스템으로 확대됩니다. 이는 예컨대, 과거 신뢰도 점수가 높은 노드로만 구성된 시스템이거나 1단계보다 훨씬 더 많은 오라클을 가진 시스템일 수 있습니다. 또한 섹션 9.4.3에서 논의된 바와 같이 DECO 또는 타운크라이어는 2단계에서 효율적이고 결정적인 판단을 보장하는 강력한 도구 역할을 할 수 있습니다. 따라서 단순화를 위해 이 2단계 시스템이 올바른 보고 값에도

달한다고 가정합니다.

모든 보고서 생성에 2단계를 의존하는 것이 매력적으로 보일 수도 있지만, 우리 설계의 이점은 일반적으로 1 단계 시스템 운영 비용만 지불하면서 2단계 시스템 보안 속성을 일관되게 달성한다는 것입니다.

감시자 우선순위는 다음과 같은 방식으로 초선형적 예치의 영향을 초래합니다. 1단계 오라클 네트워크가 잘못된 결과를 출력하고 다수의 감시자 노드가 경보를 올리면 예치 인센티브 메커니즘은, (대부분) 오작동하는 노드의 예치금에서 인출한 $\$dn/2$ 보다 더 많은 보상을 우선순위가 가장 높은 감시자에 제공합니다. 따라서 총 보상은 이 단일 감시자에게 집중되어, 공격자가 잠재적인 감시자에게 경보하지 않도록 약속해야만 하는 최소값으로 결정되게 됩니다. 우리의 메커니즘은 더 높은 우선순위의 감시자가 뇌물을 수락하고 경보하지 않기로 선택할 경우, 모든 오라클이 감시자 역할을 할 기회를 얻을 수 있기 때문에 공격자는 경보 발생을 방지하기 위해 모든 노드에 $\$dn/2$ 보다 더 많은 뇌물을 제공해야 합니다. n 개의 노드가 존재하므로 성공적인 뇌물 제공에 대한 공격자의 필수 예산은 해당 네트워크 내 노드의 n 수의 제공에 해당하는 $\$dn^2/2$ 이상에 달합니다.

9.2 배경

예치에 대한 우리의 접근 방식은 게임 이론 및 메커니즘 설계 (game theory and mechanism design, MD) 분야의 연구에 기초합니다(이론서는 [177] 참조). 게임 이론은 전략적 상호 작용에 대해 수학적으로 형식화된 연구입니다. 이러한 맥락에서 게임은 일반적으로 실제 계에서 플레이어라고 하는 게임 참가자가 사용할 수 있는 일련의 행동을 코드화하는 이러한 상호 작용의 모델입니다. 게임은 개별 플레이어가 획득한 보수 (payoff, 플레이어가 선택한 행동 및 다른 플레이어의 행동에 따라 달라지는 보상)를 지정합니다. 아마도 게임 이론에서 연구된 게임의 가장 잘 알려진 예는 죄수의 딜레마(Prisoners' Dilemma)일 것입니다 [178]. 게임 이론가들은 일반적으로 주어진 게임에서 표현되는 균형 (equilibrium 혹은 equilibria, 존재 시)을 이해하는 것을 목표로 합니다. 균형은 한 플레이어가 일방적으로 전략에서 이탈하여 더 높은 보수를 얻을 수 없도록 하는 일련의 전략(각 플레이어당 하나씩)을 의미합니다.

한편, 메커니즘 설계는 상호 작용(및 관련 게임)의 균형이 바람직한 속성을 갖도록 인센티브를 설계하는 과학입니다. MD는 게임 이론의 역으로 볼 수 있습니다. 게임 이론의 핵심은 '인센티브와 모델이 주어질 경우의 균형'을 찾는 데 있습니다. 한편, MD의 경우에는 '바람직한 균형을 가진 게임을 유도하는 인센티브'를 찾는 것이라 할 수 있습니다. 메커니즘 설계자의 일반적인 목표는 '인센티브 호환' 메커니즘을 만드는 것인데, 이는 메커니즘(예: 경매 또는 기타 정보 도출 시스템[228])의 참가자가 특정 문제(예: 특정 항목의 가치)에 대한 진실을 보고하도록 장려됨을 의미합니다. 비크리 경매(Vickrey auction, 2차 가격)는 참가자가 항목에 대해 봉인된 입찰을 제출하고, 가장 높은 입찰자가 항목을 낙찰받지만 두 번째로 높은 입찰자의 가격만큼만 지불하는 가장 잘 알려진 인센티브 호환 메커니즘입니다[214]. 암호경제학은 암호화 기술을 활용하여 탈중앙화 시스템 내에서 바람직한 균형을 만드는 영역별 형태의 MD입니다.

뇌물과 공모(collusion)는 MD 분야 전반에 걸쳐 중대한 문제를 야기합니다. 거의 모든 메커니즘은 메커니즘에 참여하는 당사자 간의 부속 컨트랙트로 정의할 수 있는 담합이 존재할 경우 중단됩니다[125, 130]. 외부 당사자가 새로운 인센티브를 게임에 도입하는 뇌물 제공은 담합보다 훨씬 더 어려운 문제를 야기합니다. 담합은 게임 참가자 간 뇌물 제공의 특별한 경우로 볼 수 있습니다.

블록체인 시스템은 종종 금전적(디지털 자산 기반) 보수가 있는 게임으로 개념화될 수 있습니다. 간단한 예는 작업 증명(Proof-of-Work) 채굴입니다. 채굴자는 블록 채굴에 사용할 해시율을 선택할 수 있는 활동 공간을 갖습니다. 채굴의 결과는 보장된 부정적 보상(전기 및 장비 비용)과 다른 활성 채굴자 수[106, 172] 및 트랜잭션 수수료에 따라 달라지는 확률론적 긍정적 보상(채굴 보조금)입니다. 쉘링코인(SchellingCoin) [68]과 같은 클라우드소싱 오라클은 또 다른 예라 할 수 있는데, 여기서 활동 공간은 오라클이 보낼 수 있는 가능한 보고서의 집합이 되며, 보수는 오라클 메커니즘에 의해 지정된 보상이 됩니다. 예를 들어, 보수는 오라클의 보고서가 다른 보고서의 중앙값에 얼마나 가까운지에 따라 달라질 수 있습니다[26, 68, 119, 185].

블록체인 게임은 공모 및 뇌물 공격에 기회를 제공합니다. 실제로 스마트 컨트랙트는 그러한 공격을 촉진할 수도 있습니다[96, 165]. 클라우드소싱된 오라클에 대한 가장 잘 알려진 뇌물 공격은 p 플러스 엡실론($p - plus - epsilon$) 공격일 것입니다[67]. 이 공격은 플레이어가 불리언 값 보고서(즉, 거짓(false) 또는 참(true))를 제출하고, 다수가 제출한 것에 동의하는 경우 p 의 보상을 받는 쉘링코인과 유사한 메커니즘의 맥락에서 발생합니다. p 플러스 엡실론 공격에서 공격자는 다수의 제출이 참인 경우에만 거짓에 투표하는 사용자에게 $\$p + \epsilon$ 의 비용을 지불하겠다고 확실하게 약속합니다. 그 결과 모든 플레이어가 다른 플레이어가 무엇을 하는지에 관계없이 거짓을 보고하도록 인센티브를 받게 되는 균형이 달성됩니다. 결과적으로 뇌물 제공자는 실제로 뇌물을 지불하지 않고도 약속된 뇌물을 통해 노드가 거짓으로 보고하도록 유도할 수 있습니다.

오라클, 특히 클라우드소싱되지 않은 오라클의 맥락에서 다른 뇌물 전략에 대한 연구는 상당히 약한 적대적 모델로 한정됩니다. 예를 들어, PoW 조건에서 연구자들은 결과 조건부 (outcome-contingent) 뇌물, 즉 대상 메시지가 성공적으로 검열되고 개별 채굴자의 행동과 관계없이 블록에 나타나지 않는 경우에만 지불되는 뇌물을 연구한 바 있습니다[96, 165]. 오라클의 경우, p 플러스 엡실론 공격을 제외하고, 뇌물 제공자가 도출되는 결과가 아닌 개별 플레이어의 행동을 조건으로 뇌물을 보내는 엄격히 제한된 뇌물 모델만 인지되고 있습니다.

여기에서 우리는 다음 하위 섹션에 설명된 바와 같이, 강력한 적대 모델에서도 인센티브가 호환되는 정보 도출 메커니즘의 설계를 개괄하겠습니다.

9.3 모델링 가정

이 섹션에서는 1단계 오라클 노드, 2단계 레이어 (판단) 노드 및 공격자 등 우리 시스템 내 플레이어의 행동과 능력을 모델링하는 방법에 대해 설명합니다.

9.3.1 1단계 인센티브 모델: 합리적 행위자

많은 블록체인 시스템은 몇 개의 정직한 참여 노드를 가정하여 보안을 설계합니다. 노드는 재정적 이익이 아니어도 프로토콜을 따르면 정직한 것으로 정의됩니다. 작업 증명 시스템은 일반적으로 대부분의 해시 파워가 정직해야 하고, 지분 증명 시스템은 일반적으로 모든 참여 지분의 2/3 이상이 정직해야 할 것을 요구하며, 아비트럼(Arbitrum)[141]과 같은 레이어-2 시스템마저 적어도 한 명의 정직한 참가자를 요구합니다.

우리의 예치 메커니즘 모델링에서는 훨씬 더 약한 가정을 합니다. (더욱 명확히 하자면, 더 약한 가정은 더 강한 보안 속성을 의미하므로 바람직함.) 우리는 공격자가 1단계 오라클 노드의 일부(소수)를 손상, 즉 통제한다고 가정합니다. 우리는 나머지 노드를 정직한 에이전트가 아닌 합리적인 기대 효용을 극대화하는 주체로서 모델링합니다. 이 노드는 전적으로 자기 이익을 위한 재정적 인센티브에 따라 행동하여 재정적 이익이 예상되는 행동을 선택합니다. 예를 들어 노드가 정직한 행동으로 인한 보상보다 더 큰 뇌물을 제공받을 경우 뇌물을 수락하게 됩니다.

적대적 노드 관련 참고: 탈중앙화 시스템에 공통으로 적용되는 신뢰 모델링에 따라 모든 노드가 합리적(즉, 악의적인 적에 의해 통제되지 않으며 순수익을 극대화하려고 함)이라 가정합니다. 그러나 우리의 주장(특히 초선형 또는 제곱 예치 영향에서)은 적대적으로 제어되는 노드 집합이 일부 양의 상수 c 에 대해 최대 $(1/2 - c)n$ 인 경우 점근적으로 유지됩니다.

9.3.2 2단계 판정 모델: 가정에 의한 정확성

다시 말하지만, 합리적인 노드에 대한 보안을 달성하는 데 도움이 되는 우리 예치 메커니즘의 중요한 기능은 2단계 시스템입니다.

우리가 제안한 예치 메커니즘에서 모든 오라클은 메커니즘의 출력이 올바르지 않다는 경보를 표시할 수 있습니다. 경보는 높은 신뢰도의 2단계 시스템을 활성화하고 올바른 결과를 보고합니다. 따라서 우리의 접근 방식에 대한 주요 모델링 요건은 올바른 판단 (correct adjudication), 즉 2단계 시스템에 의한 올바른 보고입니다.

우리 예치 모델은 부패하지 않고 최대한 신뢰할 수 있는 진실의 소스 역할을 하는 2차 시스템을 가정합니다. 이러한 시스템은 비용이 많이 들고 느리기 때문에 일반적으로 사용하기 부적절합니다. 균형의 사례에서, 즉 1단계 시스템이 올바르게 작동할 경우, 2단계 시스템은 호출되지 않습니다. 대신, 그 존재로 높은 보증의 백스톱(backstop)을 제공함으로써 전체 오라클 시스템의 보안을 강화합니다.

고신뢰도, 고비용 판단 레이어를 사용하는 것은 대부분 사법 시스템 핵심이 되는 항소 절차와 유사합니다. 이는 오라클 시스템 설계에서도 이미 일반적으로 쓰이고 있습니다(예: [119, 185]). 섹션 9.4.3에서는 우리의 메커니즘의 2단계 실현에 대한 접근 방식을 간략하게 논의합니다.

우리 예치 프로토콜은 가정된 2단계 시스템의 올바른 판단을 오라클 노드의 올바른 보고를 시행하기 위한 신뢰할 수 있는 위협 (credible threat)으로 사용합니다. 이 프로토콜은 2단계 시스템에서 잘못된 것으로 식별된 보고서를 생성하는 오라클 노드 예치의 일부 또는 전체를 몰수합니다. 따라서 오라클 노드는 결과적인 재정적 불이익을 우려하여 오작동을 지양하게 됩니다. 이 접근 방식은 낙관적 롤업에서 사용되는 방식(예: [141, 10])과 유사합니다.

9.3.3 적대적 모델

우리 예치 메커니즘은 진실한 정보를 이끌어내는 동시에 광범위하고 잘 정의된 공격자에 대한 보안을 달성하도록 설계되었습니다. 이는 명시적인 적대적 모델을 생략하거나 공격자의 좁은 범위 하위 클래스(예: 위에서 논의한 p 플러스 엡실론)에 초점을 맞춘 이전 작업을 개선합니다. 우리 목표는 실제로 직면할 가능성이 있는 모든 공격자에 대해 공식적으로 입증된 보안을 갖춘 예치 메커니즘을 설계하는 것입니다.

우리는 $\$B\$$ 로 표시되는 고정된(매개변수화 가능한) 예산을 가지고 있는 공격자를 모델링합니다. 공격자는 네트워크의 각 오라클과 비밀리에 그리고 개별적으로 통신할 수 있으며 공개적으로 관찰할 수 있는 메커니즘의 결과에 따라 개별 오라클에 보장된 뇌물 제공을 비밀리에 제안할 수 있습니다. 뇌물을 결정하는 결과에는 예를 들어 오라클이 보고한 값, 오라클이 메커니즘에 보낸 공개 메시지(예: 경보), 다른 오라클이 보고한 값, 메커니즘이 출력한 값이 포함될 수 있습니다.

무제한으로 행동 가능한 공격자로부터 보호할 수 있는 메커니즘은 존재하지 않습니다. 따라서 우리는 일부 행동의 경우 비현실적이거나 범위를 벗어난 것으로 간주합니다. 우리는 공격자가 표준 암호화 프리미티브를 깨뜨릴 수 없으며 위에서 언급한 바와 같이 고정된(잠재적으로 큰 경우) 예산 $\$B\$$ 를 가지고 있다고 가정합니다. 또한, 공격자가 오라클 네트워크의 통신을 제어하지 않는다고 가정합니다. 특히 1단계 및 2단계 노드 간의 트래픽을 크게 지연시킬 수는 없습니다. (공격자가 이러한 통신을 관찰할 수 있는지는 아래에서 설명할 것처럼 메커니즘에 따라 달라집니다.)

비공식적으로는 위에서 언급한 것처럼 공격자가 다음을 수행할 수 있다고 가정합니다. (1) 오라클 노드의 일부를 손상 (상수 c 에 대한 $(1/2 - c)$ 부분) 시킬 수 있습니다. 즉, 노드를 완전히 통제할 수도 있습니다. (2) 위에서 설명한 대로 공격자가 지정한 결과에 따라 지불을 보장하는 식으로 원하는 노드에 뇌물을 제공할 수 있습니다.

이 백서에서는 공격자가 가진 총 뇌물 제공 능력에 대해 공식적인 모델이나 완전한 분류 체계를 제공하지는 않지만, 우리의 모델에서 다루는 뇌물의 종류를 나열해보면 다음과 같습니다. 단순화를 위해 오라클은 정확한 값(w.l.o.g.)이 참(true)인 불리언 보고서를 내보내고 최종 결과는 소비 스마트 컨트랙트에서 사용할 이러한 보고서의 집계로 계산된다고 가정합니다. 뇌물 제공자의 목적은 최종 결과가 잘못된 것, 즉 거짓(false)이 되는 것입니다.

- 무조건적 뇌물 제공자 : 거짓을 보고하는 모든 오라클에 $\$b\$$ 의 뇌물을 제공합니다.
- 확률적 뇌물 제공자 : 거짓을 보고하는 모든 오라클에 약간의 확률 q 로 뇌물 $\$b\$$ 를 제공합니다.
- 거짓 결과 조건부 뇌물 제공자 : 최종 결과가 거짓일 경우 거짓을 보고한 모든 오라클에 $\$b\$$ 뇌물을 제공합니다.
- 경보 조건 없는 뇌물 제공자 : 경보가 발생하지 않는 한 거짓으로 보고한 모든 오라클에 $\$b\$$ 뇌물을 제공합니다.
- p 플러스 앱실론 뇌물 제공자 : 대다수의 오라클이 거짓을 보고하지 않는 한 거짓을 보고한 모든 오라클에 $\$b\$$ 뇌물을 제공합니다.
- 잠재적 뇌물 제공자 : 무작위 역할을 위해 선택된 오라클에 $\$b\$$ 뇌물을 미리 제공하고 거짓으로 보고합니다. 우리가 제안한 예치 프로토콜에서 모든 노드는 잠재적인 감시자 역할을 하므로, 감시자 우선순위의 무작위화는 잠재적인 뇌물에 적합하지 않을 수 있습니다. 그러나 실제로 많은 작업 증명, 지분 증명 및 허가된 시스템은 잠재적인 뇌물에 취약합니다. 이는 적대적 모델을 가정할 때 해당 사례를 고려해, 예치 프로토콜이 탄력적으로 대응할 수 있도록 보장하는 것이 중요함을 역설합니다. 자한 내용은 부록 E를 참조하시기 바랍니다.

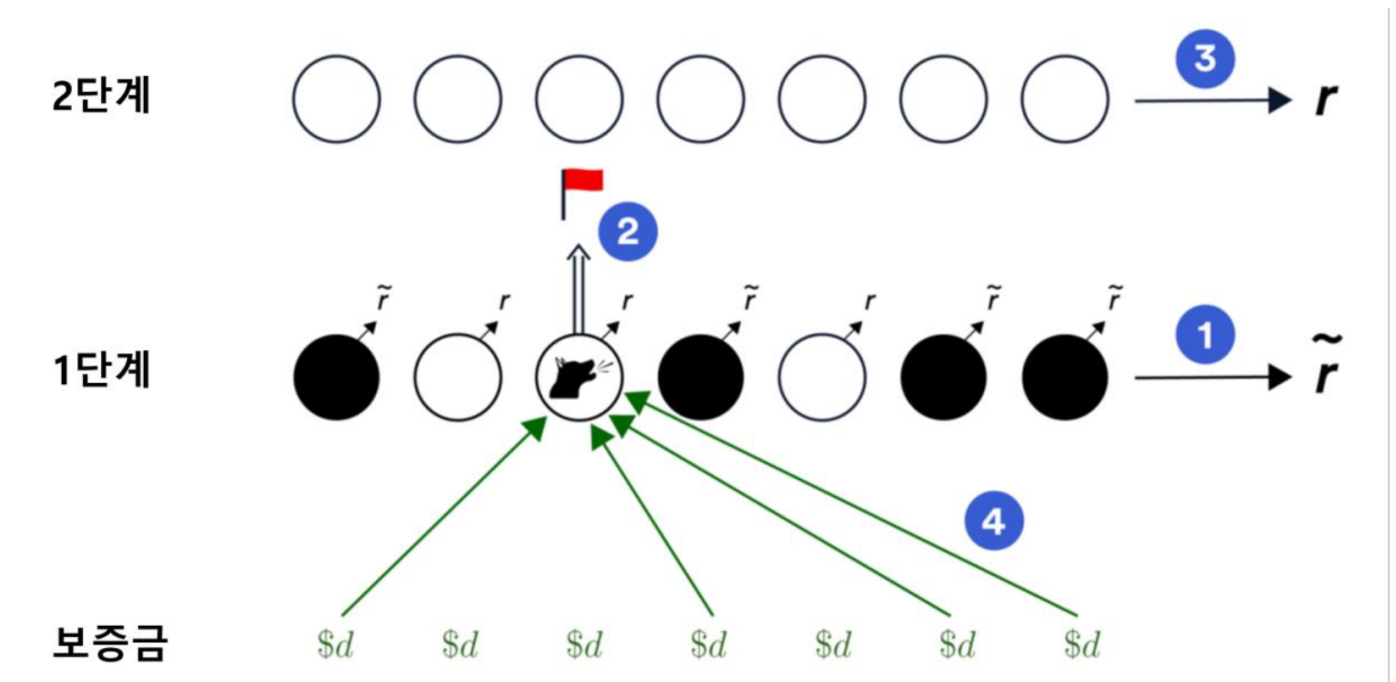
9.3.4 암호경제학적 보안은 어느 정도면 충분한가?

합리적인 공격자는 지출보다 큰 이익을 얻을 수 있는 경우에만 시스템을 공격하기 위해 돈을 지불합니다. 따라서 우리의 적대적 모델 및 제안된 예치 메커니즘에서, $\$B$ 는 공격자가 오라클 네트워크를 손상시키고 잘못된 보고서를 생성함으로써, 의존 스마트 컨트랙트에서 인출할 수 있는 잠재적 이익 척도라고 볼 수 있습니다. 사용자는 오라클 네트워크가 그 목적에 따라 충분한 수준의 암호경제학적 보안을 제공하는지를 판단할 때, 이러한 관점에서 네트워크를 평가해야 합니다.

현실에 존재하는 공격자를 고려할 때, $\$B$ 는 일반적으로 의존 스마트 컨트랙트의 총자산보다 훨씬 적을 것으로 예상됩니다. 대부분 공격자가 전체 자산을 완전히 인출하기란 불가능하기 때문입니다.

9.4 예치 메커니즘: 개요

여기에서는 우리가 현재 고려 중인 예치 메커니즘의 주요 아이디어와 일반적인 구조를 제시하겠습니다. 편의상, 단순하지만 결과 도출이 느린(다중 라운드) 프로토콜이 있다고 가정해 보겠습니다. 이 체계는 매우 실용적입니다. 메커니즘이 제공하는 경제적 보장, 즉 결함이 있는 노드에 대한 처벌과 그에 따른 인센티브를 감안해, 많은 사용자는 보고를 낙관적으로 수락할 수 있습니다. 다시 말해, 이러한 사용자는 2단계가 잠재적인 판단을 내리기 전에 보고서를 수락할 것입니다.



보고서를 낙관적으로 수락하지 않으려는 사용자는 프로토콜 실행이 종료될 때까지, 즉 2단계로 잠재적인 강화가 이루어질 때까지 대기하는 것을 선택할 수 있습니다. 그러나 이렇게 하면 보고서 확인 시간이 상당히 느려질 수 있습니다. 따라서 섹션 9.5에서는 다소 복잡한 디자인이지만 더 빠른(단일 라운드) 결과를 가져오는 몇 가지 최적화에 대해 간략하게 설명하겠습니다.

- 그림 15: 경보를 사용하는 예치 체제 구성도. 이 예시에서는 ① 대다수의 노드가 손상되었거나 뇌물을 받았고 올바른 보고 값 r 이 아닌 잘못된 값 $\tilde{r}$ 을 내보냅니다. 감시자 노드는 ② 2단계 위원회에 경보를 보내고, ③ 위원회는 정확한 보고 값 r 을 결정하고 내보냅니다. 이에 따라 손상된 노드 보증금(각 $\$d$)을 감시자 노드 ④가 몰수합니다.

다시 말하지만, 예치 메커니즘의 1단계는 기본 오라클 네트워크 자체로 구성되어 있습니다. 위에서 설명한 것처럼 우리 메커니즘의 주요한 구조는 각 라운드에서 각각의 노드가 우선순위를 부여받는 ‘감시자’로 작동할 수 있다는 것입니다. 메커니즘은 올바른 r 이 아닌 잘못된 출력 $\tilde{r}$ 에 도달하면 경보를 발생시킵니다. 이 경보는 2단계 해결을 유발하며 이는 올바른 보고서에 도달한 것으로 가정됩니다. 잘못된 보고 값이 있는 노드는 예치금이 삭감되고 감시자가 이를 가지게 된다는 점에서 처벌을 받게 됩니다. 이와 같은 기본 구조는 일반적으로 [119, 185]와 같이 오라클 시스템에서 볼 수 있습니다.

위에서 간략하게 언급한 설계의 주요한 혁신은 잠재적인 감시자의 순서대로 모든 노드에 고유한 우선순위가 할당된다는 것입니다. 즉, 감시자가 되면 우선순위에 따라 먼저 경보할 기회가 주어집니다. 가장 높은 우선순위로 경보를 발생시킬 수 있는 노드는 모든 부정행위 노드에서 삭감된 예치금 $\$d$ 를 받게 되는데, 부정확한 보고는 노드의 다수가 불량 노드라는 의미이므로, 총 금액은 $\$dn/2 = \$d \times n/2$ 이상으로 받게 될 것입니다. 결과적으로, 공격자는 임의의 노드에 뇌물을 주기 위해 최소한 이 보상에 상응하는 금액을 지불해야 합니다. 따라서 다수의 노드에 뇌물을 주려면 공격자는 거액의 뇌물, 즉 $\$dn^2/2$ 보다 많은 금액을 지불해야 합니다.

그림 15는 경보 및 감시자 강화 작업이 작동하는 방식을 개략적으로 보여줍니다.

9.4.1 메커니즘 추가 상세 정보

이제 뇌물 방지 시스템을 더 자세히 설명해볼 텐데, 이는 우리가 구축하려는 2단계 구조를 간단하게 풀어낸 것입니다. 여기서는 인센티브 메커니즘 및 2단계로의 강화 절차와 더불어 1단계 네트워크(이후 의미상 별다른 문제가 없으면 ‘네트워크’라고 지칭)를 설명하겠습니다.

불리언 값(예: BTC의 시가총액이 ETH의 시가총액을 초과하는지 여부)을 정기적으로(예: 1분에 한 번) 보고하는 n 개의 오라클 노드로 구성된 체인링크 네트워크를 가정하겠습니다. 예치 메커니즘에 따라, 노드는 두 가지 예치금을 제공해야 합니다. 즉, 다수와 동의하지 않을 경우 삭감될 수 있는 $\$d$ 예치금과, 잘못된 강화(faulty escalation) 발생 시 삭감될 수 있는 감시자 예치금 $\$dw$ 입니다. 우리는 노드가 다른 노드의 제출 값을 (예컨대, 섰 5.3에서 논의한 커밋-공개 체제 같은 이유로) 복사할 수 없다고 가정합니다. 각 라운드에서 노드는 먼저 보고서를 커밋하고 모든 노드가 커밋되면 (또는 타임아웃이 만료되면) 보고서를 공개하게 됩니다.

생성될 각 보고서에 대해 모든 노드에는 1과 n 사이의 감시자 우선순위가 무작위로 부여되며, 1이 최우선순위가 됩니다. 이 우선순위는 하나의 감시자에게 보상을 집중시킵니다. 모든 보고서가 공개되면 경보 단계가 이어집니다. 일련의 n (동기식) 라운드에서 i 의 우선순위를 갖는 노드는 해당 i 라운드에서 경보할 기회를 얻습니다.

노드가 보고서를 공개한 후 메커니즘에서 가능한 결과를 생각해 볼 차례입니다. 다시 이진 보고서를 가져와 올바른 값은 참(true)이고 잘못된 값은 거짓(false)을 내보낸다고 가정하겠습니다. 또한, 1단계 메커니즘으로 대다수 노드의 값을 최종 보고서로 출력한다고 가정합니다.

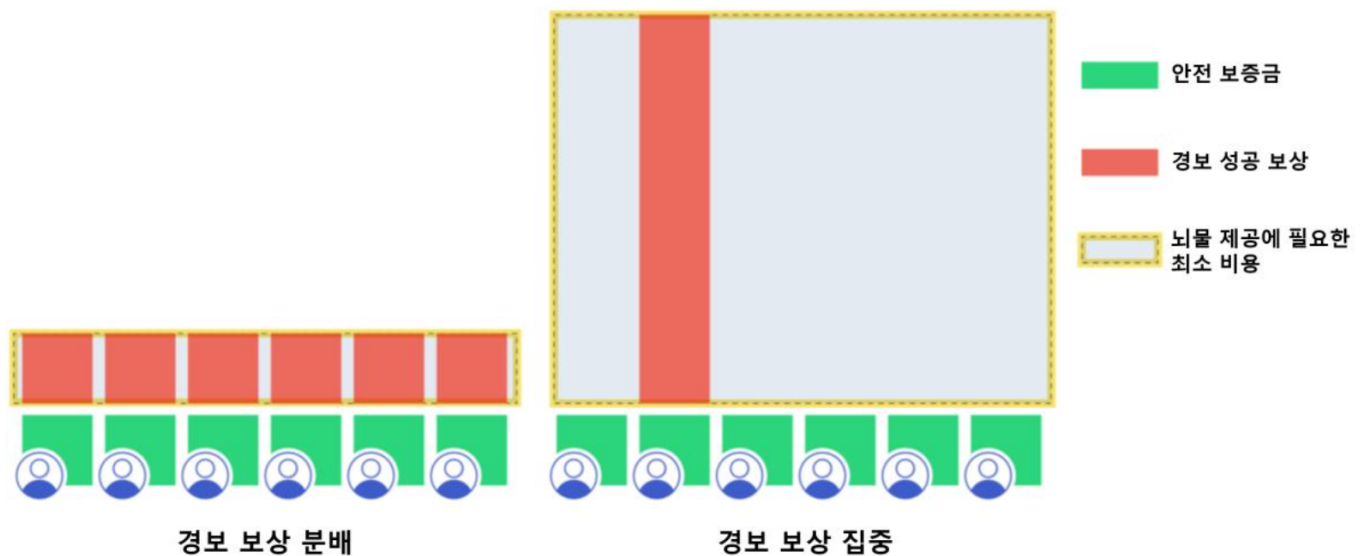
이 메커니즘에서 가능한 결과는 다음의 가지입니다.

- 완전한 일치 : 최선의 경우, 노드는 서로 완전히 동의하게 됩니다. 모든 노드는 정상적이며 동일한 값 r (참 또는 거짓)을 가진 보고서가 적시에 제공된 상태입니다. 이 경우 네트워크는 r 을 의존 컨트랙트로 전달하고 각 노드에 정해진 라운드당 지불금 $\$p$ ($\$d$ 보다 훨씬 낮음)를 보상하면 됩니다.

• 부분 일치 : 일부 노드가 오프라인 상태이거나 어떤 값이 올바른지 의견이 다를 수 있지만 대부분의 노드는 참을 보고하고 소수만 거짓을 보고합니다. 이 경우도 간단합니다. 대다수 값(참)이 연산되어 올바른 보고서 r 이 생성됩니다. r 을 보고한 모든 노드는 보상 $\$p$ 를 받지만, 거짓을 보고한 오라클은 그 예치금이 예를 들어, $\$10p$ 정도 삭감됩니다.

• 경보 : 감시자가 네트워크에서 출력한 값이 올바르지 않다고 판단할 경우, 공개적으로 경보를 울려서 메커니즘을 2단계 네트워크로 강화(escalation)시킵니다. 이때, 발생 가능한 결과는 다음과 같이 두 가지입니다.

- 올바른 경보 : 2단계의 네트워크가 1단계 네트워크의 출력이 잘못되었음을 확인하면 경보한 감시자 노드는 모든 삭감된 보증금으로 마련된 $\$dn/2$ 이상의 보상을 받게 됩니다.
- 잘못된 경보 : 1단계 및 2단계의 오라클이 서로 일치하는 값일 경우, 강화에 결함이 있는 것으로 간주하며 경보한 노드는 $\$dw$ 의 보증금을 잃게 됩니다.



- 그림 16: 집중된 경보 보상으로 증폭된 뇌물 제공 비용. 뇌물을 제공한 공격자는 경보로 얻을 수 있는 보상(빨간색 막대로 표시) 이상의 뇌물을 각 노드에 제공해야 합니다. 경보 보상이 공유될 때 이 보상은 상대적으로 작을 수 있습니다. 경보 보상 집중 시 단일 노드가 얻을 수 있는 보상은 증가합니다(높은 빨간색 막대). 결과적으로 뇌물 제공이 가능하기 위해 공격자가 지불해야 할 총금액은(회색영역) 경보 보상 분배 시보다 집중 시에 훨씬 더 커집니다.

보고서를 낙관적으로 수락하는 경우 감시 경보는 의존 컨트랙트의 실행에 어떠한 변화도 일으키지 않습니다. 2단계 위원회의 잠재적 중재를 기다리도록 설계된 컨트랙트의 경우 감시자는 컨트랙트 실행을 지연시킬 뿐 동결하지는 않습니다. 컨트랙트가 판단 기간 동안 페일오버 DON을 지정하는 것도 가능합니다.

9.4.2 제공 예치 효과

모든 노드가 감시자 역할을 할 수 있고, 집중된 보상을 보장하는 엄격한 노드 우선순위와 결합될 경우, 메커니즘은 섰 9.3.3에 설명된 뇌물 제공 공격자에 대한 제공 예치 효과를 달성할 수 있습니다. 다시 말해, 우리가 설정했던 조건에서 각각 $\$d$ 의 보증금이 있는 n 개 노드가 있는 네트워크의 경우, 뇌물 제공자(위의 모든 종류 중 하나)가 성공하려면 $\$dn^2/2$ 보다 큰 예산을 가져야 합니다.

엄밀히 말하면 뇌물 제공자는 n 개 노드 대다수를 손상해야 하므로, 최소 $(n + 1)/2$ 개 노드를 손상시켜야 합니다(n 이 홀수라고 가정). 감시자는 $\$d(n + 1)/2$ 의 보상을 받기 때문에 결과적으로 뇌물 제공자는 모든 노드에 이 금액을 지불하여 감시자 역할을 하지 않게 해야 합니다. 우리는 뇌물 제공자의 예산이 최대 $\$d(n^2 + n)/2$ 일 때, 뇌물 제공자와 오라클 간에 완벽한 균형 게임의 하위 게임에서(즉, 게임동안 어느 지점에서든 균형 상태 유지) 뇌물 제공자가 뇌물을 주지 못하게 하고 각 오라클이 참값을 보고하도록 하는 공식을 설명해 보았습니다.

우리는 성공적인 뇌물제공자가 어떻게 노드 보증금의 합계보다 훨씬 더 많은 예산을 요구할 수 있는 지를 앞서 논의한 바 있습니다. 직관적으로 설명하기 위해, 그림 16에서는 경보 보상 집중의 영향을 도식을 통해 나타냈습니다. 그림에서 볼 수 있듯이 감시자 경보에 대한 보상(즉, 거짓을 보고하는 뇌물 노드의 예금)이 모든 잠재적 경보자에게로 분배되면 개별 경보 노드가 예상할 수 있는 총금액은 $\$d$ 정도로 상대적으로 작게 될 것입니다. $\$d$ 보다 큰 금액을 지불할 가능성이 없다는 것을 알고 있는 뇌물 제공자는 거짓 결과 조건부 뇌물(false-outcome conditional bribe)을 사용하여 n 개의 노드 각각에 $\$d + E$ 보다 약간 더 많은 뇌물을 제공할 것입니다.

일반적으로 생각하는 것과 달리, 그림 16은 경보 신호를 보내는 노드에 보상을 광범위하게 분배하는 시스템이 단일 감시자에게 보상을 집중시키는 시스템보다 훨씬 취약하다는 것을 보여줍니다.

매개변수 예시: 노드 $n = 100$ (각각 $\$d = \$20K$ 의 보증금을 맡김)인 (1단계) 네트워크를 고려해 보겠습니다. 이 네트워크는 총 $\$2M$ 의 보증금을 맡고 있으며 $\$100M = \$dn^2/2$ 의 예산을 가진 뇌물 제공자로부터 보호되어야 합니다. 당연히 오라클의 수를 늘리는 것은 $\$d$ 를 증가시키는 것보다 더 효과적이며 극적인 효과를 가져올 수 있습니다. 즉, 노드 $n = 300$ 일 때, $\$d = \$20K$ 의 예금이 있는 네트워크는 최대 $\$900M$ 의 예산을 가진 뇌물 제공자로부터 안전합니다.

많은 경우, 예치 시스템은 책정된 뇌물 보호 수준보다 더 높은 가치의 스마트 계약을 보호할 수 있습니다. 대체로 계약을 공격하는 공격자가 전체 금액을 인출하기 어렵기 때문입니다. 예컨대, $\$1B$ 를 보유한 체인링크 기반의 계약은 $\$100M$ 을 보유한 뇌물 제공자에 대응할 안전 보증금 정도만 필요로 할 수 있습니다. 실제로 이러한 공격자는 해당 계약 가치의 10%만의 이익을 인출할 수 있기 때문입니다.

참고: 네트워크의 가치가 제공으로 증가할 수 있다는 발상은 유명한 메칼프의 법칙(Metcalfe's Law) [167, 235]으로 알려진 바 있습니다. 이 법칙은 네트워크의 가치가 연결된 개체의 수에 따라 제공으로 증가한다는 것입니다. 그러나 메칼프의 법칙은 잠재적인 쌍별 네트워크 연결 수의 증가, 즉 인센티브 메커니즘의 기본 제공 예치 영향과는 다른 현상에 따라 작용합니다.

9.4.3 2단계의 실현

다음 두 가지 운영 특성은 높은 신뢰성의 2단계 실현을 용이하게 합니다. (1) 2단계의 판결은 오라클 네트워크상에서 드물게 발생해야 하기 때문에 1단계의 정상 운영 비용보다 훨씬 더 많은 비용이 들 수 있습니다. (2) 낙관적으로 수락된 보고서(또는 그 실행을 위해 중재를 기다리는 컨트랙트)가 있다면 2단계는 실시간으로 실행할 필요가 없습니다. 이러한 특성 때문에 2단계가 특정 DON의 요구 사항을 충족할 수 있도록 다양한 구성 옵션이 구비됩니다.

일례로 2단계 위원회는 체인링크 네트워크에서 가장 오래 서비스를 제공하고 가장 안정적인 노드 중에서 DON(즉, 1단계)이 선택한 노드로 구성될 수 있습니다. 그러한 노드의 운영자는 상당한 관련 운영 경험뿐만 아니라, FFO에서 상당한 암묵적 인센티브를 가지고 있으며, 이는 체인링크 네트워크가 높은 신뢰성을 유지하도록 보장하려는 욕구를 자극합니다. 또한, 신뢰성에 대한 투명성을 제공하는 데 공개적으로 사용 가능한 실적 내역도 있습니다. 2단계 노드는 1단계 네트워크의 참여자일 필요는 없으며, 여러 1단계 네트워크에 걸쳐 결합 여부를 판정할 수 있습니다.

DON 노드는 DON의 2단계 위원회를 구성할 n' 개의 노드 집합을 미리 지정하고 공개적으로 커밋할 수 있습니다. 또한, 1단계 노드에 불이익을 주는 데 필요한 2단계 투표수를 결정하는 매개변수 $k' \leq n'$ 를 게시할 수도 있습니다. 보고서 관련 경보가 생성되면, 2단계의 구성원은 각 1단계 노드에서 제공한 값의 정확성에 대해 투표합니다. k' 개의 반대투표를 받은 1단계 노드는 감시자 노드에게 보증금을 몰수당합니다.

위에서 언급한 판결 자체의 드문 발생과 늘어난 실행 시간을 고려해, 1단계와 달리 2단계의 노드는 다음과 같이 처리됩니다.

1. 판결을 진행한 대가로 많은 보상을 받음.
2. 1단계에서 사용하는 다양한 데이터 트 외에 추가 데이터 소스를 활용함.
3. 매뉴얼 혹은 전문가의 검사나 개입이 있을 수 있음. 예를 들어, 소스 데이터의 오류를 식별 및 조정하고, 결함이 있는 데이터를 전달하는 정직한 노드와 부정행위를 하는 노드를 구별하기 위해 매뉴얼 및 전문가 검사 및 개입에 의존할 수 있음.

2단계 노드 선택과 결정을 관장하는 정책과 관련하여 위에서 설명한 접근 방식이 2단계의 수많은 가능성 중 일부에 불과하다는 점을 강조하고 싶습니다. 우리의 인센티브 메커니즘은 2단계가 실현되는 방식에 대해 완벽한 유연성을 제공합니다. 각 DON은 참여 노드 및 사용자의 특정 요구 사항과 기대치를 충족하는 2단계에 대한 규칙을 개별적으로 구성하고 설정할 수 있습니다.

DECO 및 타운크라이어 판결 도구: 우리 메커니즘 2단계는 의도적으로 잘못된 보고서를 생성하는 적대적 1단계 노드와 의도치 않게 소스에서 잘못된 데이터를 전달한 정직한 1단계 노드를 구별할 수 있어야 합니다. 그래야만 2단계가 삭감 조치로 우리 메커니즘의 목표인 부정행위 방지를 실현할 수 있습니다. DECO 및 타운크라이어는 2단계 노드가 이러한 중요한 구분을 안정적으로 수행할 수 있도록 하는 강력한 도구입니다.

2단계 노드는 때에 따라 1단계 노드에서 사용하는 데이터 소스를 직접 쿼리하거나 ADO 선 7.1을 사용하여 결함 있는 보고서가 결함 있는 데이터 소스에 기인한 것인지 확인할 수 있습니다. 그러나 상황에 따라 2단계 노드가 1단계 노드의 데이터 소스에 직접 액세스하지 못할 수도 있습니다. 이런 경우 올바른 판단이 불가능하게 되며, 주관적 판단에 의존해야 합니다. 이전의 오라클 분쟁 시스템은 이러한 문제를 해결하기 위해 비효율적인 확대 투표 방식에 의존했습니다.

DECO 또는 타운크라이어를 사용하면 1단계 노드가 2단계 노드에게 올바르게 동작했음을 증명할 수 있습니다. (두 시스템에 대한 자한 내용은 섹션 3.6.2를 참조합니다.) 특히, 2단계 노드에서 1단계 노드가 잘못된 보고서 값 $\tilde{r}$ 을 출력한 것으로 인식했을 경우, 1단계 노드는 DECO 또는 타운크라이어를 사용하여 2단계 노드에게 줄 변조 방지 증거(DON에 의해 권한 있는 것으로 인식된 소스(TLS 적용)에서 $\tilde{r}$ 을 올바르게 전달한 것이라는 내용)를 생성할 수 있습니다. 결정적으로 1단계 노드는 데이터 소스에 직접 액세스해야 하는 2단계 노드 없이도 이를 수행할 수 있습니다. 이에 따라 체인링크에서는 원하는 데이터 소스에 대한 올바른 판결이 가능해집니다.

9.4.4 허위 보고 보험

우리 예치 메커니즘의 강력한 뇌물 저항성은 기본적으로 삭감된 자금으로 경보자에게 보상하는 데 기인합니다. 금전적 보상이 없다면 경보자는 뇌물을 거부할 직접적인 동기가 없을 것입니다. 결과적으로, 잘못된 보고로 인해 피해를 본 사용자(예: 잘못된 가격 데이터가 스마트 컨트랙트에 전달되어 돈을 잃은 사용자)에게 보상하기 위해 삭감된 자금을 사용할 수는 없습니다.

가정에 따르면, 잘못된 보고는 잠재적 판결, 즉 2단계의 조치 후 컨트랙트가 보고서를 수락하면 발생하지 않습니다. 그러나 위에서 설명한 것처럼 가능한 효율적으로 운영하기 위해 컨트랙트는 메커니즘이 올바른 보고를 시행하리라 믿고 여기에 낙관적으로 의존할 수 있습니다. 즉, 2단계 판결 전에 보고서를 수락할 수 있습니다. 실제로 우리가 가정한 모델에서, 낙관적 행동은 합리적인 공격자의 예산이 메커니즘의 예치 영향력을 초과하지 않는 한 안전합니다.

예컨대, 가능성은 낮지만, 압도적 재정 자원을 보유한 공격자에 의해 발생할지도 모를 메커니즘 실패를 우려하는 사용자는 허위 보고 보험(misreporting insurance)과 같은 경제적 보안 차원의 추가 레이어를 원할 수 있습니다. 이미 DAO(예: [7]) 등의 혁신적인 메커니즘을 통해 가까운 미래에 체인링크 보안 프로토콜 용으로 이러한 종류의 스마트 컨트랙트 기반 상품을 제공하고자 하는 여러 보험사가 있습니다. 체인링크 노드로서의 실적 내역 및 예치금과 같은 노드의 각종 데이터는 위험에 대한 보험 통계적 평가에 쓸 매우 강력한 기반이 되어, 보험 계약자에게는 저렴하고 보험사에게는 지속 가능한 방식으로 정책 가격을 책정할 수 있도록 합니다.

기본적인 허위 보고 보험은 스마트 컨트랙트를 사용하여 효율적이면서도 신뢰할 수 있는 방식으로 구현될 수 있습니다. 간단한 예를 들면, 인센티브 메커니즘의 2단계가 1단계에서 생성된 보고서의 오류를 식별한 경우, 매개변수 보험 컨트랙트 $SCins$ 가 보험 계약자에게 자동으로 보험금을 지급할 수 있습니다. 예를 들어, 보험에 가입하려는 사용자 U (대상 컨트랙트 SC 의 생성자)는 탈중앙화된 보험사에 컨트랙트 관련 보험금 $\$M$ 을 지급하는 보험 가입 요청을 제출할 수 있습니다. 보험사는 U 를 승인하고 $SCins$ 에서 정기적인(예: 월간) 보험료 $\$P$ 를 설정할 수 있습니다. U 가 보험료를 지불하는 동안 해당 사용자의 보험은 활성 상태로 유지됩니다.

SC 에서 보고 실패가 발생하면, SC 에 대한 충돌 보고서 쌍(r_1, r_2)이 만들어집니다. 여기서 r_1 은 메커니즘의 1단계에서 서명되고 r_2 (해당되는 올바른 값의 보고서)는 2단계에서 서명됩니다. U 가 $SCins$ 에 유효한 쌍(r_1, r_2)을 제공할 수 있다면, 해당 컨트랙트는 자동으로 사용자에게 $\$M$ 을 지불하게 됩니다. 물론, 사용자는 최근까지 보험료를 성실하게 지불한 상태여야 합니다.

9.5 단일 라운드 변형

이전 하위 선에서 설명한 프로토콜에서는 2단계 위원회가 감시자가 경보를 발생시켰는지 여부를 결정하기 위해 n 라운드를 기다리도록 요구합니다. 이 요구 사항은 낙관적인 경우, 즉 1단계가 올바르게 작동하는 경우에도 동일하게 유지됩니다. 하지만 보고서를 낙관적으로 수락하지 않으려는 사용자(잠재적인 판결 이전에 수락하지 않는 경우)를 해당 방식으로 지연시키는 것은 불가능합니다.

이러한 이유로 우리는 단일 라운드만 요구하는 대체 프로토콜도 모색하고 있습니다. 이 접근 방식에서 모든 오라클 노드는 경보 발령을 원하는지에 대한 여부를 나타내는 비밀 비트(secret bits)를 제출합니다. 2단계 위원회는 이러한 값을 우선순위에 따라 확인합니다. 대략적으로 설명하면, 이러한 체제는 다음 단계를 포함할 수 있습니다.

1. **감시자 비트 제출** : 각 노드 O_i 는 생성하는 모든 보고서에 대해 1비트의 감시자 값 $w_i \in \{no_alert, alert\}$ 를 2단계의 노드에게 비밀로 공유합니다.
2. **익명의 팁** : 모든 오라클 노드는 감시자 비트가 제출된 동일 라운드에서 익명의 팁 α 를 2단계 위원회에 제출할 수 있습니다. 이 팁 α 는 현재 보고서에 대해 경보가 발생했음을 나타내는 메시지입니다.
3. **감시자 비트 검사** : 2단계 위원회는 오라클 노드의 감시자 비트를 우선순위에 따라 공개합니다. 노드는 경보하지 않을 때 no_alert 감시자 비트를 보냈을 것입니다. 그렇지 않았어도 트래픽 분석으로 모든 노드의 비트를 확인할 수 있습니다. 프로토콜은 최우선순위인 경보 감시자에 앞서 우선순위가 높은 노드의 no_alert 감시자 비트를 먼저 공개합니다.

공개된 내용이 n -라운드의 프로토콜과 동일하다고 관찰되면, 보상은 해당 체계에 따라 분배됩니다. 즉, 첫 번째로 식별된 감시자는 잘못된 보고서를 제출한 노드의 삭감된 보증금을 받게 됩니다.

익명의 팁을 사용하면 경보가 발생하지 않았을 때, 2단계 위원회가 비대화형(non-interactive) 상태를 유지할 수 있어 일반적인 통신 복잡성이 줄게 됩니다. 경보를 울리는 모든 감시자는 익명의 팁을 제출할 경제적 인센티브가 있습니다. 팁을 제출하지 않으면 모든 노드에 보상이 지급되지 않습니다.

공격자가 익명의 팁 α 의 발신자 O_i 를 네트워크 데이터를 기반으로 식별하는 것을 방지하기 위해, 익명의 팁은 익명 채널(예: Tor를 통해 전송되거나, 더 현실적으로는 클라우드 서비스 제공업체를 통해 프록시 됨)을 통해 전송될 수 있습니다. 팁이 O 에서 시작되었음을 입증하기 위해 O_i 는 링 서명(ring signature)을 사용하여 α 에 서명할 수 있습니다[39, 192]. 또는, 악의적인 오라클 노드가 2단계 위원회에 출처를 알 수 없는 서비스 거부 공격(DoS)을 하는 것을 방지하기 위해, α 는 취소 가능한 익명성을 가진 익명의 자격 증명일 수도 있습니다[73].

실제로 이 프로토콜을 구현할 수 있지만, 다소 무거운 엔지니어링 요건을 충족해야 합니다. (이를 줄이는 방법을 모색 중임). 예를 들어 1단계 노드는 2단계 노드와 직접 통신해야 하므로 디렉토리를 유지 관리해야 합니다. 익명 채널과 링 서명으로 체계의 엔지니어링 복잡성이 가중되기도 합니다. 마지막으로 아래 비교에서 간략하게 설명된 특별 신뢰 요건도 있습니다. 우리는 초선형 예치 효과를 달성하면서도 아마도 제공보다 작은 더 간단한 체제를 모색 중입니다. 이 경우 뇌물 제공자는 점근적으로 최소한 $\$n \log n$ 의 리소스를 필요로 합니다. 현재 고려 중인 체계로는 감시자 역할을 할 노드의 하위 집합을 무작위로 선택하는 방안도 있는데, 잠재적인 뇌물로 특히 강력한 공격이 이루어집니다.

비고: 이 단일 라운드 예치 메커니즘의 보안을 위해서는 오라클과 2단계 노드 사이에 접근 불가능한 채널(untappable channels)이 필요합니다. 이는 투표[82, 138]와 같은 강제 저항 시스템의 표준 요건이기도 하며 실제 활용에 있어서도 실용적입니다.

추가로, 뇌물 제공자와 협력하려는 노드 O_i 는 뇌물 제공자에게 특정 값을 암호화했음을 보여주는 방식으로 비밀 공유를 구성할 수 있습니다. 예컨대, O_i 가 뇌물 제공자가 통제하는 노드를 모를 경우, O_i 는 모든 위원회 구성원에게 0 값을 제출합니다. 뇌물 제공자는 O_i 의 준수 여부를 확률적으로 확인할 수 있습니다. 모든 단일 라운드 프로토콜에서 이 문제를 방지하려면 O_i 가 최소한 하나 이상의 정직한 2단계 노드의 ID를 알아야 합니다.

각 2단계 노드가 공유 값에 랜덤화를 추가하게 되는 대화형 프로토콜을 사용하면 뇌물 제공자가 할 수 있는 최선은 랜덤한 감시자 비트 중 O_i 에 의한 선택을 강제하는 것입니다.

9.6 암묵적 인센티브 프레임워크(IIF)

체인링크 네트워크에서 FFO는 올바른 행동에 대한 암묵적 인센티브의 한 형태입니다. 네트워크에 대한 경제적 보안을 강화하는 데 도움이 된다는 점에서 명시적 예치, 즉 보증금과 같은 기능을 합니다. 따라서 FFO는 네트워크에서 노드(유효한) 보증금 $\$d$ 일부로 포함해야 합니다.

그렇다면 체인링크 네트워크 내에서 FFO 및 기타 형태의 암묵적 인센티브를 어떻게 측정해야 할까요? 암묵적 인센티브 프레임워크(IIF)는 이러한 목적을 위해 개발한 일련의 원리와 기법입니다. IIF 작동 방식에 대한 우리의 비전은 블록체인 시스템이 제공하는 다양한 형태의 전례 없는 투명성과 블록체인 시스템이 생성하는 노드 실적에 대한 높은 신뢰도의 내역을 토대로 합니다. 이제부터는 IIF의 핵심 요소에 대한 아이디어를 매우 간략하게 살펴보겠습니다.

IIF 자체는 분석 알고리즘이 사용할 수 있도록 관련 데이터를 높은 보증과 함께 제공하는 메커니즘과 함께 암묵적 인센티브를 평가할 때 중요하다고 여겨지는 일련의 요소로 구성됩니다. 각각의 체인링크 사용자는 IIF를 여러 방식으로 사용하기를 원할 수 있습니다(예: 다른 요소에 다른 가중치를 부여할 경우). 우리는 커뮤니티에서 사용자의 개별 위험 평가 선호도에 따라 IIF 적용을 지원하는 분석 서비스가 생겨나기를 기대합니다. 이에 따라 아래에서 논의한 바와 같이 높은 보증 및 시기적절한 지원 데이터에 대한 액세스를 보장함으로써 이러한 서비스를 촉진하고자 합니다(선 9.6.4).

9.6.1 미래 수수료 기회

노드는 체인링크 생태계에 참여하여 일반 데이터 피드부터 탈중앙 ID, 공정한 시퀀싱 및 기밀 보존 디파이와 같은 고급 서비스에 이르는 다양한 서비스에 대해 네트워크가 지불하는 수수료 일부를 받습니다. 체인링크 네트워크의 수수료는 서버 실행, 데이터 라이선스 획득, 높은 가동 시간을 보장하기 위한 글로벌 인력을 채용 등 노드 운영자의 비용을 지원합니다. 미래 수수료 기회(Future Fee Opportunity, FFO)는 노드가 미래에 얻을 수 있는 순 비용인 서비스 요금을 의미하며 노드가 얻을 수 있거나, 결함 있는 동작을 한 경우 잃기도 합니다. FFO는 네트워크를 보호하는 데 도움이 되는 예치금의 한 형태입니다.

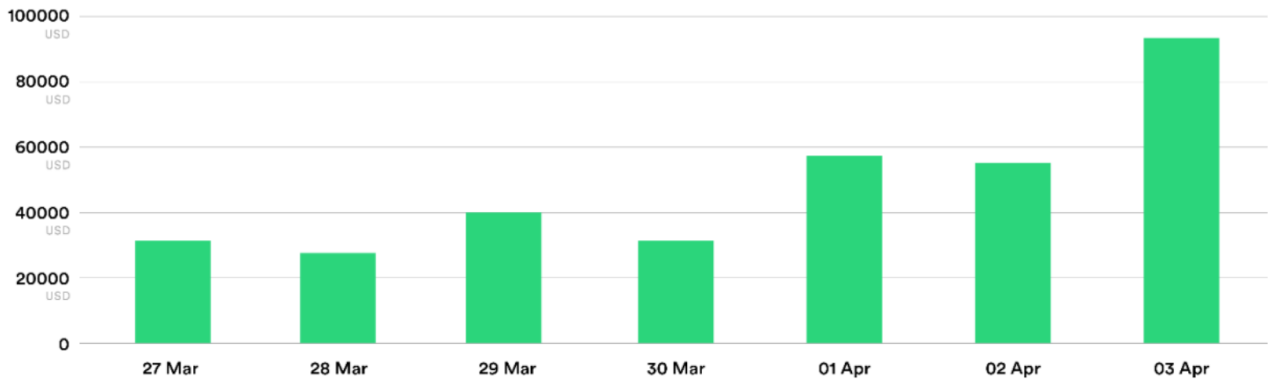
FFO의 유용한 기능 중 하나는 온체인 데이터(오프체인 데이터로 보완됨)가 노드 내역에 높은 신뢰 기록을 만들어, 투명하고 실증적인 방식으로 FFO를 연산할 수 있다는 점입니다.

FFO의 간단한 1차 측정은 일정 기간 동안 노드의 평균 순수익(즉, 총 수익에서 운영 비용을 뺀 값)으로 도출할 수 있습니다. 예를 들어, FFO는 누적 미래 순수익의 순 현재 가치 [114], 즉 모든 미래 수입의 시간 할인 가치(time-discounted value)로 계산될 수 있습니다.

그러나 노드 수익은 그림 17과 같이 바뀔 수 있습니다.

더 중요한 것은 노드 수익이 시간 흐름에 고정적인 분포로 발생하지 않을 수 있다는 것입니다. 결과적으로 FFO를 추정할 때 탐색해야 할 다른 요소는 다음과 같습니다.

• 실적 내역 : 보고서의 정확성과 적시성, 가동 시간을 포함한 운영자의 실적 내역은 사용자가 신뢰성을 평가할 객관적인 기초를 제공합니다. 따라서 실적 내역은 사용자의 오라클 노드 선택(또는 DON의 출현으로 DON 선택)에 중요한 요소를 제공합니다. 탄탄한 실적 내역은 높은 지속적인 수익으로 이어질 수 있습니다.



- 그림 17: 2021년 3월 대표 주간(1주) 동안 단일 데이터 피드(ETH-USD)로 체인링크 노드가 얻은 수익.
- 데이터 액세스 : 오라클은 개방형 API에서 다양한 형태의 데이터를 얻을 수 있지만 특정 형태의 데이터나 특정 고품질 소스는 구독 기반 또는 합의된 계약을 통해서만 사용할 수 있습니다. 접근 권한이 필요한 특정 데이터 소스는 안정적인 수익원을 창출하는 역할을 할 수 있습니다.
- DON 참여 : 노드 커뮤니티는 DON의 등장에 따라, 함께 모여 특정 서비스를 제공하게 될 것입니다. 많은 DON이 선택에 따라 운영자를 둘 것이며, 평판이 좋은 DON이 참여하여 일관된 수익원을 보장하는 데 도움이 되는 특권적인 시장 위치를 구축하게 될 것으로 기대합니다.
- 플랫폼 간 활동 : 일부 노드 운영자는 비블록체인 상에서 PoS 검증자, 데이터 제공자와 같이 다른 상황에서 잘 확립된 존재감이나 실적 추적 내역을 가졌을 수 있습니다. 이러한 다른 시스템에서의 실적(데이터가 신뢰할 수 있는 형식으로 제공될 경우)은 실적 이력 평가에 정보를 제공할 수 있습니다. 마찬가지로, 체인링크 네트워크에서의 결함 있는 행동은 사용자를 몰아냄으로써 다른 시스템에서의 수익도 위태롭게 할 수 있습니다. 즉, FFO는 플랫폼 전반에 걸쳐 확장될 수 있습니다.

9.6.2 투자성 FFO

노드 운영자는 체인링크 네트워크에 참여해 운영으로 수익을 얻을 뿐만 아니라, 작업을 실행할 수 있는 새로운 기회를 창출할 목적으로 자신을 만들고 포지셔닝할 수도 있습니다. 다시 말해, 네트워크상에서의 오라클 노드의 소비는 디파이 및 기타 스마트 컨트랙트 애플리케이션 도메인의 미래와 오라클 네트워크의 새로운 비블록체인 애플리케이션에 대한 긍정적인 약속이 될 것입니다. 오늘날 노드 운영자는 기존 체인링크 네트워크에서 사용할 수 있는 수수료를 받는 동시에, 미래 네트워크에서 사용할 수 있는 수수료(정직한 행동일 경우)를 얻는 데 유리한 평판, 실적 내역 및 운영 전문 지식을 쌓아가고 있습니다. 현재 체인링크 생태계에서 운영되는 노드는 추가적인 체인링크 서비스가 도입됨에 따라 신규 사용자보다 더 유리한 위치에서 수수료를 벌게

될 것입니다. 이러한 이점은 신규 사업자뿐만 아니라, 기존의 유명 테크놀로지 회사에게도 적용될 수 있습니다. 예를 들어, 전통적인 기술 제공업체인 T-Systems(Deutsche Telekom의 자회사)와 대규모 중앙 집중식 거래소인 Kraken은 체인링크 생태계에서 이미 초기 입지를 구축했습니다[28, 143].

이러한 오라클 노드의 미래 기회에 대한 참여는 그 자체가 일종의 투자 성격의 FFO로 간주될 수 있으며, 체인링크 네트워크의 예치금의 한 형태로도 볼 수 있겠습니다.

9.6.3 외부 평판

이미 설명한 바와 같이, IIF는 운영자가 완전히 익명인 네트워크에서 운영될 수 있습니다. 즉, 관련된 사람 또는 현실 계의 단체 등을 공개하지 않고도 운영할 수 있습니다. 그러나 사용자가 공급자를 선택하는 데 외부의 평판은 잠재적으로 중요하게 작용합니다. 외부 평판이란, 가명이 아닌 실제 신원으로 비롯되는 신뢰성에 대한 인식을 의미합니다. 실제 신원에 대한 평판 위험은 일종의 암묵적 인센티브로 볼 수 있습니다. 우리는 IIF의 관점에서 FFO 추정에 포함될 수 있는 플랫폼 간 활동 수립 수단까지 고려해, 즉 암호경제학적 의미에서 평판에 대해 생각해 보았습니다.

가명 연결과 달리, FFO 추정의 한 요소로 외부 평판을 사용할 경우의 이점은 외부 평판이 실적을 운영자의 기존 활동뿐만 아니라 미래 활동과도 연결한다는 것입니다. 예를 들어, 개인에게 나쁜 평판이 생기면 그 사람이 소속된 기업의 평판에까지 영향을 끼칠 수 있습니다. 다시 말해, 외부 평판은 가명화된 실적 내역보다 더 넓은 범위의 FFO를 포착할 수 있습니다. 개인이나 기존 회사에 따라다니는 부정행위의 영향은 가명을 쓸 때보다 회피하기 어렵기 때문입니다.

체인링크는 IIF에서 외부 평판 사용을 지원하는 탈중앙 ID 기술(4.3절)과 호환됩니다. 이러한 기술은 검증을 통해 운영자가 주장하는 실제 ID의 정확성을 보장하는 데 도움이 될 수 있습니다.

9.6.4 개방형 IIF 분석

앞서 언급한 바와 같이, IIF는 암묵적 인센티브 분석을 위한 신뢰할 수 있는 오픈 소스 데이터와 도구 제공을 목표로 합니다. 그 목적은 커뮤니티 내의 제공자가 체인링크 사용자 베이스에서 필요한 다양한 위험 평가에 맞는 분석을 개발할 수 있도록 하는 것입니다.

노드 수익 및 실적에 관한 상당한 양의 과거 데이터는 높은 신뢰도와 불변의 형태로 체인에 상주 중입니다. 그러나 우리 목표는 오프체인 보고(Off-Chain Reporting, OCR)이나 DON 활동 등 오프체인에서만 볼 수 있는 행동 데이터까지 포함하여 가능한 포괄적인 데이터를 제공하는 것입니다. 이러한 데이터는 잠재적으로 방대할 수 있습니다. 데이터를 저장하고 무결성을 보장하는 가장 좋은 방법, 즉 변조로부터 보호하는 방법은 섰 3.3에서 논의된 기술을 사용하여 DON의 도움을 받는 것입니다.

일부 인센티브는 예치금 및 기본적인 FFO와 같은 직접적인 측정에 적합합니다. 투자성 FFO 및 평판과 같은 다른 요소는 객관적으로 측정하기 어렵지만, 우리는 체인링크 생태계의 역사적 성장, 평판의 소셜 미디어 지표 등을 포함한 데이터를 지원하는 것으로, 정량화하기 어려운 요소에도 적용할 수 있는 IIF 분석 모델을 지원할 수 있다고 믿습니다.

검증된 ID 정보와 같이 IIF에서 사용되는 기타 데이터뿐만 아니라 노드의 오프체인 실적 내역과 관련된 데이터를 모니터링, 검증, 기록하기 위해서는 특수한 전용 DON이 발생해야 할 것이라고 상상할 수 있을 겁니다. 이러한 DON은 체인링크 커뮤니티에 서비스를 공급하는 모든 분석 제공업체에 대해 균일하고 신뢰할 수 있는 IIF 데이터를 지원할 수 있습니다. 또한, 분석 제공업체의 주장을 커뮤니티에서 독립적으로 검증할 수 있도록 하는 이른바 골든 레코드(golden record)를 제공할 수도 있을 것입니다.

9.7 종합적 논의: 노드 운영자 인센티브

노드 운영자에 대한 명시적 및 암묵적 인센티브에 대한 위의 논의를 종합하면 노드 운영자가 체인링크 네트워크에 참여하고 혜택을 받는 방식에 대해 전체적인 관점을 얻을 수 있습니다.

한편, 개념적 지침으로서 주어진 체인링크 노드 운영자 $\$S$ 에 의해 예치된 총자산을 다음과 같은 대략적이고 형식화된 식으로 표현할 수 있습니다.

$$\$S \approx \$D + \$F + \$FS + \$R,$$

여기서,

- $\$D$: 운영자가 참여하는 모든 네트워크에 걸쳐 명시적으로 예치된 모든 예치금의 합
- $\$F$: 운영자가 참여하는 모든 네트워크에 걸친 모든 FFO 합계의 순 현재 가치
- $\$FS$: 운영자의 투자성 FFO의 순 현재 가치
- $\$R$: 오라클 노드에서 식별된 부정행위로 인해 위험에 처할 수 있는 체인링크 생태계 외부 운영자의 평판 자산

대체로 개념적이지만, 이러한 대략적인 균형은 체인링크 노드의 높은 신뢰성 성능을 선호하는 다양한 경제적 요인이 존재한다는 사실을 보여줍니다. $\$D$ 이외의 요소는 현재 체인링크 네트워크에도 존재합니다.

9.8 경제적 보안의 선순환

IIF에서 미래 수수료 기회(FFO)로 수수료 지불을 나타내는 초선형 예치 영향의 조합은 오라클 네트워크에서의 경제적 보안 선순환을 유도할 수 있습니다. 이러한 선순환은 일종의 규모의 경제로 볼 수 있습니다. 특정 네트워크가 확보한 총량이 증가하면, 고정된 경제적 보안을 증가시키기 위해 지불해야 하는 추가적 지분은 사용자당 평균 비용과 마찬가지로 감소합니다. 따라서 사용자가 기존 네트워크에 가입하는 것이 새로운 네트워크를 생성하여 네트워크 경제적 보안을 동일한 수준으로 높이는 것보다 수수료 측면에서 더 저렴한 방안이 됩니다. 중요한 것은 새 사용자가 추가될 때마다 해당 네트워크의 모든 기존 사용자에게 서비스 비용 또한 낮아진다는 점입니다.

특정 수수료 구조(예: 예치된 금액에 대한 특정 수익률)에서 네트워크 증가로 벌어들이는 총수수료가 증가하면, 이는 네트워크 내 비중을 더 많이 확보하기 위한 추가 예치를 촉진시키게 됩니다. 특히, 개별 노드가 시스템에서 보유할 수 있는 총예치금에 상한이 있으면, 새로운 수수료 지불 정책이 시스템에 적용되어 FFO가 인상되면 노드의 수 n 도 증가하게 될 것입니다. 인센티브 시스템 설계의 초선형 예치 영향 덕분에 시스템의 경제적 보안은 n 보다 더 빠르게 증가하게 됩니다(예: 9.4절에서 다룬 메커니즘의 경우에는 n^2 였음). 결과적으로 경제적 보안에 필요한 평균 비용, 즉 1달러의 경제적 보안에 기여하는 예치금은 줄어들게 될 것입니다. 따

라서 네트워크는 사용자에게 더 낮은 수수료를 부과할 수 있습니다. 오라클 서비스에 대한 수요가 탄력적이라 가정하면(간단한 설명은 [31] 참조), 결국 수요 증가로 이어져 추가 수수료와 FFO가 발생할 것입니다.

예시 5. 인센티브 체계를 갖춘 오라클 네트워크의 경제적 보안은 예치금 $\$dn$ 에 대해 $\$dn^2$ 가 되고 예치금 1달러의 경제적 보안에 대한 기여는 n 이 되기 때문에 경제적 보안 1달러당 평균 비용(즉, 경제적 보안 1달러의 기여를 제공하는 예치 금액)은 $1/n$ 이 됩니다.

경제적 인센티브가 노드 당 $\$d \leq \$10K$ 의 상한을 가지며, 전적으로 FFO로만 구성되는 네트워크를 가정해 보겠습니다. 네트워크에 있는 노드의 개수 $n = 3$ 이라고 가정합니다. 그러면 경제적 보안의 1달러당 평균 비용은 약 $\$0.33$ 이 됩니다.

만약 네트워크의 총 FFO가 $\$30,000$ 이상(예: $\$31,000$)으로 상승하고, 노드당 FFO의 상한선이 적용되면 네트워크는 (최소한) $n = 4$ 로 커집니다. 최종적으로 경제 보안의 1달러당 평균 비용은 약 $\$0.25$ 로 떨어집니다.

그림 18에서는 오라클 네트워크의 경제적 보안에 관한 전체적인 선순환 개념을 보여줍니다.

경제적 보안의 선순환은 사용자가 수수료를 공동으로 모은 결과입니다. 집단적 FFO는 더 큰 네트워크 크기와 더 큰 집단 보안을 위해 작동합니다. 또한, 경제적 보안의 선순환이 DON의 재정적 지속성을 달성하는 데 유리하게 작용한다는 점에 주목해야 합니다. 사용자 요구를 충족시키기 위해 DON이 일단 생성된 뒤에는 수수료 수익이 오라클 노드의 운영 비용을 초과하는 지점까지 성장해야 합니다.

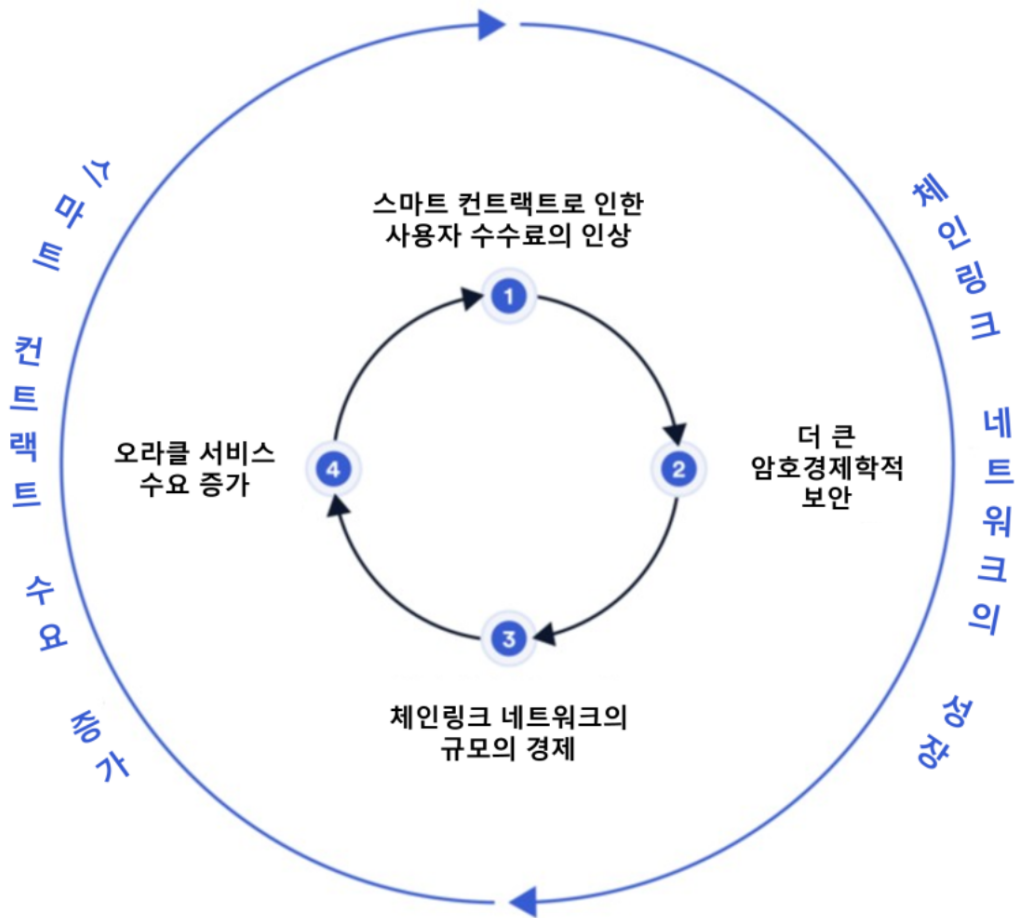


그림 18: 체인링크 예치의 선순환 구조. 오라클 네트워크에 대한 사용자 수수료 지불의 증가는 ① 네트워크의 성장과 그 경제적 보안의 성장으로 이어지게 됩니다. ② 이러한 초선형 성장은 체인링크 네트워크에서 규모의 경제를 실현합니다. ③ 구체적으로는, 수수료 지불이나 기타 예치금 증가로 발생하는 1달러당 경제적 보안, 즉 경제적 보안의 평균 비용이 감소하는 것을 의미합니다. ④ 더낮아진 비용 부담은 사용자의 오라클 서비스에 대한 수요 증가를 자극합니다.

9.9 네트워크 성장을 견인하는 추가 요인

체인링크 생태계가 계속 확장됨에 따라 사용자를 더 많이 끌어들이면서 블록체인 경제 인프라로서의 중요성은 급속히 증가하게 될 것으로 보입니다. 오라클 네트워크가 제공하는 가치는 초선형적입니다. 즉, 네트워크 자체의 크기보다 빠르게 성장합니다. 이러한 가치 성장은 규모의 경제(서비스 규모가 증가하면서 사용자당 비용 효율성이 높아짐)와 네트워크 효과(사용자가 DON을 더 널리 채택하면서 네트워크 용도가 다양해짐)에서 비롯됩니다.

기존 스마트 컨트랙트가 계속해서 더 많은 가치를 확보하고, 완전히 새로운 스마트 컨트랙트 애플리케이션이 더 탈중앙화된 서비스로 가능하게 되면 DON에 지불되는 총사용료 및 수수료는 증가합니다. 수수료 풀의 증가는 더 많은 탈중앙화 서비스를 생성하기 위한 수단과 인센티브로 전환되어 선순환으로 이어집니다. 이 선순환은 하이브리드 스마트 컨트랙트 생태계에서 중요한 닭과 달걀의 문제를 해결합니다. 혁신적인 스마트 컨트랙트 기능은 아직 존재하지 않지만(예: 새로운 디파이 시장은 종종 새로운 데이터 피드를 요구함), 이러한 기

능을 위해서는 충분한 경제적 수요로 뒷받침되는 탈중앙화 서비스가 필요합니다. 기존 DON에서 다양한 스마트 컨트랙트에 의한 수수료 풀링(pooling)은 증가하는 사용자 기반에 따른 추가적인 탈중앙화 서비스의 수요를 알리고, DON을 통한 해당 서비스 창출과 새롭고 다양한 하이브리드 스마트 컨트랙트의 지속적인 활성화로 이어지게 될 것입니다.

요약하면, 우리는 체인링크 예치 메커니즘의 선순환으로 네트워크가 안정될수록, 온체인 경제에 탈중앙화 서비스를 공급할 수 있는 체인링크 네트워크의 더 큰 성장으로 이어진다고 믿습니다.

10. 결론

우리는 이 백서를 통해 체인링크의 진화에 대한 비전을 제시했습니다. 이 비전의 주요한 주제는 단순한 데이터 전달보다 훨씬 광범위한 스마트 컨트랙트 서비스를 제공할 수 있는 오라클 네트워크의 역량입니다. 체인링크는 DON을 미래의 탈중앙화 서비스를 위한 기반으로 사용하여 성능 및 기밀성이 강화된 오라클 기능을 제공하는 것을 목표로 할 것입니다. 오라클 네트워크는 예치, 신중하게 고안된 가드레일, 기반 메인 체인에 대한 서비스 수준의 시행과 같은 원칙적인 암호 경제 메커니즘의 조합을 통해 강력한 신뢰 최소화를 제공할 것입니다. 또한, DON은 레이어-2 시스템이 트랜잭션에 대한 유연하고 공정한 순서화 정책을 시행하고 메모리 라우팅 트랜잭션에 대한 수수료 비용을 줄이는 데 도움을 줄 것입니다. 종합하면 이러한 기능은 모두 안전하고 풍부한 기능을 제공하는 하이브리드형 스마트 컨트랙트를 지향합니다.

DON 유연성은 기존 체인링크 서비스를 향상시키고 많은 추가 스마트 컨트랙트 기능 및 애플리케이션을 발생시킵니다. 여기에는 다양한 오프체인 시스템에 대한 원활한 연결, 기존 데이터에서 탈중앙화된 ID 생성, 인프라에 중요한 트랜잭션의 적시 제공을 보장하는 우선 채널, 기밀성 보존 디파이 도구 등이 있습니다.

우리가 여기에 제시한 비전은 야심찬 것입니다. 단기적으로 하이브리드 컨트랙트가 오늘날 스마트 컨트랙트의 범위를 넘어서는 목표를 달성할 수 있도록 권한을 부여하는 한편, 장기적으로 탈중앙화 메타레이어를 실현하는 것을 목표로 합니다. 다행히 우리는 컨센서스 알고리즘에서 영지식 증명 시스템에 이르기까지, 커뮤니티에서 빠르게 발전하며 연구된 새로운 도구와 아이디어를 활용할 수 있습니다. 따라서, 체인링크 사용자 커뮤니티의 기대에 부응하여 이 백서의 아이디어 구현을 우선할 것입니다. 우리는 전 계의 연결을 통해 스마트 컨트랙트를 강화하고 차대 금융 및 법률 시스템의 중추로서 탈중앙화 기술을 확립하기 위한 다음 단계를 위해 노력하고 있습니다.

감사의 글

본 백서의 수치 자료에 대한 렌더링 작업에 애써 주신 줄리안 알테리니와 손 리에게 감사의 말씀을 전합니다.