

엑셀라(Axelar)

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.

엑셀라 네트워크: 블록체인 생태계를 연결하는 애플리케이션

Draft 1.0
2021년 1월

초록

여러 블록체인 생태계가 등장하여 사용자와 애플리케이션 개발자에게 매력적인 독특하고 차별화된 기능을 제공하고 있습니다. 그러나 이러한 생태계 간의 소통은 매우 드물고 단편적입니다. 블록체인 생태계 간 애플리케이션이 마찰 없이 소통할 수 있도록 하기 위해, 우리는 엑셀라를 제안합니다. 엑셀라 스택은 간단한 크로스체인 통신을 가능하게 하는 탈중앙화 네트워크, 프로토콜, 도구 및 API를 제공합니다. 엑셀라 프로토콜 제품군은 크로스보더 라우팅 및 전송 프로토콜로 구성됩니다. 네트워크는 검증자들의 탈중앙화된 공개 네트워크에 의해 구동되며, 누구든 참여하고, 사용하며, 기여할 수 있습니다. 비잔틴 합의, 암호학, 인센티브 메커니즘은 크로스체인 요청에 고유한 높은 안전성과 라이브니스 요구사항을 충족하도록 설계되었습니다.

서론

블록체인 시스템은 빠르게 인기를 얻고 있으며 자산 토큰화, 탈중앙화 금융, 그리고 기타 분산형 애플리케이션의 새로운 활용 사례를 끌어들이고 있습니다. Ethereum, Monero, EOS, Cardano, Terra, Cosmos, Avalanche, Algorand, Near, Celo, Polkadot과 같은 주요 플랫폼들은 각각 독자적인 기능과 개발 환경을 제공하여 다양한 애플리케이션, 사용 사례 및 최종 사용자에게 매력적으로 다가갑니다. 그

러나 현재 각 플랫폼의 유용한 기능은 해당 플랫폼의 네이티브 토큰 보유자, 즉 생태계 사용자 중 1% 미만에게만 제공되고 있습니다. 플랫폼 개발자가 자신의 블록체인을 다른 생태계에 쉽게 연결할 수 있게 할 수 있을까요? 애플리케이션 개발자가 자신에게 가장 적합한 플랫폼에서 구축하면서도 여러 블록체인 생태계 간에 소통할 수 있게 할 수 있을까요? 사용자가 자신의 지갑에서 직접 모든 블록체인 애플리케이션과 상호작용할 수 있도록 할 수 있을까요?

블록체인 생태계를 연결하고 애플리케이션이 마찰 없이 상호작용할 수 있도록 하기 위해 우리는 엑셀라 네트워크를 제안합니다. 검증자들은 집합적으로 비잔틴 합의 프로토콜을 실행하며, 크로스체인 요청을 지원하는 프로토콜을 운영합니다. 누구든 네트워크에 참여하고 사용하며 기여할 수 있습니다. 기본 네트워크는 크로스체인 요청에 고유한 높은 안전성과 라이브니스 요구사항을 충족하도록 최적화되어 있습니다. 또한 엑셀라 네트워크는 프로토콜 제품군과 API를 포함합니다. 핵심 프로토콜은 다음과 같습니다:

- **Cross-Chain Gateway Protocol (CGP).** 이 프로토콜은 인터넷의 Border Gateway Protocol과 유사합니다. 이는 여러 독립적인 블록체인 생태계를 연결하며, 그들 간 라우팅을 담당합니다. 블록체인들은 “특수 언어”를 배울 필요가 없으며, 플랫폼 개발자는 체인에 맞춤형 변경을 할 필요가 없습니다. 체인들은 전 세계 네트워크에 쉽게 연결될 수 있습니다.
- **Cross-Chain Transfer Protocol (CTP).** 이 프로토콜은 인터넷의 파일 전송, 하이퍼텍스트 전송 프로토콜과 같은 애플리케이션 계층 프로토콜과 유사합니다. 라우팅 프로토콜(CGP 및 기타 라우팅 기술) 위에 위치하는 애플리케이션 계층 프로토콜 스택입니다. 애플리케이션 개발자는 어느 체인의 디앱이라도 연결하여 크로스체인 요청을 실행할 수 있습니다. 사용자는 CTP 프로토콜을 통해 HTTP GET/POST 요청과 유사한 간단한 API 호출로 어느 체인의 애플리케이션과도 상호작용할 수 있습니다. 개발자는 자산을 임치, 해제, 전송할 수 있으며, 서로 다른 블록체인 플랫폼 간 두 주소 간 전송을 지원합니다. 또한 크로스체인 애플리케이션 트리거를 실행할 수 있습니다 (예: 체인 A의 디앱이 체인 B의 특정 애플리케이션이 “이자율 > X”와 같은 조건을 충족하면 자신의 상태를 업데이트). 나아가 체인 간 앱 간 일반적인 크로스체인 요청을 수행할 수 있습니다 (예: 체인 A의 스마트 계약이 체인 B의 스마트 계약 상태를 업데이트하도록 호출). 이 프로토콜은 블록체인 생태계 전반의 프로그램 조합성을 가능하게 합니다.

엑셀라 네트워크는 다음과 같은 이점을 제공합니다:

- *블록체인 플랫폼 개발자를 위해.* 블록체인을 다른 모든 블록체인 생태계에 쉽게 연결할 수 있습니다. 네트워크에 연결하려면 체인에 임계값 계정(threshold account)만 설정하면 됩니다.
- *디앱 개발자를 위해.* 애플리케이션 개발자는 어디서든 디앱을 호스팅하고, 자산을 임치/해제/전송하며, CTP API를 통해 다른 체인의 애플리케이션과 소통할 수 있습니다.
- *사용자를 위해.* 사용자는 자신의 지갑에서 생태계 전반의 모든 애플리케이션과 직접 상호작용할 수 있습니다.

개발자를 위한 플랫폼. 마지막으로 엑셀라 네트워크는 개발자와 글로벌 커뮤니티를 위한 플랫폼입니다. 거버넌스 모델은 누구에게나 개방되어 있습니다. 개발자는 새로운 통합 지점, 라우팅, 애플리케이션 계층 프로토콜을 제안할 수 있으며, 사용자는 제안에 대한 투표를 통해 채택 여부를 결정할 수 있습니다. 승인되면 검증자가 변경사항을 채택하게 됩니다.

1.1 기존 상호운용성 솔루션

블록체인 간 상호운용성을 해결하기 위한 기존 시도들은 중앙화 거래소, 상호운용 가능한 생태계, 래핑 자산, 토큰 브릿지 네 가지 범주 중 하나에 속합니다. 아래에서 이러한 접근 방식을 간단히 요약합니다.

중앙화 시스템. 오늘날 중앙화 시스템은 생태계의 상호운용성 요구를 충족할 수 있는 유일하게 확장 가능한 솔루션입니다. 이들은 어떤 자산이든 상장하거나 어떤 플랫폼이든 비교적 쉽게 온보딩할 수 있습니다. 그러나 중앙화 시스템은 다양한 보안 문제로 잘 알려져 있으며, 강력한 보안, 투명성, 개방형 거버넌스를 요구하는 신흥 탈중앙화 금융 시스템을 지원하기에는 적합하지 않습니다. 이들만으로는 성장하는 탈중앙화 애플리케이션을 구동할 수 없습니다.

상호운용성 허브. Cosmos, Polkadot, Ava Labs와 같은 프로젝트는 자체 생태계에 속한 *사이드체인* 간 상호운용성을 맞춤형 체인 간 통신 프로토콜로 해결하려 합니다. 예를 들어, Cosmos Zone이라는 사이드 체인을 생성하면 Cosmos Hub와 통신할 수 있습니다. 사이드체인은 텐더민트 합의에 기반해야 하며 Cosmos Hub가 이해할 수 있는 프로토콜을 네이티브로 구사해야 합니다. 그러나 다른 언어를 사용하는 블록체인 및 생태계와의 연결은 외부 기술에 의존합니다.

쌍대 브릿지. 래핑 자산(예: Wrapped Bitcoin)은 생태계의 부족한 크로스체인 상호운용성 공백을 메우려 합니다. 예를 들어 *BTC* [9]는 맞춤형 프로토콜로, 스마트 계약과 담보를 영리하게 조합하여 전송을 보안합니다. 이 솔루션은 각 체인 쌍마다 대상 체인에 새로운 스마트 계약을 배포해야 하므로 상당한 엔지니어링 노력이 필요합니다. 해당 스마트 계약은 원본 체인의 상태 증명을 해석해야 합니다(이는 각 사이드체인이 원칙적으로 다른 체인의 상태를 해석하는 것과 유사). 현재까지 이 접근법으로 배포된 브릿지는 소수에 불과합니다. 또한, 기초 블록체인 중 하나가 합의 규칙이나 트랜잭션 형식을 업그레이드하려고 하면 이 접근법은 확장되지 않습니다. 이는 해당 체인의 상태에 의존하는 모든 스마트 계약이 업그레이드되어야 하기 때문입니다. 또한 검증자를 구성하고 그들이 다양한 자산을 묶어 과잉담보를 설정해야 하므로 자산 전송의 경제 효율성이 제한됩니다.

플랫폼 개발자가 다른 생태계와 연결하기 위해 상태 전이 로직을 스마트 계약으로 다시 작성하는 단일 목적 브릿지도 존재합니다. 하지만 이들은 여러 확장성 문제를 안고 있으며, 생태계 전체가 균일하게 확장되지 못하게 하고 애플리케이션에 추가적인 종속성을 도입합니다. 예를 들어, 한 플랫폼이 변경되면 모든 브릿지의 스마트 계약을 업그레이드해야 합니다. 결국 생태계는 업그레이드 불가능한 교착 상태에 빠집니다. 또한, 단일 목적 브릿지가 플랫폼 A와 B를 연결하고, 또 다른 브릿지가 B와 C를 연결한다고 해서 A의 애플리케이션이 C의 애플리케이션과 소통할 수 있는 것은 아닙니다. 또 다른 단일 목적 브릿지를 구축하거나 애플리케이션 로직을 다시 설계해야 할 수도 있습니다.

상호운용성을 해결하려는 다른 시도로는 연합형 오라클(예: Ren)과 애플리케이션 특화 상호운용 블록체인이 있습니다.

요약하자면, 기존 상호운용성 솔루션은 각 생태계 간 통신을 위해 서로 다른 통신 프로토콜을 이해해야 하므로, 플랫폼 개발자와 애플리케이션 개발자 모두에게 막대한 엔지니어링 작업을 요구합니다. 따라서 오늘날 블록체인 공간에서 상호운용성은 사실상 존재하지 않습니다. 결국 플랫폼 개발자는 플랫폼 자체를 구축하고 자신의 사용 사례에 맞게 최적화하며, 다른 블록체인에 쉽게 연결할 수 있기를 원합니다. 애플리케이션 개발자는 자신에게 가장 적합한 플랫폼에서 디앱을 구축하면서도 사용자, 유동성을 활용하고, 다른 체인의 디앱과

소통할 수 있기를 원합니다.

2 확장 가능한 크로스체인 통신을 위한 탐구

근본적으로 크로스체인 통신은 이질적인 네트워크들이 동일한 언어로 소통할 수 있는 능력을 확보해야 합니다. 이를 해결하기 위해 우리는 엑셀라 프로토콜 제품군을 설명하고, 그 고수준 속성을 기술하며, 이러한 속성이 확장 가능한 크로스체인 통신의 핵심을 어떻게 해결하는지 설명합니다.

1. **“플러그 앤 플레이” 통합.** 블록체인 플랫폼 개발자가 크로스체인을 지원하기 위해 “특수 언어”를 사용하도록 무거운 엔지니어링이나 통합 작업을 수행할 필요가 없어야 합니다. 크로스체인 프로토콜은 기존 또는 신규 블록체인을 마찰 없이 연결할 수 있어야 합니다. 새로운 자산은 모든 생태계에서 자동으로 사용 가능해야 합니다.
2. **크로스체인 라우팅.** 네트워크 주소 발견, 라우팅 경로, 네트워크 탐색과 같은 기능은 인터넷의 핵심이며, BGP 및 기타 라우팅 프로토콜에 의해 지원됩니다. 이와 유사하게, 블록체인 생태계 간 통신을 지원하기 위해 우리는 주소, 애플리케이션, 라우팅의 발견을 지원해야 합니다.
3. **업그레이드 지원.** 블록체인 생태계 중 하나가 변경되더라도 다른 블록체인의 상호운용성에 영향을 주어서는 안 됩니다. 시스템은 업데이트를 인식해야 하며, 이를 지원하는 데 최소한의 노력이 필요해야 합니다(즉, “상태 전이 로직”을 다시 작성할 필요가 없어야 하고, 애플리케이션이 중단되어서는 안 됩니다).
4. **애플리케이션을 위한 통일된 언어.** 애플리케이션은 자신이 속한 체인과 관계없이 자금, 해제, 전송, 다른 애플리케이션과의 소통을 위한 단순한 프로토콜이 필요합니다. 이 프로토콜은 체인 불가지론적이어야 하며, 단순한 호출을 지원해야 합니다. 이는 사용자가 HTTP/HTTPS 프로토콜을 통해 어떤 웹서버와도 통신할 수 있는 것과 유사합니다. 더 많은 네트워크와 자산이 하위 라우팅 프로토콜에 합류함에 따라, 애플리케이션은 소프트웨어 스택을 다시 작성하지 않고도 이를 활용하여 통신할 수 있어야 합니다.

다음으로, 이러한 프로토콜이 충족해야 할 보안 요구사항을 요약합니다.

1. **탈중앙화된 신뢰.** 네트워크와 프로토콜은 탈중앙화되고 개방적이어야 하며, 누구나 공정하게 참여할 수 있어야 합니다.
2. **높은 안전성.** 시스템은 높은 안전성 보장을 충족해야 합니다. 시스템은 크로스체인 네트워크가 처리하는 동안 자산과 상태의 안전성을 보존해야 합니다.
3. **높은 라이브니스.** 시스템은 높은 라이브니스 보장을 충족해야 하며, 크로스체인 기능을 활용하는 애플리케이션을 지원해야 합니다.

이 속성들의 부분 집합을 만족하는 것은 쉽습니다. 예를 들어, 친구들과 함께 연합형 멀티시그 계정을 만들고 해당 체인에서 자산을 잠그고 해제할 수 있습니다. 그러나 이러한 시스템은 본질적으로 담합 및 검열 공격에 취약하며, 검증자가 이를 보호하기 위한 적절한 인센티브를 결여합니다. 누구나 참여할 수 있고 올바른 인센티브가 제공되는 탈중앙화 네트워크와 프로토콜 제품군을 구축하면 마찰 없는 크로스체인 통신을 가능하게

할 수 있습니다. 하지만 이를 해결하는 것은 합의, 암호학, 메커니즘 설계 프로토콜을 신중히 결합해야 하는 어려운 문제입니다.

3 엑셀라 네트워크

엑셀라 네트워크는 플랫폼 개발자 – 통합 작업이 필요 없는 – 와 애플리케이션 개발자 – 전 세계 이동성에 접근하고 전체 생태계와 소통할 수 있는 단순한 프로토콜 및 API – 모두의 요구를 충족하는 크로스체인 통신에 대한 통합 솔루션을 제공합니다.

엑셀라 네트워크는 서로 다른 언어를 사용하는 블록체인 생태계를 연결하는 탈중앙화 네트워크와 그 위의 프로토콜 제품군 및 API로 구성되어, 애플리케이션이 크로스체인 요청을 손쉽게 수행할 수 있게 합니다. 이 네트워크는 Bitcoin, Stellar, Terra, Algorand와 같은 독립형 블록체인과 Cosmos, Avalanche, Ethereum, Polkadot과 같은 상호운용성 허브를 연결합니다. 우리의 목표는 애플리케이션 개발자가 독자적인 크로스체인 프로토콜을 직접 구축하거나 새로운 브릿지가 개발될 때마다 애플리케이션을 다시 작성하지 않고도 범용 프로토콜과 API를 사용하여 이러한 앱을 더 쉽게 구축할 수 있도록 하는 것입니다. 이를 위해 우리는 Cross-Chain Gateway Protocol(섹션 6 참조)과 Cross-Chain Transfer Protocol(섹션 7 참조)을 포함한 프로토콜 제품군을 설계했습니다.

네트워크의 핵심 구성 요소는 기본적인 탈중앙화 프로토콜입니다. 검증자들은 집합적으로 엑셀라 네트워크를 유지하고 엑셀라 블록체인을 보호하는 노드를 실행합니다. 이들은 사용자에게 의해 위임 절차를 통해 선출됩니다. 검증자는 위임받은 지분에 비례하여 투표권을 받습니다. 검증자들은 플랫폼이 연결된 여러 블록체인의 현재 상태에 대해 합의합니다. 블록체인은 크로스체인 라우팅 및 전송 프로토콜을 유지하고 실행하는 역할을 합니다. 거버넌스 규칙은 네트워크 참여자들이 어떤 블록체인을 연결할지, 어떤 자산을 지원할지와 같은 프로토콜 결정을 시행할 수 있도록 합니다.

엑셀라 블록체인은 Cosmos Hub와 유사한 위임 지분 증명(DPoS) 모델을 따릅니다. 사용자는 합의에 참여하고 고품질 서비스를 유지하기 위해 자신의 지분을 묶는 검증자를 선출합니다. DPoS 모델은 대규모 탈중앙화된 검증자 집합을 유지하고, 검증자가 브릿지와 암호학적 임계값 체계를 유지하는 데 책임을 지도록 강력한 인센티브를 제공합니다. 합의 과정의 일부로 검증자들은 다른 블록체인의 라이트 클라이언트 소프트웨어를 실행하여 해당 블록체인의 상태를 검증할 수 있습니다. 검증자들은 이러한 상태를 엑셀라 블록체인에 보고하며, 충분한 수의 보고가 모이면 Bitcoin, Ethereum 및 기타 체인의 상태가 엑셀라에 기록됩니다.

따라서 엑셀라 기본 계층은 어느 시점에서든 외부 블록체인의 상태를 인지하게 되며, 이는 다른 블록체인으로부터의 “인커밍 브릿지”를 형성합니다. 검증자들은 집합적으로 다른 블록체인에서 임계값 서명 계정을 유지합니다(예: 검증자의 80%가 승인하고 공동 서명해야 거래가 실행됨). 이를 통해 자산과 상태를 체인 간에 잠고고 해제하며, 상태를 다른 블록체인에 게시하는 “아웃고잉 브릿지”를 제공합니다. 종합적으로, 엑셀라 네트워크는 탈중앙화된 크로스체인 읽기/쓰기 오라클로 볼 수 있습니다.

이 문서의 나머지 부분에서는 네트워크의 기초와 구성 요소(섹션 4), 네트워크의 기술적 세부사항(섹션 5), 크로스체인 게이트웨이 프로토콜(섹션 6), 크로스체인 전송 프로토콜(섹션 7)을 설명합니다.

4 기초

4.1 표기법과 가정

라운드 R 에서 엑셀라 검증자 집합을 V^r 이라 합시다. 각 검증자는 *가중치*를 가지며, 이는 해당 검증자의 투표권을 나타내는 $(0, 1]$ 구간의 수입니다. 모든 검증자의 가중치 합은 1입니다. 한 검증자가 엑셀라 프로토콜의 규칙에 일관되게 노드를 운영한다면 그 검증자는 *정상*이라 합니다. 블록을 최종화하거나 크로스체인 요청에 서명하기 위해서는 엑셀라는 총 가중치가 $> F$ 인 정상 검증자를 요구합니다. 우리는 매개변수 $F \in [0.5, 1]$ 를 *프로토콜 임계값*이라 부릅니다.

엑셀라는 즉시 최종성을 가진 위임 지분 증명 블록체인에 기반할 수 있습니다. 검증자들은 각 라운드 i 에서 *비잔틴 장애 허용(BFT) 합의*를 실행하여 i_{th} 블록을 최종화합니다. i_{th} 블록이 최종화되면, 새로운 BFT 합의가 실행되어 $i + 1_{th}$ 블록을 최종화하며, 이 과정이 반복됩니다. 검증자는 지분 위임을 통해 선출됩니다. 일정 지분을 보유한 사용자는 검증자 노드를 직접 운영할 수도 있고, 기존 검증자에게 자신의 투표권(지분)을 위임할 수도 있습니다. 그 후 검증자는 위임자의 지분에 따라 대신 투표합니다. 검증자 집합은 업데이트될 수 있으며, 검증자는 집합에 참여하거나 탈퇴할 수 있고, 사용자는 투표권을 위임하거나 철회할 수 있습니다. [17]

다양한 블록체인은 서로 다른 네트워크 가정을 전제로 작동합니다. 동기식 통신은 메시지가 전달되는 시간에 고정된 상한 Δ 가 존재하며, Δ 는 알려져 있고 프로토콜에 내장될 수 있음을 의미합니다. 비동기식 통신은 메시지가 전달되는 데 임의로 긴 시간이 걸릴 수 있으며, 단 하나의 악의적 검증자가 존재하더라도 비동기 네트워크에서는 BFT 프로토콜을 구축할 수 없다는 것이 알려져 있습니다. 동기성과 비동기성 사이의 현실적 절충은 *부분 동기식 통신* 가정입니다. 네트워크는 어떤 알려지지 않은 전역 안정화 시간(GST) 전까지는 완전히 비동기적일 수 있으나, GST 이후에는 알려진 상한 Δ 를 가진 동기식이 됩니다.

일반적인 블록체인은 $> F$ 정상 검증자가 있다는 가정하에 작동합니다. 동기식 네트워크에서는 보통 $F = 1/2$ 로 설정되지만, 부분 동기식 네트워크라는 더 약한 가정에서는 $F = 2/3$ 로 설정됩니다. Bitcoin, 그 포크들, 그리고 현재 작업증명(PoW) 버전의 Ethereum은 동기성을 전제로만 작동합니다. Algorand와 Cosmos 같은 다른 체인들은 부분 동기성만 요구합니다. 엑셀라를 통해 체인을 연결할 때, 연결은 이 체인들 중 가장 강한 네트워크 가정(예: Bitcoin과 Cosmos를 연결할 때는 동기성)을 전제로 작동합니다. 엑셀라 블록체인 자체는 부분 동기식 환경에서 작동하며 따라서 $F = 2/3$ 를 요구하지만, 다른 블록체인의 보안을 신뢰하고 그 보안을 활용한다면 임계값 요구사항을 개선할 수 있습니다.

4.2 암호학적 기초

디지털 서명. *디지털 서명 체계*는 알고리즘 (*Keygen, Sign, Verify*)의 튜플입니다. *Keygen*은 키쌍 (PK, SK)를 출력합니다. SK 소유자만이 메시지에 서명할 수 있으며, 누구나 공개키 PK 로 서명을 검증할 수 있습니다. 오늘날 대부분의 블록체인 시스템은 ECDSA, Ed25519 또는 이들의 몇 가지 변형과 같은 표준 서명 체계를 사용합니다.[2,3]

임계값 서명. *임계값 서명 체계*는 n 명의 참여자가 서명 체계의 비밀키를 분할하여, $t + 1$ 명 이상의 하위 집합은 협력해 서명을 생성할 수 있지만, t 명 이하의 하위 집합은 서명을 생성하거나 비밀키에 대한 정보를 알 수 없도록 합니다. ECDSA와 EdDSA에 대한 임계값 프로토콜이 생성한 서명은 독립적으로 실행한 알고리즘이 생성한 서명과 동일하게 보입니다.

임계값 서명 체계는 일반 서명 체계의 *Keygen*과 *Sign* 알고리즘을 분산형 n 자 프로토콜 $T. Keygen, T. Sign$ 으로 대체합니다. 이러한 프로토콜은 보통 공용 브로드캐스트 채널과 참여자 간의 개인 채널을 모두 필요로 하며, 여러 라운드의 통신을 포함합니다. $T. Keygen$ 이 성공적으로 완료되면 각 사용자는 비밀키 SK 의 분할키 s_i 와 해당 공개키 PK 를 보유합니다. $T. Sign$ 프로토콜은 이들이 특정 메시지에 대해 공개키 PK 하에서 유효한 서명을 생성할 수 있도록 합니다. 이 서명은 원래 서명 체계의 *Verify* 알고리즘으로 누구나 검증할 수 있습니다.

4.3 임계값 서명의 속성

탈중앙화 네트워크에서 특히 바람직한 임계값 체계의 속성은 다음과 같습니다:

불성실한 다수에 대한 보안. 일부 임계값 체계는 참여자 n 의 다수가 정직할 때만 안전하다는 제한을 가집니다. 따라서 임계값 매개변수 t 는 $n/2$ 보다 작아야 합니다.[15] 이 제한은 보통 $t + 1$ 명의 악의적 참여자만으로도 비밀키를 복구할 수 있음에도 불구하고, $2t + 1$ 명의 정직한 참여자가 서명을 위해 필요하다는 사실과 동반됩니다. 이러한 제한을 받지 않는 체계는 *불성실한 다수에 대해 안전하다*고 합니다. 섹션 5.2에서 논의하듯, 크로스체인 플랫폼은 네트워크의 안전성을 극대화하고 가능한 한 많은 부정 행위를 견뎌낼 수 있어야 하므로, 불성실한 다수를 견딜 수 있는 체계가 필요합니다.

프리서명, 비대화식 온라인 서명. 메시지를 서명할 때 참여자의 통신 부담을 줄이기 위해, 최근 프로토콜들은 서명 작업 중 상당 부분을 메시지가 알려지기 전에 “오프라인”으로 수행할 수 있음을 확인했습니다. 이 오프라인 단계의 결과물을 *프리서명*이라 합니다.[18,13] 프리서명의 생성은 $T. Keygen$ 과 $T. Sign$ 과 구별되는 별도의 프로토콜 $T. Presign$ 으로 간주됩니다. 프리서명 프로토콜의 출력은 서명 단계에서 사용될 때까지 참여자들이 비밀로 유지해야 합니다. 이후 서명할 메시지가 알려지면, $T. Sign$ 에서 소량의 “온라인” 작업만 추가로 수행하면 서명이 완성됩니다. 온라인 $T. Sign$ 단계는 참여자 간 통신을 요구하지 않습니다. 각 참여자는 (프리서명을 이용한 메시지 계산)과 자신의 서명 분할키 s_i 를 공개하기만 하면 됩니다. 공개된 후 서명 분할키 $s_1, \dots, s_{t+1}$ 은 누구든 쉽게 결합하여 실제 서명 s 를 생성할 수 있습니다. 이 속성을 *비대화식 온라인 서명*이라 합니다.

강건성. 임계값 체계는 악의적 참여자 일부가 메시지에 서명하거나 비밀키를 알아낼 수 없음을 보장합니다. 그러나 이는 악의적 행위자가 다른 모든 참여자가 키 또는 서명을 생성하지 못하도록 막을 가능성을 배제하지는 않습니다. 일부 체계에서는 단 한 명의 악의적 행위자만 있어도 $T. Keygen$ 또는 $T. Sign$ 이 중단되어 유용한 출력이 없을 수 있습니다. 이 경우 유일한 대안은 프로토콜을 다시 시작하거나 다른 참여자를 포함시키는 것입니다. 탈중앙화 네트워크의 경우, $t + 1$ 명의 참여자가 정직하다면 일부 악의적 참여자가 잘못된 메시지를 보내거나 메시지를 누락하더라도 $T. Keygen$ 과 $T. Sign$ 이 성공해야 합니다. 이 속성을 *강건성*이라 합니다.

잘못된 행위자 식별. $T. Keygen$ 또는 $T. Sign$ 에서 악의적 행위자를 식별하는 능력을 *잘못된 행위자 식별(fault attribution)*이라 합니다. 잘못된 행위자 식별이 없다면 악의적 행위자를 배제하거나 처벌하기 어려워, 그들이 야기한 비용이 모든 참여자에게 전가됩니다. 이 속성은 탈중앙화 네트워크에서도 중요하며, 악의적 행위는 슬래싱 규칙을 통해 식별 및 경제적으로 억제되어야 합니다.

동시 환경에서의 보안. 서명 체계는 여러 개의 키 생성 및 서명 알고리즘 인스턴스가 병렬로 실행되는 동시 환경에서 안전해야 합니다. 이러한 속성을 만족하는 ECDSA 및 Schnorr 체계의 버전이 존재합니다.[13,22]

ECDSA와 EdDSA는 블록체인 영역에서 가장 널리 배포된 서명 체계입니다. 따라서 두 체계의 임계값 버전은 최근 연구 및 개발에서 활발히 다뤄지고 있습니다. 최신 기술에 관심 있는 독자는 참고문헌 [22,13,18]과 최근의 서베이 논문을 참고할 수 있습니다.[12]

5 엑셀라 네트워크

5.1 개방형 크로스체인 네트워크 설계

엑셀라 네트워크가 유지하는 브릿지는 임계값 계정(threshold accounts)에 의해 보강되며, (거의) 모든 검증자가 집합적으로 모든 크로스체인 요청을 승인해야 합니다. 누구나 이러한 브릿지를 보호하기 위해 참여할 수 있는 네트워크를 설계하려면 다음과 같은 기술적 요구사항을 충족해야 합니다:

- **개방형 멤버십.** 모든 사용자는 (네트워크의 규칙에 따라) 검증자가 될 수 있어야 합니다.
- **멤버십 갱신.** 검증자가 정직하게 시스템을 떠날 경우, 그 키는 적절히 폐기되어야 합니다.
- **인센티브와 슬래싱.** 악의적인 검증자는 식별 가능해야 하며, 그들의 행동은 프로토콜에 의해 규명되고 대응되어야 합니다.
- **합의.** 임계값 체계는 독립적 프로토콜로 정의됩니다. 노드 간 메시지를 전파하려면 브로드캐스트와 점대점 비공개 채널이 모두 필요합니다. 또한 검증자들은 임계값 체계의 각 호출에서 최신 상태에 합의해야 하며, 이는 종종 여러 라운드의 상호작용을 포함합니다.
- **키 관리.** 모든 PoS 시스템의 일반적인 검증자들이 자신의 키를 신중히 보호해야 하듯, 엑셀라 검증자도 자신의 임계값 지분을 보호해야 합니다. 키는 주기적으로 회전되어야 하며, 온라인/오프라인 파트로 분리되는 등 관리가 필요합니다.

엑셀라는 커뮤니티가 합의를 수행할 검증자 집합을 선출하는 위임 지분 증명(DPoS) 모델로 시작합니다. 표준 임계값 체계는 모든 참여자를 동일하게 취급하며 합의에서 “가중치” 개념을 갖지 않습니다. 따라서 네트워크는 검증자의 가중치를 반영하도록 이를 조정해야 합니다. 간단한 방법은 더 큰 검증자에게 여러 개의 임계값 지분을 할당하는 것입니다. 검증자가 집합적으로 수행하는 세 가지 기본 기능은 다음과 같습니다:

- **임계값 키 생성.** 표준 블록체인 서명 체계(ECDSA, Ed25519)를 위한 기존 임계값 키 생성 알고리즘은 여러 참여자 간의 대화형 프로토콜입니다. 엑셀라 네트워크에서 특별한 트랜잭션은 검증자들에게 이 상태 기반 프로토콜 실행을 시작하도록 지시합니다. 각 검증자는 비밀 상태를 안전하게 보관하는 임계값 데몬 프로세스를 실행합니다. 프로토콜의 각 단계에서:
 1. 검증자는 프로토콜 상태를 로컬 메모리에 유지합니다.
 2. 비밀 데몬을 호출하여 다른 검증자에게 보낼 메시지를 생성합니다.
 3. 메시지를 브로드캐스트 또는 비공개 채널을 통해 다른 검증자에게 전파합니다.

4. 각 검증자는 상태 전이 함수를 실행하여 자신의 상태를 갱신하고, 프로토콜의 다음 단계로 진행하며 위 단계를 반복합니다.

프로토콜이 끝나면 임계값 공개키가 엑셀라 체인에 생성되며, 이는 사용자(예: 예치용)나 초기 요청을 생성한 애플리케이션에 표시될 수 있습니다.

- **임계값 서명.** 엑셀라 네트워크에서의 서명 요청은 키 생성 요청과 유사하게 처리됩니다. 예를 들어, 사용자가 체인 중 하나에서 자산을 인출하고자 할 때 호출됩니다. 이는 대화형 프로토콜이며, 라운드 간 상태 전이는 엑셀라 블록체인 뷰 및 각 검증자의 로컬 메모리에 의해 전파된 메시지에 따라 촉발됩니다.
- **검증자 멤버십 변경 처리.** 검증자 집합은 새로운 이해관계자가 합류할 수 있도록 주기적으로 교체되어야 합니다. 검증자 집합이 갱신되면 새로운 집합에 공유될 임계값 키도 갱신되어야 합니다. 따라서 언제든지 누구나 참여할 수 있도록 허용한다면 임계값 키를 매우 자주 갱신해야 할 것입니다. 이를 방지하기 위해 우리는 검증자를 매 T 블록마다 교체합니다. T 라운드 간격 내에서 집합 V^R 과 임계값 키는 고정됩니다. 매 T 의 정수 배 라운드에서 검증자 집합을 다음과 같이 갱신합니다:
 1. 라운드 R 에서, 엑셀라 상태는 현재 검증자 집합 V^R 을 기록합니다. $R + 1$ 이 T 의 배수가 아니면 $V^{R+1} = V^R$ 입니다.
 2. 라운드 $((i - 1)T, iT]$ 동안, 사용자는 바인딩/언바인딩 메시지를 게시합니다.
 3. 라운드 iT 종료 시, 이러한 메시지가 V^{iT-1} 에 적용되어 V^{iT} 가 생성됩니다.
- **검증자 회전 시 임계값 키 생성 및 서명.** 엑셀라 블록체인은 라운드 R 에서 새로운 키나 임계값 서명 요청을 발행할 수 있습니다. 서명 과정은 한 라운드 이상 소요되며, 합의를 지연시키지 않기 위해 서명은 라운드 $R + 10$ 이 시작되기 전에 완료되어야 합니다. 특히, 검증자는 라운드 $R + 9$ 의 인증서와 라운드 R 에서 발행된 각 키 생성/서명 요청의 서명을 확인한 후에야 라운드 $R + 10$ 을 시작합니다. 모든 라운드 R 요청의 결과는 블록 $R + 11$ 에 포함되어야 합니다. 즉, 라운드 $R - 11$ 의 결과를 포함하지 않은 라운드 R 블록 제안은 무효이며, 검증자는 이에 투표하지 않습니다. 모든 임계값 메시지가 검증자 집합 갱신 전에 서명되도록 보장하기 위해, 엑셀라는 라운드가 $-1, -2, \dots, -9 \pmod{T}$ 에 해당할 때는 임계값 요청을 발행하지 않습니다.

5.2 네트워크 보안

블록체인 시스템의 보안은 다양한 암호학적, 게임 이론적 프로토콜뿐만 아니라 네트워크의 탈중앙화에 의존합니다. 예를 들어, 지분 증명 블록체인에서는 적절한 인센티브가 없다면 검증자들이 담합하여 기록을 다시 쓰고 다른 사용자의 자금을 탈취할 수 있습니다. 작업 증명 네트워크에서는 충분한 탈중앙화가 없다면 긴 포크와 이중 지출을 발생시키기 쉽습니다. 이는 Bitcoin Gold와 Ethereum Classic에 대한 여러 공격으로 입증되었습니다.

블록체인 보안에 대한 대부분의 연구는 독립 체인에 집중되어 왔습니다. 그러나 체인이 상호운용되면 새로운 공격 벡터를 고려해야 합니다. 예를 들어, Ethereum이 작은 블록체인 X와 연결되어 있다고 가정합니다. 섹션 1.1에서 요약한 엔지니어링 문제와 함께, X의 신뢰 가정이 위배될 경우 어떤 일이 발생하는지를 결정해야 합니다. 이 경우, ETH가 X로 이동했다면, X의 검증자들은 담합하여 ETH를 모두 보유하는 위조된 X의 기록을 생성하고, 이 위조된 합의 증명을 Ethereum에 게시해 ETH를 탈취할 수 있습니다. 상황은 X가 여러 다른 체인과 직접 브릿지로 연결되어 있을 때 더 악화되며, X가 포크하면 그 효과가 모든 브릿지를 통해 확산

됩니다. 각 쌍대 브릿지마다 복구 거버넌스 지침을 마련하는 것은 개별 프로젝트에 지나치게 부담스럽습니다.

엑셀라 네트워크는 다음과 같은 메커니즘을 통해 보안 문제를 해결합니다:

- **최대 안전성.** 엑셀라는 안전성 임계값을 90%로 설정하여, 네트워크가 잠금 자금을 인출하거나 상태 증명을 위조하기 위해서는 거의 모든 검증자가 담합해야 합니다. 실제로 적절한 인센티브가 제공되면 PoS 검증자는 매우 높은 가동률(거의 100%)을 보여왔습니다. 따라서 엑셀라 네트워크는 높은 임계값에도 불구하고 블록을 생성할 수 있습니다. 그러나 드물게 문제가 발생해 네트워크가 정지할 경우, 시스템을 재부팅할 수 있는 강력한 대체 메커니즘이 필요합니다.
- **최대 탈중앙화.** 네트워크는 임계값 서명 체계를 사용하기 때문에 검증자의 수는 이론적으로 무한히 늘릴 수 있습니다. 멀티시그를 사용하는 경우처럼 검증자 수에 따라 복잡도와 수수료가 선형적으로 증가하는 제약이 없으므로, 네트워크는 검증자 수, 거래 제한, 수수료에 의해 제약되지 않습니다.
- **강력한 대체 메커니즘.** 높은 안전성 임계값을 가진 네트워크에서 가장 먼저 고려해야 할 질문은 네트워크 자체가 정지하면 어떻게 되는가입니다. 엑셀라 네트워크가 정지할 경우, 사용자들이 자금을 복구할 수 있는 대체 메커니즘이 있어야 합니다. 이를 위해 엑셀라 검증자가 집합적으로 제어하는 블록체인 X의 각 임계값 브릿지 계정에는 “긴급 해제 키”가 존재합니다. 이 키는 수천 명의 참여자에게 공유될 수 있으며, 심지어 블록체인 X의 커뮤니티 전반에 공유되는 맞춤형 키일 수도 있습니다. 따라서 엑셀라 네트워크가 정지할 경우, 이 키가 대체 수단으로 작동해 자산 복구를 가능하게 합니다(아래 세부 설명 참조).
- **대체 메커니즘의 최대 탈중앙화.** 이 대체 메커니즘은 누구나 비용 없이 참여할 수 있는 2차 복구 집합을 포함합니다. 이 사용자는 온라인 상태일 필요도 없고, 노드를 운영하거나 서로 조율할 필요도 없습니다. 이들은 오직 엑셀라 네트워크가 정지하고 회복할 수 없을 때만 “소집”됩니다. 네트워크 보안은 기본 검증자 집합의 매우 높은 임계값과 최대한 탈중앙화된 2차 복구 집합에 의해 강화됩니다.
- **공유 거버넌스.** 공통 프로토콜이 엑셀라 네트워크를 지배합니다. 사용자들은 집합적으로 네트워크를 통해 어떤 체인을 지원할지 투표할 수 있습니다. 또한 네트워크는 예기치 못한 긴급 상황에서 사용자를 보상하기 위한 자금 풀을 배정하며, 이는 거버넌스 프로토콜을 통해 관리됩니다.

아래에서는 다양한 보안 메커니즘을 논의합니다.

대체 메커니즘. 엑셀라가 높은 임계값 때문에 정지하면, “긴급 해제 키”가 네트워크 제어권을 가집니다. 이 키를 구현하는 방법은 다양하며, 특정 체인/애플리케이션은 “복구 집합”을 위한 다른 변형을 선택하거나 아예 옵트아웃할 수 있습니다:

- **옵션 a.** 블록체인 프로젝트의 재단 및 커뮤니티 내 신뢰할 만한 사람들에게 키를 공유합니다.
- **옵션 b.** 위임 지분 증명 메커니즘을 통해 선출된 당사자들에 키를 공유합니다.
- **옵션 c.** 체인/애플리케이션 X의 자산과 정보를 관리하는 계정의 경우, X의 이해관계자/검증자에게 맞춤형 키를 공유합니다. X가 거버넌스 메커니즘을 갖추고 있다고 가정하면, 엑셀라가 정지했을 때 동일한 거버넌스 메커니즘을 적용해 행동 방침을 결정할 수 있습니다.

복구 사용자들의 신원과 공개키가 주어지면, 누구도 모르는 복구 키의 지분을 생성하는 간단한 프로토콜이 실행됩니다. 또한 복구 집합의 사용자는 거버넌스 메커니즘에 의해 소집되기 전까지 온라인 상태일 필요가 없습니다. 표준 분산 키 생성 프로토콜에 따라, 각 엑셀라 검증자는 임의의 값을 공유합니다. 복구 비밀키는 이러한 값들의 합으로 생성됩니다. 합산은 공개적으로 수행되는 대신, 모든 지분은 복구 사용자의 공개키로 암호화되어 동형적으로 합산됩니다(이는 가법적 동형 암호화와 추가적인 영지식 증명을 필요로 하며, 둘 다 쉽게 구현 가능합니다). 이 프로토콜의 결과는 복구 공개키 RPK 와, 해당 비밀키의 지분 $Enc_i(s_i)$ 를 복구 사용자들의 공개키로 암호화한 수천 개의 암호문입니다. 이들은 소유자에게 분배되거나(예: 체인에 게시) 합니다. 엑셀라 브릿지 계약에는 특정 조건에서 RPK 를 사용하여 자금을 복구하는 옵션이 포함됩니다. 마지막으로, 이 복구 키는 갱신할 수 있으며, 지분을 보유한 사용자 집합도 변경할 수 있습니다. 이는 참여 주주들에게 추가적인 작업을 요구하지 않습니다.

엑셀라에 연결된 체인 X가 고장날 경우, 몇 가지 대안이 있습니다:

- 하루 동안 X에서 입출금할 수 있는 자산의 USD 가치를 제한합니다. 이렇게 하면 악의적인 체인 X는 검증자가 이를 감지하기 전에 전체 자산의 일부만 탈취할 수 있으며, 이후 거버넌스 메커니즘이 작동합니다.
- 엑셀라 거버넌스 모듈을 사용하여 상황에 대한 결정을 투표할 수 있습니다. 예를 들어, 단순한 버그로 커뮤니티가 X를 재시작한다면, 엑셀라 거버넌스는 연결을 중단된 시점에서 재개할 수 있도록 결정할 수 있습니다.
- ETH가 X로 이동했다면, 맞춤형 Ethereum 복구 키를 사용하여 ETH 자산에 대한 처리를 결정할 수 있습니다.

6 크로스체인 게이트웨이 프로토콜(CGP)

이 섹션에서는 많은 애플리케이션의 요구에서 공통적으로 등장하는 두 가지 핵심 예시를 통해 크로스체인 게이트웨이 프로토콜과 라우팅 메커니즘을 설명합니다:

- **상태 동기화(섹션 6.2).** 소스 블록체인 S 의 상태에 대한 정보를 대상 블록체인 D 의 상태에 게시합니다.
(예: Bitcoin 블록 헤더를 Ethereum 블록체인에 게시.)
- **자산 전송(섹션 6.3).** 디지털 자산을 S 에서 D 로, 다시 D 에서 S 로 전송합니다.
(예: Bitcoin 블록체인에서 Ethereum 블록체인으로 비트를 전송한 뒤, 다시 Bitcoin 블록체인으로 되돌리기.)

단순화를 위해 체인 D 는 최소한의 스마트 컨트랙트 지원을 갖춘다고 가정하며, S 는 어떤 블록체인이든 될 수 있습니다.

6.1 다른 체인상의 계정

서로 다른 체인을 브릿지하기 위해, 임계값 계정이 각 체인에 생성되어 가치와 정보의 흐름을 제어합니다. 체인 $Chain$ 에 대해, 해당 계정을 $Chain_{Axelar}$ 로 표기합니다.

Bitcoin 계정. Bitcoin 및 기타 비(非) 스마트 컨트랙트 체인의 경우 엑셀라 검증자는 섹션 5.1에 따라 임계값 ECDSA 키를 생성합니다. 이 키는 Bitcoin 상의 ECDSA 계정을 제어하며, 사용자가 예치를 보내는 목적지 주소가 됩니다. 사용자 요청에 따라 개인화된 임계값 키를 생성할 수 있습니다. 키는 주기적으로 갱신될 수 있으며, 최신 키와 개인화 키는 엑셀라 노드를 조회하여 확인할 수 있습니다.

스마트 컨트랙트 지원 체인의 임계값 브릿지 계정. 체인을 SC로 표기합니다. 검증자는 해당 체인이 지원하는 키 유형에 따라 섹션 5.1에 따라 임계값 ECDSA 또는 ED25519 키를 생성합니다. 우리가 지칭하는 체인이 명확한 경우 이 키를 PK_{Axelar} 로 표기합니다. 이 키는 SC 상의 스마트 컨트랙트 계정 SC_{Axelar} 를 제어하며, SC의 어떤 애플리케이션도 SC_{Axelar} 를 조회하여 해당 키의 PK 주소를 알 수 있습니다. 이를 통해 어떤 SC 애플리케이션도 SK_{Axelar} 가 서명한 메시지를 인식할 수 있습니다. 프로토콜은 또한 PK_{Axelar} 의 순환(회전) 값을 고려해야 하며, 이는 다음과 같이 이루어집니다:

1. SC에 SC_{Axelar} 를 초기화합니다. 이는 상태의 일부로 PK_{Axelar} 를 저장하며, 엑셀라의 제네시스 값으로 초기화됩니다. SC_{Axelar} 에는 PK를 갱신하는 규칙도 포함됩니다.
2. PK_{Axelar} 를 갱신하려면, 현재의 SK_{Axelar} 서명이 첨부된 형식 $(update, PK_{new})$ 의 트랜잭션을 제출해야 합니다. 그러면 컨트랙트는 $PK_{Axelar} = PK_{new}$ 로 설정합니다.
3. 검증자가 SC에 대한 임계값 키를 PK^i 에서 PK^{i+1} 로 갱신할 때마다, 엑셀라는 검증자에게 SK^i 를 사용해 $(update, PK^{i+1})$ 에 서명하도록 요청합니다. 이후 이 서명은 SC_{Axelar} 에 게시되어 PK_{Axelar} 가 갱신됩니다.

6.2 상태 동기화

S 체인의 상태에 대한 임의의 질문을 q_S 로 표기합니다. 이러한 질문의 예시는 다음과 같습니다:

- “트랜잭션 tx가 (있다면) 몇 번째 블록 라운드에 등장했는가?”
- “특정 데이터 필드의 값은 무엇인가?”
- “블록 라운드 314159에서 S 전체 상태의 머클 루트 해시는 무엇인가?”

q_S 에 대한 올바른 답을 a_S 로 표기하고, 최종 사용자 또는 애플리케이션이 a_S 가 체인 D 에 게시되기를 요구한다고 가정합니다. 엑셀라 네트워크는 다음과 같이 이 요구를 충족합니다:

1. 사용자는 브릿지 계정 중 하나(이후 검증자에 의해 수집됨) 또는 엑셀라 블록체인에 직접 요청 q_S 를 게시합니다.
2. 엑셀라 합의의 일부로, 각 검증자는 체인 S , D 에 대한 노드 소프트웨어를 실행해야 합니다. 엑셀라 검증자는 S 체인 노드 소프트웨어의 API를 조회하여 답 a_S 를 얻고, 그 답을 엑셀라 체인에 보고합니다.
3. 라운드 R 에서 가중치 합 $> F$ 의 검증자가 동일한 답을 보고하면, 엑셀라는 검증자에게 a_S 에 서명하도록 요청합니다.
4. 검증자들은 임계값 암호화를 사용해 a_S 에 서명합니다. 이 서명은 블록 $R + 11$ 에 포함됩니다.
5. 누구나 블록 $R + 11$ 에서 서명된 값 a_S 를 가져와 D 에 게시할 수 있습니다.

- 요청은 처리되었습니다. 이제 D 의 어떤 애플리케이션도 서명된 값 a_S 를 가져와 최신 PK_{Axelar} 를 D_{Axelar} 에 조회하고, a_S 의 서명이 PK_{Axelar} 에 대응하는지 검증할 수 있습니다. 검증자는 또한 a_S 를 체인 D 의 브릿지 계정에 게시하며, 애플리케이션은 이를 조회할 수 있습니다.

6.3 크로스체인 자산 전송

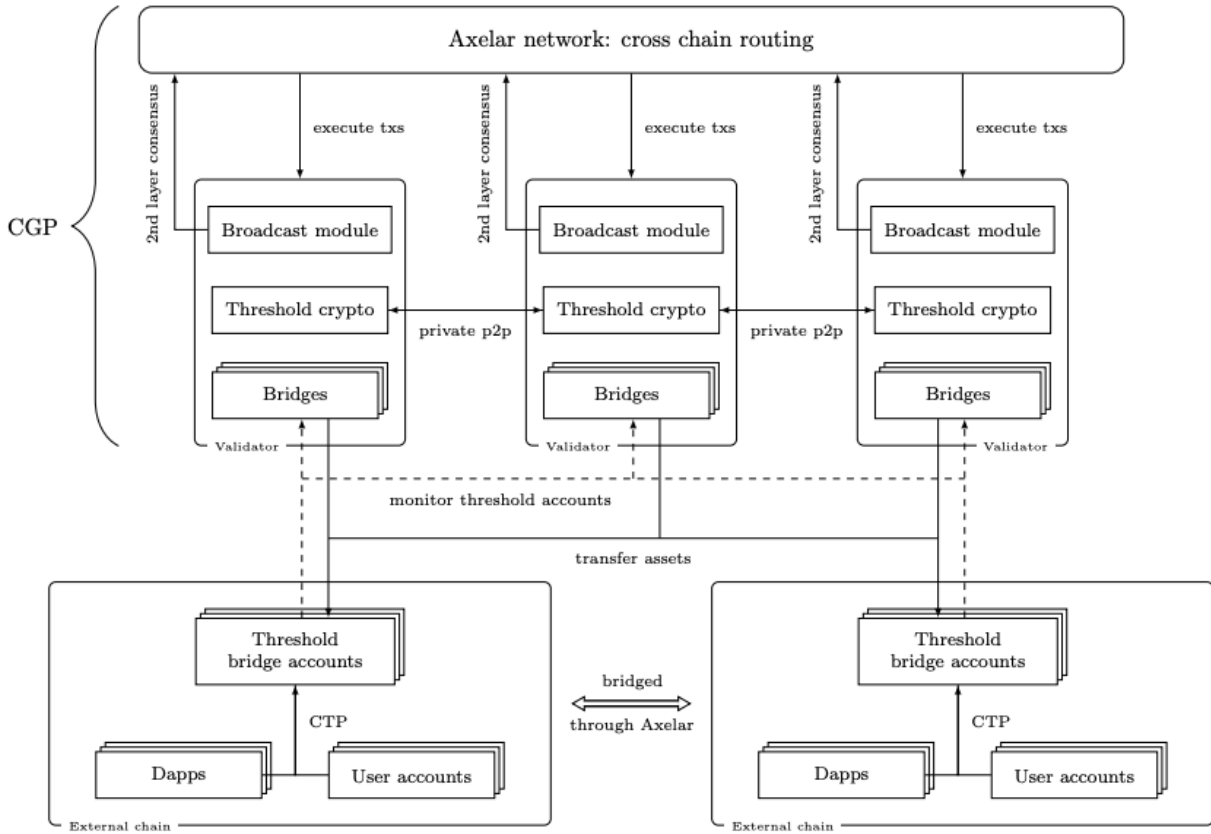
네트워크는 섹션 6.2의 상태 동기화 워크플로를 확장하여 디지털 자산의 크로스체인 전송을 가능하게 합니다.

충분한 공급의 pegged- S 토큰이 초기화 시점에 D_{Axelar} 에 의해 발행되어 관리됩니다. 사용자가 소스 체인 S 의 토큰 x 를 대상 체인 D 의 pegged- S 토큰 x 로 교환하고, 이를 사용자가 선택한 D 주소 w_D 에 예치하기를 요구한다고 가정합니다. 우리는 스마트 컨트랙트를 지원하지 않는 Bitcoin과 같은 체인을 포함하여, 임의의 소스 체인 S 를 지원하는 완전 일반적 워크플로를 제시합니다:

- 사용자(또는 사용자를 대리하는 애플리케이션)는 전송 요청 (x, w_D) 를 임계값 브릿지 계정에 게시하며, 이는 이후 엑셀라 네트워크로 라우팅됩니다.
- 엑셀라 검증자는 임계값 암호화를 사용해 S 용 신규 예치 주소 d_S 를 공동으로 생성합니다. 그들은 d_S 를 엑셀라 블록체인에 게시합니다.
- 사용자(또는 대리 애플리케이션)는 엑셀라 블록체인을 모니터링하여 d_S 를 확인합니다. 사용자는 자신이 선호하는 S 체인 소프트웨어를 사용하여 일반적인 S 트랜잭션 t_{xS} 를 통해 S -토큰 x 를 주소 d_S 로 전송합니다.
(임계값 속성을 갖는 d_S 의 특성상, 검증자의 임계값 수가 협력하지 않으면 d_S 에서 토큰을 지출할 수 없습니다.)
- t_{xS} 는 엑셀라에 게시됩니다. 검증자는 S 체인 노드 소프트웨어의 API를 조회하여 t_{xS} 의 존재 여부를 확인하고, 응답이 “true”이면 그 답을 엑셀라 체인에 보고합니다.
- 라운드 R 에서 가중치 합 $> F$ 의 검증자가 t_{xS} 에 대해 “true”를 보고하면, 엑셀라는 검증자에게 D_{Axelar} 에서 w_D 로 pegged- S 토큰 x 를 전송하는 트랜잭션 a_D 에 서명하도록 요청합니다.
- 검증자들은 임계값 암호화를 사용해 a_D 에 서명합니다. 이 서명은 블록 $R + 11$ 에 포함됩니다.
- 누구나 블록 $R + 11$ 에서 서명된 값 a_D 를 가져와 D 에 게시할 수 있습니다.
- a_D 가 D 에 게시되면 요청은 처리되며, 전송이 완료됩니다.

이제 사용자가 체인 D 의 wrapped- S 토큰 x' 를 체인 S 로 되돌려, 사용자가 선택한 S 주소 w_S 에 예치하기를 요구한다고 가정합니다. 워크플로는 다음과 같습니다:

- 사용자는 자신이 선호하는 D 체인 소프트웨어를 사용해 일반적인 D 트랜잭션을 통해 wrapped- S 토큰 x' 를 c_D 에 예치함으로써 전송 요청 (x', w_S) 를 시작합니다.
- (x', w_S) 는 엑셀라에 게시됩니다. 검증자는 D 체인 노드 소프트웨어의 API를 조회하여 (x', w_S) 의 존재 여부를 확인하고, 응답이 “true”이면 그 답을 엑셀라 체인에 보고합니다.



[Figure 1: Component diagram]

3. 라운드 R 에서 가중치 합 $> F$ 의 검증자가 (x', w_S) 에 대해 “true”를 보고하면, 엑셀라는 검증자에게 S_{Axelar} 에서 w_S 로 S 토큰 x' 을 전송하는 트랜잭션 a_S 에 서명하도록 요청합니다.
4. 검증자들은 임계값 암호화를 사용해 a_S 에 서명합니다. 이 서명은 블록 $R + 11$ 에 포함됩니다.
5. 누구나 블록 $R + 11$ 에서 서명된 값 a_S 를 가져와 S 에 게시할 수 있습니다.
6. a_S 가 S 에 게시되면 요청은 처리되며, 전송이 완료됩니다.

CGP 라우팅 계층이 지원하는 추가 요청에는 체인 간 자산의 잠금, 잠금 해제, 전송 등이 포함됩니다.

원자적 크로스체인 트랜잭션 흐름 달성. 크로스체인 요청 유형에 따라, 엑셀라는 해당 트랜잭션들이 여러 체인에서 모두 실행되거나 전혀 실행되지 않도록 노력합니다. 이를 위해, 모든 요청은 엑셀라 블록체인에서 다음 상태 중 하나를 가집니다: (*initialized, pending, completed, timed out*). *pending* 단계에서 *timeout*이 발생하면, 요청은 오류 코드를 반환합니다. 일부 타임아웃 이벤트는 *환불 이벤트*도 시작합니다. 예를 들어, 한 체인의 자산이 다른 체인의 자산으로 전환되어야 하는데 수신 체인이 트랜잭션을 처리하지 않았다면, 자산은 원래 사용자에게 환불됩니다.

7 크로스체인 전송 프로토콜(CTP)

CTP는 애플리케이션이 크로스체인 기능을 손쉽게 활용할 수 있도록 하는 애플리케이션 수준 프로토콜입니다. 우리는 통합 방식을 (예: DeFi에서 사용되는) 자산 전송 기능에 초점을 맞춰 설명합니다. 이러한 애플리케이션은 일반적으로 프론트엔드 GUI, 하나의 체인에 배포된 스마트 컨트랙트, 프론트엔드와 스마트 컨트랙트 사이에서 트랜잭션을 게시하는 중간 노드의 세 가지 주요 구성 요소로 이루어집니다. 프론트엔드는 사용자의 지갑과 상호작용하여 예치 수락, 인출 처리 등을 수행합니다. 애플리케이션은 HTTP/HTTPS의 GET/POST 메시드에 유사한 CTP 쿼리를 호출함으로써 크로스체인 기능을 활용할 수 있습니다. 이러한 쿼리는 이후 실행을 위해 CGP 계층에 의해 수집되며, 결과가 사용자에게 반환됩니다.

- **CTP 쿼리.** 애플리케이션 개발자는 어떤 체인에서든 애플리케이션을 호스팅하고, 임계값 브릿지 계정과 스마트 컨트랙트를 통합하여 CTP 쿼리를 실행할 수 있습니다.
- **임계값 브릿지 계정.** 애플리케이션 개발자가 체인 A에 컨트랙트를 구축한다고 가정합니다. 그러면 크로스체인 지원을 얻기 위해 임계값 브릿지 컨트랙트를 참조할 것입니다. 이 컨트랙트는 애플리케이션이 다음을 수행하도록 허용합니다:
 - 통신하려는 블록체인을 등록합니다.
 - 활용하려는 해당 블록체인의 자산을 등록합니다.
 - 예치 수락, 인출 처리, 기타 기능(예: ERC-20 컨트랙트 호출과 유사)과 같은 자산에 대한 연산을 수행합니다.

체인 A에 네이티브로 존재하는 유명 DeFi 애플리케이션 MapleSwap이 임계값 브릿지 계정에 등록했다고 가정해봅시다. 해당 체인에서 애플리케이션을 관리하는 계약은 엑셀라 검증자들이 집합적으로 관리합니다. 사용자가 각각 두 체인에 걸쳐 존재하는 자산 X와 Y 사이의 거래쌍에 예치를 제출하려고 한다고 합니다. 사용자가 이러한 요청을 제출하면, 해당 요청은 임계값 브릿지 계정을 통해 엑셀라 네트워크로 라우팅되어 처리됩니다. 이때, 다음 단계들이 수행됩니다:

1. 엑셀라 네트워크는 이 애플리케이션이 해당 자산에 대한 크로스체인 지원을 등록했다는 것을 인식합니다. 네트워크는 임계값 암호화와 합의를 활용하여 사용자를 위한 예치 키를 체인 A와 B에서 생성합니다.
2. 관련 공개키가 애플리케이션에 반환되어 사용자에게 표시되며, 사용자는 자신이 선호하는 지갑을 사용해 예치를 제출할 수 있습니다. 해당 비밀키는 모든 엑셀라 검증자에게 분산 공유됩니다.
3. 예치가 확인되면, 엑셀라는 해당 사용자가 관련 체인에 자산을 예치했음을 기록하기 위해 크로스체인 디렉터리를 업데이트합니다.
4. 엑셀라 검증자들은 다자간 프로토콜을 실행하여, 체인 A의 임계값 브릿지 계정을 업데이트할 수 있는 임계값 서명을 생성합니다.
5. 이후 CTP 쿼리는 DeFi 애플리케이션 스마트 컨트랙트에 반환되며, 애플리케이션은 자신의 상태를 업데이트하거나, 수익률 공식과 환율을 조정하거나, 기타 애플리케이션 상태 관련 조건을 실행할 수 있습니다.

이 과정 전반에서 엑셀라 네트워크는 고수준에서 탈중앙화된 크로스체인 읽기/쓰기 오라클로 기능하며, CGP는 체인 간 라우팅 계층, CTP는 애플리케이션 프로토콜로 작동합니다.

추가적인 크로스체인 요청. CTP는 블록체인 간 애플리케이션 사이의 보다 일반적인 크로스체인 요청을 지원합니다. 예를 들어:

- **공개키 네임 서비스(PKNS) 수행.** 이는 공개키를 전화번호/트위터 핸들에 매핑하는 범용 디렉터리입니다(일부 프로젝트, 예: Celo, 는 이러한 기능을 자체 플랫폼 내에서 제공합니다).
- **크로스체인 애플리케이션 트리거.** 체인 A의 애플리케이션은 체인 B의 다른 애플리케이션이 특정 검색 조건(예: $\text{금리} < X$)을 충족하면 자신의 상태를 업데이트할 수 있습니다.
- **스마트 컨트랙트 조합성.** 체인 A의 스마트 컨트랙트는 체인 B의 컨트랙트 상태를 기반으로 자신의 상태를 업데이트하거나, 체인 B의 스마트 컨트랙트를 업데이트하는 동작을 트리거할 수 있습니다.

고수준에서 이러한 요청들은 CTP, CGP, 엑셀라 네트워크가 집합적으로 블록체인 간 임의의 검증 가능한 상태 정보를 전달하고 기록할 수 있기 때문에 처리될 수 있습니다.

8 요약

향후 수년간, 주요 애플리케이션과 자산은 여러 블록체인 생태계 위에 구축될 것입니다. 엑셀라 네트워크는 이러한 블록체인을 균일한 크로스체인 통신 계층에 플러그인할 수 있도록 사용될 수 있습니다. 이 계층은 플랫폼 구축자와 애플리케이션 개발자의 요구를 충족하는 라우팅 및 애플리케이션 수준 프로토콜을 제공합니다. 애플리케이션 개발자는 자신에게 가장 적합한 플랫폼 위에 구축하면서도 단순한 프로토콜과 API를 활용해 글로벌 크로스체인 유동성, 사용자에게 접근하고, 다른 체인과 소통할 수 있습니다.

참고문헌

[1] Althea peggy. <https://github.com/cosmos/peggy>. [2쪽 참조.]

[2] Deterministic usage of the digital signature algorithm (dsa) and elliptic curve digital signature algorithm (ecdsa). <https://tools.ietf.org/html/rfc6979>. [5쪽 참조.]

[3] Edwards-curve digital signature algorithm (eddsa). <https://tools.ietf.org/html/rfc8032>. [5쪽 참조.]

[4] Eos.io technical white paper v2. <https://github.com/EOSIO/Documentation/blob/master/TechnicalWhitePaper.md>. [1쪽 참조.]

[5] Ethereum: A secure decentralised generalised transaction ledger. <https://ethereum.github.io/yellowpaper/paper.pdf>. [1쪽 참조.]

[6] The near white paper. <https://near.org/papers/the-official-near-white-paper/>. [1쪽 참조.]

[7] Rainbow bridge. <https://github.com/near/rainbow-bridge>. [2쪽 참조.]

- [8] Ren: A privacy preserving virtual machine powering zero-knowledge financial applications. <https://whitepaper.io/document/419/ren-litepaper>. [3쪽 참조.]
- [9] tbtc: A decentralized redeemable btc-backed erc-20 token. <https://docs.keep.network/tbtc/index.pdf>. [2쪽 참조.]
- [10] Thorchain: A decentralized liquidity network. <https://thorchain.org/>. [3쪽 참조.]
- [11] Kurt M. Alonso. Zero to monero. <https://www.getmonero.org/library/Zero-to-Monero-1-0-0.pdf>. [1쪽 참조.]
- [12] Jean-Philippe Aumasson, Adrian Hamelink, and Omer Shlomovits. A survey of ecdsa threshold signing. Cryptology ePrint Archive, Report 2020/1390, 2020. <https://eprint.iacr.org/2020/1390>. [6쪽 참조.]
- [13] Ran Canetti, Nikolaos Makriyannis, and Udi Peled. Uc non-interactive, proactive, threshold ecdsa. Cryptology ePrint Archive, Report 2020/492, 2020. <https://eprint.iacr.org/2020/492>. [6쪽 참조.]
- [14] cLabs Whitepapers. <https://celo.org/papers>. [1쪽 참조.]
- [15] Ivan Damgård, Tomas Pelle Jakobsen, Jesper Buus Nielsen, Jakob Illeborg Pagter, and Michael Bækvang Østergaard. Fast threshold ECDSA with honest majority. In *SCN, volume 12238 of Lecture Notes in Computer Science*, pages 382–400. Springer, 2020. [6쪽 참조.]
- [16] Manu Drijvers, Kasma Edalatnejad, Bryan Ford, Eike Kiltz, Julian Loss, Gregory Neven, and Igores Stepanovs. On the security of two-round multi-signatures. In *IEEE Symposium on Security and Privacy*, pages 1084–1101. IEEE, 2019. [6쪽 참조.]
- [17] Cynthia Dwork, Nancy Lynch, and Larry Stockmeyer. Consensus in the presence of partial synchrony. <https://groups.csail.mit.edu/tds/papers/Lynch/jacm88.pdf>. [5쪽 참조.]
- [18] Rosario Gennaro and Steven Goldfeder. One round threshold ecdsa with identifiable abort. Cryptology ePrint Archive, Report 2020/540, 2020. <https://eprint.iacr.org/2020/540>. [6쪽 참조.]
- [19] Yossi Gilad, Rotem Hemo, Silvio Micali, Georgios Vlachos, and Nickolai Zeldovich. Algorand: Scaling byzantine agreements for cryptocurrencies. Proceedings of the 26th Symposium on Operating Systems Principles, 2017. <https://dl.acm.org/doi/pdf/10.1145/3132747.3132757>. [1쪽 참조.]
- [20] Evan Kereiakes, Do Kwon, Marco Di Maggio, and Nicholas Platiás. Terra money: Stability and adoption. https://terra.money/Terra_White_paper.pdf. [1쪽 참조.]

- [21] Aggelos Kiayias, Alexander Russell, Bernardo David, and Roman Oliynykov. Ouroboros: A provably secure proof-of-stake blockchain protocol. <https://eprint.iacr.org/2016/889.pdf>. [1쪽 참조.]
- [22] Chelsea Komlo and Ian Goldberg. Frost: Flexible round-optimized schnorr threshold signatures. Cryptology ePrint Archive, Report 2020/852, 2020. <https://eprint.iacr.org/2020/852>. [6쪽 참조.]
- [23] Jae Kwon and Ethan Buchman. Cosmos: A network of distributed ledgers. <https://cosmos.network/resources/whitepaper>. [1쪽, 2쪽 참조.]
- [24] Avalanche Team. Avalanche platform. <https://www.avalabs.org/whitepapers>. [1쪽, 2쪽 참조.]
- [25] Gavin Wood. Polkadot: Vision for a heterogeneous multi-chain framework. <https://polkadot.network/PolkaDotPaper.pdf>. [1쪽, 2쪽 참조.]