

아르키움(Arcium)

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.

아르키움 퍼플페이지: 인코딩된 슈퍼컴퓨터

- Yannik Schrade
- Leopold Joy
- Daniel Filipe Nunes Silva
- Nicolas Le Bel
- Lukas Steiner
- Arihant Bansal
- Nicolas Schapeler

2026년 5월 11일

초록

본 초록은 암호자산 백서에 대한 소개로 읽혀야 한다. 잠재적 보유자는 본 요약만이 아니라 암호자산 백서 전체 내용을 기반으로 해당 암호자산 구매 여부를 결정해야 한다. 본 암호자산의 대중 대상 제공은 금융 상품의 매수 제안 또는 권유를 구성하지 않으며, 그러한 제안 또는 권유는 관련 국가 법률에 따라 투자설명서(prospectus) 또는 기타 공식 문서를 통해서만 이루어질 수 있다. 본 암호자산 백서는 유럽의회 및 이사회 규정(EU) 2017/1129(36)에서 정의하는 투자설명서(prospectus) 또는 연합법 및 국가법에 따른 기타 제안 문서에 해당하지 않는다.

아르키움 네트워크(Arcium Network)는 안전한 다자간 연산(MPC, Multi-Party Computation)과 독자적인 토큰 경제 모델을 기반으로 하는 탈중앙화 기밀 연산 프로토콜을 도입한다. 최첨단 MPC 프로토콜과 최신 블록체인 인프라를 결합함으로써, 아르키움은 기밀 투표부터 연합 머신러닝에 이르기까지 기밀성 보존 연산을 확장 가능하고 신뢰 불필요한 방식으로 실행할 수 있도록 지원한다.

네트워크의 핵심에는 ARX 토큰이 존재하며, 이는 고정 공급량을 가진 유틸리티 토큰으로서 노드 운영자의 인센티브를 네트워크의 연산 처리량과 정렬하는 데 사용된다. 노드 운영자는 자신의 검증된 연산 성능에 비례하는 하드웨어 자원을 활성화하기 위해 ARX를 스테이킹하며, 이를 통해 인프라 성장이 실제 네트워크 수요와 정렬된다. 연산 고객은 각 연산에 대해 기본 가격과 고객이 직접 정의한 우선순위 수수료를 지불하며, 이러한 수수료는 해당 작업을 수행한 Arx 노드에 전액 귀속된다.

아르키움의 중개자 없는 아키텍처와 암호경제적 인센티브 설계는 노드 운영자, 위임자, 연산 고객 등 참여자들이 집단적으로 네트워크 효율성을 향상시키는 자기 강화적 생태계를 가능하게 한다. 토큰 유틸리티를 물리적 하드웨어 제공 및 온체인 거버넌스에 연결함으로써, 아르키움은 주문형 기밀 연산이 경제적으로 지속 가능하면서도 보편적으로 접근 가능한 새로운 패러다임을 제시한다.

핵심 혁신 요소에는 독립적인 비잔틴 장애 허용(Byzantine fault tolerant, 이하 BFT) MPC 실행을 가능하게 하는 다자간 실행 환경(MXE, Multi-Party eXecution Environments), 온체인 연산 우선순위 수수료 시장, 그리고 스테이킹을 통해 네트워크 사용량과 수요가 연결되는 고정 공급 유틸리티 토큰이 포함된다. 그 결과, 아르키움 네트워크는 틈새 암호 연산 작업에서부터 엔터프라이즈급 기밀 컴퓨팅에 이르기까지 확장 가능한 네트워크로 자리매김하며, 모든 과정은 투명하고 인센티브가 정렬된 메커니즘에 의해 운영된다.

면책 조항

본 암호자산 백서는 유럽연합 회원국 내 어떠한 관할 당국의 승인도 받지 않았다. 본 암호자산의 제공자는 본 암호자산 백서의 내용에 대해 전적인 책임을 진다.

본 암호자산 백서는 규정(EU) 2023/1114 제2편(Title II)을 준수하며, 경영진의 최선의 지식에 근거할 때 본 백서에 제시된 정보는 공정하고 명확하며 오해의 소지가 없으며, 본 백서는 그 의미에 영향을 줄 수 있는 어떠한 누락도 포함하지 않는다.

본 백서에서 언급된 암호자산은 일부 또는 전부 가치가 상실될 수 있으며, 항상 양도 가능하거나 유동적이지 않을 수 있다.

본 백서에서 언급된 유틸리티 토큰은 암호자산 프로젝트의 실패 또는 중단과 같은 경우, 본 백서에서 약속된 재화 또는 서비스와 교환되지 못할 수 있다.

본 백서에서 언급된 암호자산은 유럽의회 및 이사회 지침 97/9/EC에 따른 투자자 보상 제도의 적용 대상이 아니다. 또한 본 백서에서 언급된 암호자산은 유럽의회 및 이사회 지침 2014/49/EU에 따른 예금 보장 제도의 적용 대상도 아니다.

1 토큰 제공자 및 발행자 정보

발행 주체, 법적 형태, 경영진, 기업 거버넌스 및 MiCAR 부속서 I “유틸리티 토큰 암호자산 백서를 위한 공시 항목(DISCLOSURE ITEMS FOR THE CRYPTO-ASSET WHITE PAPER FOR A UTILITY TOKEN)” Part A에 따라 요구되는 기타 모든 공시 정보는 아래 섹션에 포함되어 있다.

항목(Field)	내용(Value)
명칭(Name)	Arcium Association
법적 형태(Legal Form)	스위스 법률에 따른 협회(Association under Swiss law)
등록 주소(Registered Address)	Grabenstrasse 25, 6340 Baar, Switzerland
등록일(Date of Registration)	2022년 7월 8일
식별번호(Identifier)	CHE-355.161.855
전화번호(Telephone)	[number]
이메일(Email)	contact@arcium.com
응답 시간(Response Time)	영업일 기준 1~2일
발행자 사업(Issuer business)	IT, 소프트웨어 개발

1.1 재무 상태

Arcium Association은 2022년 중반에 설립되었다. 이에 따라 부속서 I 제10조에서 요구하는 3개 연도 기준 대신, 2022년 중반부터 2023년 말까지의 첫 번째 장기 사업연도를 기준으로 재무 상태를 제시할 수 있다.

2022년 7월부터 2023년 12월까지의 기간 동안 Arcium Association의 재무 상태는 양호했던 것으로 평가된다. 이 기간 동안 회사는 운영을 시작하고, 본사, 경영진, 거버넌스 구조를 구축하였다. 초기 자본은 창립 멤버들로부터의 투자로 조달되었으며, 이들은 현재 경영진을 구성하고 있다. 이후 시드 및 전략적 투자 라운드를 통해 총 900만 달러 규모의 추가 자금을 유치하였다.

1.2 핵심 사업

Arcium Association은 아arki움 네트워크의 개발 및 채택을 촉진하고 지원한다. 아arki움 네트워크는 안전한 데이터 협업을 가능하게 하는 기밀 컴퓨팅 네트워크이다. 분산형 다자간 연산(MPC, Multi-Party Computation) 아키텍처를 활용함으로써, 데이터는 처리 전 과정에서 암호화된 상태로 유지되며, 단일 실패 지점을 제거하고 사이버 위협으로부터 보호된다.

네트워크의 온체인 오케스트레이션(on-chain orchestration)은 연산 스케줄링, 보상, 보안 집행을 관리한다. 노드들은 클러스터를 형성하여 안전한 연산을 수행하며, 블록체인 시스템과 원활하게 통합되어 기밀성을 유지하면서도 효율성을 향상시킨다.

아рки움의 유연한 구성 구조는 사용자가 자신의 보안 요구사항에 맞추어 MPC 설정을 커스터마이징할 수 있도록 한다. 기밀성과 탈중앙화된 신뢰를 핵심 가치로 삼음으로써, 아рки움은 산업 전반에 걸쳐 안전한 데이터 활용 방식을 재정의하고 있다.

1.3 경영진

- Yannik Schrade, CEO, c/o Arcium Association, Grabenstrasse 25, 6340 Baar, Switzerland
- Lukas Steiner, COO, c/o Arcium Association, Grabenstrasse 25, 6340 Baar, Switzerland
- Nicolas Schapeler, CTO, c/o Arcium Association, Grabenstrasse 25, 6340 Baar, Switzerland
- Julian Descher, CSO, c/o Arcium Association, Grabenstrasse 25, 6340 Baar, Switzerland

2 서론

2.1 탈중앙화 기밀 연산의 필요성

데이터는 세 가지 상태로 존재한다: 저장 중(at rest), 전송 중(in transit), 그리고 사용 중(in use). 저장 중인 데이터와 전송 중인 데이터에 대한 암호화는 이미 잘 정립되어 있지만, 실제로 처리되고 있는 데이터, 즉 사용 중인 데이터는 여전히 위험에 노출되어 있다. 런타임 데이터 보안을 위한 기존 접근 방식은 하드웨어 기반 신뢰 실행 환경(TEE, Trusted Execution Environment)에 의존하며, 이는 안전한 엔클레이브(enclave) 안에서 연산을 격리하는 구조이다. 그러나 TEE는 다음과 같은 치명적인 문제를 가진다:

- 하드웨어 종양화: TEE(Intel SGX, AMD SEV 등)는 독점 하드웨어에 의존하며, 종양화된 병목 지점을 형성한다. 스펙터(Spectre), 멜트다운(Meltdown), 플런더볼트(Plundervolt)와 같은 취약점은 반복적으로 TEE의 무결성을 훼손해왔다.
- 신뢰 가정: TEE는 사용자가 하드웨어 제조사와 제3자 검증(attestation) 서비스 모두를 신뢰해야 하므로, 상당한 신뢰 가정과 종양화 문제를 초래한다.
- 성능 한계: 분산 시스템 전반에 걸쳐 TEE를 확장할 경우 지연과 조정 오버헤드가 발생하여, 실시간 고처리량 애플리케이션에는 비현실적이다.

또 다른 암호학적 접근 방식인 완전 동형 암호(FHE, Fully Homomorphic Encryption)는 데이터를 복호화하지 않고도 연산을 가능하게 한다. 그러나 FHE는 평균 연산 대비 최대 1,000배에 달하는 계산 오버헤드를 요구하며, 이로 인해 대부분의 애플리케이션에서 실용성이 떨어진다.

아рки움은 다자간 연산(MPC)을 통해 탈중앙화 기밀 컴퓨팅을 개척함으로써 이러한 한계를 해결한다. TEE나 FHE와 달리, 아рки움의 MPC 기반 프레임워크는 다음과 같은 특성을 가진다:

- 하드웨어 의존성 제거: 연산은 취약한 하드웨어 엔클레이브가 아니라 암호학적으로 보호된다.
- 신뢰 분산: 단일 주체가 연산을 통제하지 않는다. MPC 프로토콜은 데이터를 여러 노드에 분산시켜 어느 한 참여자도 평균 데이터에 접근할 수 없도록 한다.
- 평균 수준 속도 달성: 아рки움 클러스터의 병렬 실행은 FHE의 과도한 지연과 달리 거의 네이티브 수준의 성능을 구현한다.

아рки움의 아키텍처는 MPC를 기존의 제한적 활용 사례를 넘어 확장시킨다:

- 병렬 실행: MXE(Multi-Party eXecution Environments)는 독립 상태를 가진 원자적 병렬 연산을 가능하게 하며, 글로벌 합의의 병목을 우회한다.
- 구성 가능한 신뢰 모델: 개발자는 맞춤형 MPC 프로토콜을 활용하여 속도, 보안성, 탈중앙화 간 균형을 조정할 수 있다. 예를 들어, 의료 애플리케이션은 BFT를 우선시할 수 있으며, 게임은 보다 경량화된 프로토콜을 사용할 수 있다.
- 체인 불가시론적 암호화: arxOS는 어떠한 블록체인 또는 오프체인 시스템과도 통합될 수 있으며, 분절된 생태계 전반의 데이터 흐름을 암호화할 수 있는 역량을 가진다.

이처럼 하드웨어 의존적 TEE에서 암호학 기반 MPC로의 전환은 기밀 컴퓨팅을 민주화한다. 금융 기관은 암호화된 데이터셋 위에서 리스크 분석을 수행할 수 있고, AI 모델은 최초 입력값을 노출하지 않은 채 동적 기반 기업 데이터셋 위에서 학습될 수 있으며, 탈중앙화 자율 조직(DAO)은 안전하게 투표를 진행할 수 있다. 이 모든 과정은 종양화된 하드웨어 벤더에 의존하지 않는다. 런타임 데이터라는 마지막 프론티어를 암호화함으로써, 아рки움은 인터넷 자체를 기밀성 우선 인프라로 전환한다.

2.2 아рки움의 비전

아르키움은 데이터 기밀성이 디지털 인프라의 구조 자체에 자연스럽게 통합되어, 산업 전반에서 기밀성을 훼손하지 않으면서도 신뢰 불필요하고 고성능의 연산이 가능해지는 미래를 지향한다. 중앙화된 중개자와 취약한 하드웨어 엔클레이브를 탈중앙화 MPC 및 투명하고 스테이킹 기반으로 정렬된 암호경제적 인센티브 구조로 대체함으로써, 아르키움은 암호화된 연산을 제한된 도구에서 보편적 유틸리티로 전환한다. 네트워크는 고정 공급량 ARX 토큰 모델을 통해 노드 운영자, 위임자, 사용자 간 인센티브를 정렬하면서도, 금융 기록, 의료 분석, AI 모델 등 민감한 데이터를 암호학적 보장에 처리할 수 있도록 개발자, 기업, 개인 모두에게 역량을 제공한다.

사용 중 데이터 보안을 넘어, 아르키움은 Web2 수준의 연산 성능과 Web3의 무허가 혁신을 연결한다. 아르키움 네트워크는 암호학적 복잡성을 구성 가능한 MXE로 추상화하고 최신 블록체인 인프라(예: 솔라나)와 통합함으로써, 개발자가 기밀성 우선 애플리케이션을 구축하기 위한 진입 장벽을 낮춘다. 이러한 비전은 궁극적으로 수요에 따라 연산 능력이 동적으로 확장되고, 민감한 작업이 체인·관할권·산업 전반에 걸쳐 병렬적으로 실행되며, 단 한 바이트의 평문도 노출되지 않는 글로벌 분산형 자생 생태계인 인코딩된 슈퍼컴퓨터로 귀결된다.

3 아키텍처 개요

3.1 핵심 구성 요소

3.1.1 Arx 노드(Arx Nodes)

Arx 노드는 집단적 MPC 프로세스에 기여하는 개별 연산 제공자이다. 노드는 클러스터를 통해 여러 연산에 동시에 참여한다. Arx 노드가 참여하는 각 프로세스에서, 해당 노드는 전체 연산 데이터의 일부 조각—특정 암호화 데이터에 대한 키셰어(keyshare)—를 보유한다. 이후 노드들은 서로의 데이터 조각을 공개하지 않은 상태에서 이러한 조각들 위에서 협력적으로 연산을 수행한다. 이를 통해 데이터의 기밀성과 보안이 보장되면서도, 네트워크는 전체 데이터셋에 대한 함수를 계산할 수 있다. 노드들은 안전한 MPC 프로토콜을 통해 연산 결과를 공유하며, 이를 통해 어떤 단일 노드도 원본 전체 데이터셋에 접근하지 못한 상태에서 공개적으로 암호학적으로 검증 가능한 최종 결과를 도출할 수 있다.

3.1.2 클러스터(Clusters)

아르키움 네트워크에서 클러스터는 집단적으로 연산을 수행하는 Arx 노드들의 그룹을 의미한다. 클러스터는 연산 고객(섹션 4.2 참조)에 의해 생성되며, 노드 집합은 네트워크 내 신뢰성에 대한 과거 평판, 사용 가능한 연산 자원 용량 및 기타 요소를 기준으로 선정된다. 또한 하나 이상의 노드가 사용할 수 없게 되거나 신뢰할 수 없는 것으로 판단될 경우, 클러스터는 새로운 노드로 교체되는 마이그레이션 과정을 거칠 수 있다.

3.1.3 다자간 실행 환경(MXE, Multi-Party eXecution Environments)

MXE는 안전한 연산 실행을 위한 격리된 전용 가상 환경이다. 이는 데이터 제공 및 처리 방식, 암호화 스킴 등의 세부 구성을 미세 조정할 수 있을 정도로 높은 구성 가능성을 제공한다. 연산 고객에 의해 생성되는 MXE는 MPC 프로토콜에 종속되지 않으며, 클러스터 위에서 동작한다. 특정 클러스터 위에서 실행될 수도 있고, 가용성에 따라 사전 선택된 여러 클러스터 사이를 순환하며 실행될 수도 있다. MXE는 '지속형(persistent)'과 '단일 사용(single-use)'으로 구분된다. 지속형 MXE는 미래에도 무기한 재사용이 가능하며, 단일 사용 MXE는 하나의 연산에만 사용된 뒤 자동으로 폐기된다.

3.1.4 연산(Computations)

아르키움 네트워크에서 연산은 MXE에 의해 수행되는 독립적이고 개별적인 작업을 의미한다. 연산은 먼저 연산 고객에 의해 '정의' 형태로 생성되며, 여기에는 연관된 로직 회로 정의(logic circuit definition), 실행 권한이 부여된 주체(예: 개인, 온체인 컨트랙트 등), 입력 및 출력 데이터 처리 설정, 콜백 설정(예: 미래 연산 예약, 다른 온체인 프로그램의 함수 실행 등), 그리고 기타 속성이 포함된다. 각 연산 정의는 반드시 하나의 MXE와 연결되어야 한다. 실제 개별 연산 호출은 항상 사전에 설정된 연산 정의를 기반으로 수행된다.

3.1.5 온체인 컨트랙트(On-Chain Contracts)

아르키움 네트워크의 모든 오케스트레이션은 블록체인을 통해 처리된다. 스마트 컨트랙트 프로그램(예: Solana 상 프로그램)은 온체인 멤풀(memppool) 구성 및 연산 순서 지정, 보상 계산 및 분배, 토큰 스테이킹 및 슬래싱, 그리고 다양한 운영 역할을 수행한다. 네트워크 참여자 간 모든 상호작용은 아르키움 네트워크의 온체인 프로그램을 통해 조율된다. 아르키움 네트워크는 기존 블록체인 인프라(예: Solana)를 합의 계층으로 활용함으로써, 상태 및 합의 관리를 위한 독립적인 레이어1 솔루션의 필요성을 제거한다. 이러한 아키텍처는 합의를 블록체인에 위임함으로써 아르키움 네트워크 내부의 완전한 병렬화를 가능하게 하며, 네트워크 오케스트레이션 작업이 효율적으로 확장될 수 있도록 한다.

Arx 노드는 오프체인 작업자로 이해할 수 있으며, 온체인 아르키움 네트워크 프로그램은 조직 허브 역할을 수행한다. 이는 온체인 멤풀에 연산 작업을 수집하고, 이를 실행을 위해 Arx 노드 클러스터에 할당하며, 이후 결과 및 보상 처리를 담당한다.

3.1.6 연산 단위(CU, Computation Units)

연산 단위(CU)는 네트워크 전반에서 고정된 양의 연산 작업을 나타내는 상수이다. CU는 아르키움 네트워크에서 가장 작은 연산 단위이며, 네트워크 내 모든 산술 연산(연산을 구성하는 요소들)은 CU 기준으로 측정된다.

3.1.7 에폭(Epochs)

아르키움 네트워크의 시간은 에폭(Epoch)을 통해 추적 및 관리된다. 에폭은 일정한 길이를 가지며, 작업 실행 및 스케줄링, Arx 노드 보상 분배, 스테이킹 토큰의 락업 기간 등 다양한 기능의 시간 기준으로 활용된다.

3.2 다자간 연산(MPC) 프로토콜

초기 아르키움 네트워크는 케르베로스(Cerberus)와 맨티코어(Manticore)라는 두 가지 MPC 프로토콜 백엔드를 지원한다. 이 두 백엔드는 서로 다른 트레이드오프를 가지며, 보다 다양한 애플리케이션을 가능하게 한다. 아르키움 네트워크는 확장 가능하고 프로토콜 불가지론적(protocol-agnostic)으로 설계되어 있으며, 향후 추가적인 MPC 프로토콜이 지원될 예정이다.

예를 들어, 맨티코어는 연산 집약적인 작업에 적합하기 때문에 머신러닝 모델 학습에 활용될 수 있으며, 이후 새롭게 학습된 모델을 활용한 예측 작업은 케르베로스를 통해 수행할 수 있다.

케르베로스와 맨티코어 프로토콜은 Arcis로 작성된 회로를 네이티브로 실행하며, 효율적인 시크릿 셰어링(secret-sharing) 및 검증 가능한 출력 생성을 위해 Arcis의 타입 시스템을 활용한다. 개발자는 Arcis로 작성된 회로를 사용해 애플리케이션 요구사항(비잔틴 복원력 vs. 정직하지만 호기심 많은(honest-but-curious) 가정 등)에 따라 프로토콜을 유연하게 전환할 수 있다.

3.2.1 케르베로스

케르베로스는 아르키움의 핵심 MPC 연산 백엔드이다. 각 셰어(share)는 메시지 인증 코드(MAC, Message Authentication Code)로 인증되며, 이를 통해 정직한 노드는 무결성을 검증하고 변조를 감지할 수 있다. 악의적 행위가 탐지될 경우 연산은 즉시 중단된다. 특정 클러스터 내 최소 하나의 노드가 정직하다는 가정하에, 연산 결과는 올바른 것으로 보장된다. 케르베로스는 현재 지원되는 두 가지 MPC 프로토콜 중 가장 강력한 보안 보장을 제공한다.

3.2.2 맨티코어

맨티코어는 “정직하지만 호기심 많은(honest but curious)” 환경에서 동작한다. 또한 온라인 단계(online phase)에 필요한 전처리 데이터를 생성한 뒤 오프라인 상태로 전환되는 “신뢰된 딜러(trusted dealer)”에 의존한다. 이러한 구조는 실행 속도를 향상시키며, 머신러닝 및 인공지능(AI) 애플리케이션에 필요한 복잡한 연산을 가능하게 한다. 그러나 이러한 속도 향상은 상대적으로 약한 보안 보장을 대가로 한다. 맨티코어는 특정 클러스터의 노드 운영자를 신뢰할 수 있고, 정직하게 행동할 경제적 인센티브가 존재하는 환경을 위해 설계되었다.

3.3 ArxOS: 분산 실행 엔진(Distributed Execution Engine)

ArxOS는 아르키움 네트워크의 실행 계층을 구성하는 Arx 노드, 클러스터, MXE를 집합적으로 지칭하는 개념적 용어이다. 이 용어는 연산 자원의 탈중앙화된 조정을 설명하며, 여기서:

- Arx 노드는 분산 처리 코어 역할을 수행한다.
- 클러스터는 고유한 설정과 역량을 가진 동적으로 구성되는 연산 그룹으로 작동한다.
- MXE는 암호화된 실행 환경을 정의한다.

ArxOS는 독립된 시스템이 아니라, 이러한 구성 요소들이 어떻게 상호작용하여 기밀성 보존 연산을 구동하는지를 설명하기 위한 개념적 프레임워크이다. 이는 MPC 프로토콜의 오케스트레이션, 자원 할당, 병렬화된 작업 실행을 추상화하며, 전통적 컴퓨터의 운영체제가 수행하는 역할을 탈중앙화되고 신뢰 불필요한 방식으로 구현한 것에 해당한다.

4 네트워크 참여자 유형

아르키움 네트워크는 서비스를 획득하거나 제공하기 위해 경제적으로 네트워크와 상호작용하는 세 가지 유형의 참여자로 구성된다.

4.1 Arx 노드 운영자(Arx Node Operators)

Arx 운영자는 아르키움 네트워크의 핵심 서비스 제공자로, 물리적 하드웨어 인프라 위에서 Arx MPC 노드 소프트웨어를 운영함으로써 연산 실행에 참여하고, 완료한 연산 작업에 비례하는 보상을 획득한다. Arx 노드를 실행할 때 운영자는 자신의 노드가 네트워크에 제공할 수 있는 연산 자원의 규모를 지정한다. 그러나 이러한 하드웨어 자원이 실제 작업 수행 자격을 얻기 위해서는, 해당 Arx 노드에 스테이킹된 총 담보(collateral)가 연산 자원 규모에 비례해야 한다. 운영자는 또한 자신의 Arx 노드 구성에서 제공 가능한 보안 기능(예: TEE 지원 여부)에 대한 세부 정보를 명시한다. 사용 가능한 연산 자원 및 노드 기능 외에도, 연산 고객은 과거 평판을 기준으로 Arx 노드(클러스터 내)를 선택한다. 따라서 Arx 운영자는 지속적인 연산 작업 흐름을 확보하기 위해 최대한 신뢰성 높은 서비스를 설계·운영하고, 평판을 유지하는 데 집중한다. 평판은 과거 슬래싱 기록, 완료된 연산 통계 등을 기준으로 연산 고객이 오프체인에서 평가한다. 사업을 더욱 확장하기 위해 운영자는 네트워크에 제공하는 인프라 자원의 규모를 늘릴 수 있다.

Arx 노드 운영자는 자신이 수행한 연산 작업에 대해 두 가지 형태의 보상을 받는다: 자기 위임 용량에 대응하는 프로토콜 수준 보상과, 제3자 위임자에게 제공하는 하드웨어 자원 활성화 서비스에 대한 운영 수수료이다. 필수적인 최소 자기 위임 스테이킹은 필요 시 계획된 마이그레이션 비용을 충당하는 데 사용된다.

4.2 연산 고객(Computation Customers)

네트워크의 수요 측(buy-side) 참여자인 연산 고객은 클러스터에서 운영되는 Arx 노드로부터 맞춤형 기밀 컴퓨팅 서비스를 구매한다. 연산 서비스를 이용하기 위해 연산 고객은 먼저 MXE를 생성해야 하며, 이후 자신이 수행하려는 연산의 회로와 세부 내용을 담은 연산 정의를 생성해야 한다. 이후 고객은 사전에 정의된 연산 유형에 따라 연산 실행을 요청할 수 있으며, 이때 연산 수수료를 지불해야 한다. 해당 수수료는 네트워크가 정의한 기본 비용과 고객이 직접 설정하는 우선순위 수수료로 구성되며, 이를 통해 고객은 추가 비용을 지불하고 자신의 연산을 우선 처리되도록 할 수 있다.

4.3 제3자 위임자(Third-Party Delegators)

각 제3자 위임자는 아arki움 네트워크에서 운영되는 Arx 노드에 자신의 스테이킹을 위임함으로써, 해당 Arx 노드가 연산 수행을 통해 획득한 보상의 비례 지분(pro-rata share)을 수령한다. Arx 운영자는 실제 물리적 인프라 운영 비용을 부담하므로, 운영자가 부과하는 비율 기반 수수료(rate-based fee)가 제3자 위임자의 보상에서 차감된다. 제3자 스테이킹은 Arx 운영자가 자신의 하드웨어를 활성화하여 작업 수행 자격을 획득할 수 있도록 지원한다. 또한 제3자 스테이킹은 부정행위 발생 시 슬래싱 대상이 되므로, 제3자 위임자는 Arx 운영자와 동일한 담보 리스크를 부담한다.

Arx 운영자는 선택적으로 제3자 위임을 비활성화할 수 있으며, 이 경우 자신의 노드에는 자기 위임만 허용된다. 이러한 기능의 사용 사례는 제한적일 수 있으나, 일부 노드 운영자는 제3자 위임 수수료를 통한 추가 수익보다, 단일 대규모 자기 위임을 선호할 수 있다(예: 슬래싱 책임을 자기 보관(self-custodied) 담보에만 한정하려는 대형 기관). 이 경우 운영자는 자신의 자기 위임만으로도 전체 하드웨어 자원을 충분히 활성화할 수 있기 때문에, 제3자 위임을 제한하는 편을 선호할 수 있다.

5 ARX 토크노믹스

5.1 ARX 토큰

ARX 토큰은 아arki움 네트워크의 유틸리티 토큰이다. 해당 토큰은 제네시스 시점에 고정 총 공급량으로 설정되며, 출시 이후 프로토콜 차원의 추가 발행이나 소각 메커니즘은 존재하지 않는다. ARX에 대한 수요는 네트워크 사용량과 직접적으로 연결된다. 노드 운영자와 제3자 위임자는 Arx 노드 하드웨어를 활성화하기 위해 ARX를 스테이킹해야 하며, 이를 통해 토큰 유틸리티와 네트워크 전체 연산 처리량이 정렬된다.

Arx 노드에 대한 보상은 전적으로 연산 고객의 지불에서 발생한다. 연산 고객은 기본 가격과 고객 정의 우선순위 수수료를 체인 네이티브 토큰으로 지불한다. 프로토콜은 노드 운영 보조를 위해 추가 ARX를 발행하지 않으며, 우선순위 수수료 흐름에서 ARX를 소각하지도 않는다. 모든 우선순위 수수료는 해당 연산을 수행한 Arx 노드에 전액 귀속된다.

5.2 노드 하드웨어 클레임

아arki움 네트워크의 각 Arx 노드는 생성 시 공개적으로 하드웨어 클레임을 선언해야 한다. 이 클레임은 해당 노드가 최대 부하 상태에서 단위 시간당 처리 가능한 연산 단위의 수로 정의된다. Arx 노드는 정직한 클레임을 제출하도록 인센티브가 설계되어 있다. 클레임을 과소평가할 경우 노드는 더 적은 연산 작업(및 따라서 더 적은 보상)을 받게 되며, 반대로 과대평가할 경우 지나치게 큰 연산 작업 수행 시 속도가 부족하거나 하드웨어 한계로 인해 충돌이 발생할 수 있으며, 이 경우 비참여로 간주되어 슬래싱 대상이 될 수 있다.

Arx 노드는 출시 이후 자유롭게 하드웨어 클레임을 상향 조정할 수 있다. 그러나 하드웨어 클레임을 하향 조정하는 것은 다음 두 경우에만 허용된다: (a) 현재 어떤 클러스터에도 속하지 않는 경우, 또는 (b) 자신이 속한 클러스터 중 하드웨어 클레임 감소로 인해 클러스터 전체 성능이 요구 최대 연산 부하 아래로 떨어질 경우, 해당 클러스터의 마이그레이션 비용을 자기 위임 스테이크로 부담할 의사가 있는 경우.

5.3 스테이킹 및 하드웨어 활성화

각 노드 운영자는 노드별 클레임을 제출해야 하지만, 이것만으로는 해당 연산 자원이 네트워크에서 사용 가능 상태가 되지 않는다. 대신 Arx 노드에 ARX 토큰 위임이 이루어져야 연산 자원이 활성화된다. 노드가 더 많은 스테이킹을 받을수록 더 많은 하드웨어가 활성화되며, 네트워크 작업 수행 자격을 얻게 된다. 충분하지 않은 스테이킹은 Arx 노드가 전체 연산 자원을 활용하지 못하도록 만든다.

Arx 노드 하드웨어 활성화를 계산하기 위해, 네트워크 전반에는 연산 부하 단위당 요구되는 스테이킹 규모를 나타내는 고정 비율 상수가 존재한다. 이 상수는 REQUIRED_STAKE_PER_UNIT_OF_CU_LOAD_CAPACITY라 불린다. 해당 비율은 노드의 하드웨어 클레임에 곱해져, 노드의 클레임을 증명하기 위해 필요한 총 위임 규모를 계산한다. 하드웨어 클레임은 노드가 최대 사용 상태에서 처리 가능한 최대 연산 부하를 정의하므로, 만약 노드가 선언된 하드웨어 용량에 비례하는 총 위임 규모보다 적은 스테이킹만 받았다면, 해당 노드는 최대 성능 수준까지 연산 작업을 배정받지 못한다(즉, 위임된 스테이킹 규모에 비례한 수준까지만 작업을 수행할 수 있다).

특정 노드에 대한 총 위임 규모가 `REQUIRED_STAKE_PER_UNIT_OF_CU_LOAD_CAPACITY` 비율을 초과할 경우, 노드 운영자는 하드웨어를 업그레이드해야 한다(이를 통해 더 높은 하드웨어 클레임에 비례한 추가 위임을 수용 가능). 그렇지 않을 경우, 추가 위임에 대한 보상은 거의 선형적인 감쇠 곡선을 따르게 되며, 이는 하드웨어 용량을 초과하는 과도한 위임에 대해 강력한 비인센티브로 작동한다.

아르키움 네트워크의 스테이킹 구조는 중앙화를 억제한다. Arx 노드가 제공하는 하드웨어 자원과 이를 활성화하기 위해 요구되는 스테이킹 규모 사이에 거의 선형적인 관계를 형성함으로써, 가장 인기 있거나 평판이 높은 노드라 하더라도 자신이 네트워크에 제공하는 실질적 유틸리티(연산 자원)에 비례하는 수준 이상의 스테이킹을 받을 수 없도록 설계되어 있다.

5.4 스테이킹 위임 메커니즘

Arx 노드에 위임되는 총 스테이킹은 운영자의 자기 위임과 ARX 토큰 보유자의 제3자 위임으로 구성된다. 각 노드는 최소 자기 위임 규모를 반드시 유지해야 하며, 이는 필요 시 마이그레이션 비용을 충당하기 위해 사용된다. 자기 위임 스테이킹과 제3자 스테이킹 모두 노드 하드웨어 클레임 활성화 기능을 수행하지만, 운영자의 자기 위임은 또한 자신의 서비스 제공 능력에 대한 신뢰를 나타내는 역할도 한다. 게임이론적 관점에서 이는 운영자가 네트워크에 직접적인 이해관계를 가지고 있음을 보여주는 신호로 작용한다.

반면, 제3자 위임자는 직접 노드 인프라를 운영하지는 않지만, 자신이 신뢰할 수 있다고 판단한 Arx 노드에 ARX 자본을 위임함으로써 네트워크에 기여한다. 노드 운영자와 마찬가지로, 제3자 위임자 역시 노드 부정행위 발생 시 자신의 자본이 슬래싱될 위험을 부담한다. 따라서 제3자 위임자는 노드에 대한 지속적인 실사를 수행할 인센티브를 가진다. 이 과정에서 온체인 및 오프체인 평판을 검토하고, 필요 시 다른 노드로 스테이킹을 재위임할 수 있다.

5.5 슬래싱

Arx 노드의 부정행위 발생 시, 운영자의 자기 위임을 포함한 모든 위임 스테이킹은 동일한 비율로 슬래싱된다. 슬래싱은 부정행위에 대한 강력한 억제 장치 역할을 하며, 동시에 실패한 연산과 관련된 비용(즉, 정상적으로 작업을 수행한 다른 노드에 대한 보상)을 충당하는 메커니즘을 제공한다.

5.5.1 슬래싱 유형

아르키움 네트워크에서 Arx 노드에 대한 처벌은 크게 두 가지 유형으로 나뉜다: 비참여(non-participation, 예: 다운타임 또는 의도적 무응답)와 치팅(cheating, 잘못된 연산 결과 제공)이다. 두 경우 모두 슬래싱으로 처벌되지만, 비참여는 치팅보다 더 무거운 처벌을 받는다. 이는 비참여를 탐지하는 비용이 치팅 탐지 비용보다 더 크기 때문이다. 치팅은 암호학적으로 탐지되지만, 비참여 탐지는 종종 더 넓은 범위의 Arx 노드(해당 연산 클러스터 외부 노드 포함)를 동원하여 비참여 주장(withholding claim)을 검증하는 합의 프로토콜을 실행해야 하므로 추가적인 비용이 발생한다.

슬래싱 패널티는 비율 기반으로 적용된다. 즉, 슬래싱 규모는 중단된 연산의 크기에 비례한다. 따라서 더 높은 하드웨어 사양을 가진 노드(그리고 이에 따라 더 많은 스테이킹이 위임된 노드)는 더 큰 슬래싱 패널티를 받을 수 있다. 이는 더 많은 스테이킹이 더 큰 하드웨어 활성화에 필요하기 때문이며, 결과적으로 증가된 스테이킹은 증가된 슬래싱 리스크에 대한 비례 담보 역할을 수행한다.

5.5.2 탐지 및 분쟁 해결

비참여는 치팅과 달리 암호학적으로 탐지될 수 없기 때문에, 아르키움 네트워크는 보다 넓은 범위의 Arx 노드 합의를 통해 비참여 분쟁 해결 메커니즘을 수행한다. 프로토콜은 어떤 참여자든 다른 참여자를 비참여로 고발할 수 있도록 허용한다. 이에 대해 피고발자는 자신의 참여를 증명하기 위해 특정 메시지를 클러스터 내 모든 피어(peer)에게 전송해야 하며, 이 피어들은 릴레이(relay) 역할을 수행한다. 초기 검증이 실패할 경우, 해당 고발은 원래 클러스터 외부의 더 넓은 Arx 노드 집합으로 확대될 수 있으며, 이를 통해 보다 광범위한 검증과 투명성을 보장한다. 이러한 전략은 검열 또는 방해 가능성을 제거할 뿐 아니라, 노드 인센티브를 네트워크 건강성과 정렬시켜 안정적이고 신뢰 가능한 연산 환경을 촉진한다.

5.5.3 치팅 탐지

또 다른 형태의 부정행위는 참여자 간 잘못된 데이터를 전송하는 것이다. 이러한 경우 수신 피어는 유효하지 않은 데이터를 탐지하고 프로토콜 중단을 트리거한다. 예를 들어 고전적인 BDOZ MPC 프로토콜에서는 정당한 중단과 악의적 중단을 구분할 수 없기 때문에, 노드가 특정 연산을 사실상 아무런 불이익 없이 선택적으로 검열할 가능성이 존재한다. 다행히도 이러한 프로토콜은 치팅 시도가 다른 피어 및 제3자에게 암호학적으로 증명될 수 있도록 확장 가능하다. 아르키움 네트워크의 케르베로스 프로토콜은 이러한 개선을 포함하며, 여기에 슬래싱 메커니즘과 이후 다른 클러스터가 해당 연산을 이어받을 수 있을 수 있다는 구조가 결합됨으로써, 연산 검열을 시도할 인센티브는 사실상 제거된다.

6 경제 메커니즘

6.1 연산 가격 체계

MPC 작업 연산 고객이 연산 수행을 위해 지불하는 비용은 기본 가격과 우선순위 수수료의 두 부분으로 구성된다. 이 두 비용은 모두 해당 연산을 수행하는 클러스터 내 모든 노드에 균등하게 분배된다. 기본 가격은 스테이킹 가중치 노드 운영자 투표를 통해 결정되며, 최소한 노드 운영자가 연산 참여로 인해 발생하는 운영 비용을 충당할 수 있도록 하는 최소 비용 보장 수준을 의미한다. 반면 우선순위 수수료는 자유 시장에 의해 결정되며, 연산 실행 순서를 좌우한다. 더 높은 우선순위 수수료를 설정할수록 연산 실행을 가속화할 수 있다.

Arx 토큰은 Arx 노드 스테이킹에 사용되지만, 실제 연산 비용(기본 가격 및 우선순위 수수료)은 체인 네이티브 토큰(예: 솔라나의 SOL)으로 지불된다. 향후 아크키움의 멀티체인 구조를 지원하게 되면, 노드 운영자는 여러 체인의 네이티브 토큰으로 연산 수익을 획득하게 되며, 각 체인은 고유하고 동적인 환산 비율을 가진다.

6.1.1 기본 가격

실행 비용은 항상 최소한 경제적으로 지속 가능한 수준으로 유지되어야 한다. 이는 Arx 노드의 비참여가 허용되지 않기 때문이다. 프로토콜에 의해 설정되는 기본 가격은 모든 시장 상황에서 Arx 노드 운영 비용을 보장하기 위한 기준선 역할을 수행한다. 기본 가격 자체는 연산 단위 기준의 단가이며, 실제 연산 크기에 따라 동일한 기본 가격이라도 최종 비용은 달라진다.

기본 가격이 존재하지 않는다면, 특정 시점에서 네트워크 내 연산 수요가 부족할 경우 Arx 노드는 연산에 대한 충분한 보상을 받지 못할 수 있다. 솔라나와 같은 전형적인 자유시장형 탈중앙화 네트워크와 달리, 아크키움의 Arx 노드는 수요 감소로 인해 수수료가 낮아졌다고 해서 단순히 참여를 중단할 수 없다. 이는 비참여로 간주되어 슬래싱 대상이 되기 때문이다. 또한 아크키움 네트워크에 진입하거나 이탈하는 비용은 상당하다. 이는 락업 기간과 자기 위임 스테이킹에서 마이그레이션 비용을 부담해야 하는 구조 때문이다. 따라서 아크키움은 수요 수준과 무관하게 노드 운영 비용을 보장하기 위해 중앙 계획형 접근 방식을 기본 가격 체계에 적용한다.

기본 가격(CU당 가격)은 새로운 에폭 시작 전에 설정된다. 노드 운영자는 다음 에폭에서의 CU 가격에 대해 자기 위임 스테이킹을 기반으로 한 가중치 투표를 수행한다. 현재 에폭 동안 언제든지 다음 에폭 가격에 대한 투표를 제출할 수 있으며, 기권은 현재 가격에 찬성하는 것으로 간주된다. 최종적으로 다음 에폭의 CU 가격은 스테이킹 가중치 평균값으로 결정되며, 기권 스테이킹 역시 기존 가격에 대한 찬성표로 계산된다.

이 투표에는 오직 자기 위임된 노드 운영자 스테이킹만 참여할 수 있다. 또한 자기 위임 스테이킹이 노드 전체 하드웨어를 활성화하는 데 필요한 수준 이하인 경우에만 전액이 반영된다. 만약 자기 위임 규모가 노드 하드웨어 전체 활성화에 필요한 스테이킹 규모를 초과한다면(제3자 위임 제외 기준), 해당 한도 이하 부분만 기본 가격 투표에 반영된다.

제3자 위임자는 이 투표에 참여할 수 없다. 게임이론적 관점에서 볼 때, 실제 운영 비용을 직접 부담하는 것은 노드 운영자뿐이므로, 가격을 정확하게 설정할 진정한 이해관계자를 가진 주체 역시 운영자이기 때문이다. 반면 제3자 위임자는 직접 인프라 비용을 부담하지 않으며(간접적으로는 운영 수수료를 통해 부담), 따라서 단기적 이익을 위해 과도하게 높은 기본 가격을 선호할 가능성이 있다. 노드 운영자는 지속 가능하고 경쟁력 있는 가격을 설정할 유인을 가진다. 가격이 너무 높으면 연산 수요 및 수익이 감소하고, 너무 낮으면 연산당 수익이 감소하여 결국 노드 운영 자체가 지속 불가능해질 수 있기 때문이다. 따라서 균형 잡힌 기본 가격은 플루토크라틱(plutocratic) 투표를 통해 자유롭게 형성된다.

락업 기간은 Arx 노드가 연산 수익에 대해 장기적 관점을 가지도록 만든다. 따라서 단기 운영 비용 변동이나 작업량 변화는 큰 영향을 미치지 못한다. 또한 비참여에 대한 슬래싱 패널티와 네트워크 이탈 비용은 시장 변동성으로 인해 일시적으로 연산 보상이 부족해지더라도 노드가 네트워크를 떠나거나 비참여 상태로 전환하는 것을 억제한다.

Arx 노드는 키 관리 리스크를 줄이기 위해 별도의 서브키(subkey)를 사용하여 기본 가격 투표를 수행한다. 이는 투표 키가 일반적으로 노드 상의 핫 스토리지에 저장되기 때문이다. 노드는 외부 가격 오라클과 연결된 맞춤형 자동 투표 스크립트를 운영하는 등의 방식을 선택할 수 있다.

6.1.2 우선순위 수수료 시장

Arx 노드의 연산 수익은 고객 정의 우선순위 수수료와 기본 가격 수익의 조합으로 구성된다. 서로 다른 우선순위 수수료 값은 독립적인 수수료 시장을 형성하며, 자유시장 수요와 공급이 더 빠른 연산 실행을 위한 프리미엄 수준을 결정한다. 클러스터 간 연합에 따라 분리된 수수료 시장이 형성되는데, 이는 특정 연산이 같은 클러스터를 공유하는 다른 연산들과만 우선순위를 경쟁하는 것이 아니라, 동일한 노드를 일부 공유하는 다른 클러스터의 연산들과도 경쟁하기 때문이다. 여러 클러스터에 속한 Arx 노드는 제한된 연산 자원을 공유하므로, 서로 다른 클러스터 간 수수료 시장을 연결하는 역할을 한다.

아크키움 네트워크의 온체인 멤버는 이러한 상호 연결된 클러스터 연합(공동 노드를 공유하는 클러스터 집합)을 기준으로 우선순위 큐로 분리되어 구성된다. 이론적으로는 연산량이 매우 높고 다수의 클러스터가 동시에 연산을 수행하는 경우, 네트워크 전체가 하나의 거대한 연합을 형성할 수도 있다. 그러나 이 메커니즘은 수수료 시장을 더 작은 단위로 분리할 수 있을 때 항상 그렇게 되도록 보장한다.

만약 연산 고객이 특정 클러스터의 모든 자원을 하나의 대규모 연산에 즉시 사용하고자 한다면(해당 클러스터 내 이미 진행 중인 연산 완료 이후 즉시 실행), 고객은 동일한 Arx 노드 집합을 공유하는 모든 재귀적 클러스터 연합 내 다른 연산보다 더 높은 우선순위 수수료를 설정해야 한다. 또한 실제 연산 실행 순서는 우선순위 수수료만을 기준으로 결정된다. 따라서 특정 클러스터의 일부 Arx 노드가 이미 더 높은 우선순위 연산으로 인해 포화 상태에 있거나, 스테이킹 부족으로 전체 하드웨어를 활용하지 못하는 경우, 다른 연산은 단순히 실행을 기다려야 한다.

6.2 수익 분배

6.2.1 노드 보상 분배

특정 연산에 대한 보상은 해당 연산이 실행된 클러스터 내 Arx 노드들 사이에 균등하게 분배된다. 클러스터 단위 보상 분배는 각 클러스터가 수행한 연산 수에 전적으로 기반하며, 클러스터 내부에서는 완전한 평등주의적 방식으로 보상이 분배된다.

6.2.2 서브노드 위임 보상 분배

Arx 노드가 수령한 보상은 해당 노드에 스테이킹을 위임한 제3자 위임자들에게 비례 배분된다. 배분 비율은 각 위임자의 스테이킹 규모에 따라 결정된다. 이후 각 노드별로 정의된 고정 비율의 운영 수수료가 위임자 보상에서 차감되어 노드 운영자에게 지급된다.

만약 특정 Arx 노드에 이미 모든 하드웨어가 활성화되어 작업 수행 자격을 갖추었음에도 추가 스테이킹이 계속 위임되는 과잉 스테이킹이 존재할 경우, 노드 하드웨어 클레임을 충족하는 데 필요한 수준을 초과한 스테이킹에 대해서는 서브선형 함수 곡선이 적용된다. 결과적으로, 모든 하드웨어가 활성화된 이후에는 추가 스테이킹에 대한 수익률이 지속적으로 감소한다. 이는 노드에 위임된 스테이킹 규모와 노드가 제공하는 실제 연산 자원 규모 사이에 거의 선형적인 관계를 강제한다. 따라서 슬래싱이 발생할 경우, 항상 충분한 규모의 스테이킹 담보가 존재하게 된다. 또한 이러한 곡선 구조는 특정 노드에 대한 과도한 중앙집중형 스테이킹을 방지하여, 네트워크가 개방적이고 경쟁적인 상태를 유지하도록 만든다.

보상 분배는 아르키움 네트워크의 에폭 주기를 따른다. 즉, 각 에폭 종료 시점에 보상이 자동으로 복리 처리된다. 위임자는 보상을 해제하기 위해 일부 스테이킹을 위임 회수할 수 있다. 회수 요청 이후에는 한 개의 전체 에폭이 지나야 자금이 유동화되지만, 현재 에폭에서 새롭게 발생한 보상에 대해 회수를 요청한 경우에는 현재 에폭 종료 시점에 즉시 유동화된다(추가 1에폭 지연은 적용되지 않는다). 이후 스테이킹을 출금하여 보상을 수령할 수 있다.

위임자는 또한 재위임 기능을 통해 기존 위임 스테이킹을 다른 Arx 노드로 이동할 수 있다. 이는 단일 온체인 트랜잭션으로 수행되며, 한 개의 전체 에폭 이후 효력이 발생한다. 이는 기존 방식처럼 연위임 후 다시 위임하는 경우 발생하는 2에폭 지연을 줄여준다.

7 다자간 실행 환경(MXE, Multi-Party eXecution Environments)

7.1 실행 워크플로우

회로를 실행하기 위해서는 몇 가지 선행 조건이 충족되어야 한다.

먼저, 현재 MXE가 사용하는 클러스터의 피어들이 메모로부터 해당 연산을 처리할 수 있는 용량을 갖추고 있는지 확인되어야 한다. 이를 위해 필요한 전처리 값(preprocessing values)이 충분히 존재하는지 검증해야 하며, 만약 부족할 경우 해당 목적을 위해 새로운 값을 생성해야 한다. 실제 운영에서는 Arx 노드 소프트웨어가 항상 새로운 연산에 대비할 수 있도록 자동으로 전처리 프로토콜 라운드를 스케줄링하며, 이 과정에서 노드 내 유휴 연산 및 저장 자원을 최대한 활용한다.

이러한 조건이 충족되면 연산이 실행될 수 있다. 활성 클러스터의 Arx 노드들은 제공된 입력값을 가져와 회로에 주입한다. 이후 게이트는 결과가 도출될 때까지 노드들에 의해 병렬적으로 처리되며, 결과가 완성되면 해당 결과는 클러스터 내 모든 Arx 노드에 의해 공개 및 복구된다. 이후 각 노드는 결과에 서명하여 블록체인 프로그램에 제출하며, 이를 통해 다른 애플리케이션은 해당 결과를 신뢰하고 활용할 수 있게 된다.

7.2 연산 처리

연산은 MXE 내부에서 정의 및 스케줄링된다. 클러스터는 자신과 연결된 MXE를 위해 연산을 수행한다. 아르키움 네트워크 자체는 상태를 보유하지 않는 구조이므로, 모든 연산 실행은 완전한 병렬화가 가능하다. 하나의 클러스터는 동시에 여러 연산을 수행할 수 있으며, 유일한 병목 요소는 클러스터 내 가장 성능이 낮은 노드의 하드웨어 용량이다. 네트워크 전반의 병렬화는 네트워크가 공유 상태에 기반한 작업을 수행할 필요가 없다는 점에 의해 더욱 강화된다(많은 경우 개별 MXE조차도 공유 상태를 요구하지 않는다). 따라서 아르키움 네트워크는 임의의 런타임과 복잡도를 가진 연산을 병렬적으로 수행하는 고도로 병렬화된 블랙박스 이해할 수 있다.

MXE는 여러 클러스터를 사용하도록 구성될 수 있지만, 개별 연산 실행은 항상 MXE와 연결된 하나의 클러스터 내에서만 수행된다. 여러 클러스터를 사용하는 MXE는 특정 클러스터의 응답성에 전적으로 의존하지 않음으로써 가용성 보장을 향상시킬 수 있다. 예를 들어, 하나의 클러스터에서 연산이 실패할 경우 이미 연결된 다른 클러스터를 통해 즉시 재시도할 수 있으므로, 클러스터 비가용성에 보다 빠르게 대응할 수 있다.

7.3 MXE 암호화

아르키움 네트워크에서 데이터 암호화는 MXE 수준에서 처리된다. MXE를 생성할 때 연산 고객은 함께 사용할 암호화 스킴을 정의한다. MXE의 암호화 스킴은 MXE 연산 내부 데이터를 암호화하는 데 사용되는 암호화 키로 정의된다. 동일한 연산 고객이 소유한 여러 MXE 간에 동일한 암호화 스킴을 공유할 수도 있으며, 이를 통해 별도의 암호화 스킴을 반복 생성할 필요가 없어 약간의 비용 절감 효과를 얻을 수 있다. 특정 MXE 활용 사례에서 암호화가 필요하지 않은 경우에는, 암호화 스킴 파라미터를 선택적으로 none으로 설정할 수 있다.

7.4 사이드 채널 공격 저항성

TEE(Trusted Execution Environment)와 같은 일부 대안적 기밀 실행 플랫폼은 민감한 정보를 은폐하기 위해 하드웨어에 의존한다. 그러나 이러한 접근 방식은 등장 이후 다양한 유형의 사이드 채널 공격에 취약한 것으로 드러났으며, 이는 정보 추출 및 조작을 가능하게 함으로써 기밀성 보장을 무력화할 수 있다. 반면 BDOZ MPC 프로토콜은 계산적 및 정보 이론적 보안에만 의존한다. 따라서 보안 가정이 충족되는 한, 악의적인 피어 또는 다수의 피어라 하더라도 정보를 추출하거나 위조할 수 없다.

그러나 이론적으로 안전하더라도, 일부 불완전한 MPC 구현은 여전히 타이밍 공격에 취약할 수 있다. 이는 공격자가 데이터를 수신할 때 발생하는 미세한 지연 차이를 분석함으로써 다른 피어의 비공개 세어에 대한 정보를 추론하는 공격 방식이다. 특정 로컬 연산이 피연산자 값에 따라 더 많은 또는 더 적은 시간을 요구하기 때문이다. 이에 대한 해결책은 로컬 연산을 항상 상수 시간으로 수행하는 것이다. 이렇게 하면 해당 방식으로 정보를 추출하는 것이 불가능해진다. 따라서 아크리움 네트워크의 MPC 구현은 모두 상수 시간 방식으로 구현되어 있다.

8 클러스터

8.1 클러스터 구성

클러스터를 구성하는 Arx 노드 목록 외에도, 클러스터 생성 과정에서는 다양한 추가 속성이 정의된다. 클러스터의 최대 연산 부하(단일 대규모 연산 또는 다수의 소규모 연산 분산 처리 여부와 무관)는 해당 클러스터 내 가장 성능이 낮은 Arx 노드의 하드웨어 사양에 의해 암묵적으로 결정된다. 그러나 클러스터 생성 시에는, 클러스터가 처리해야 할 최대 연산 부하 또한 각 노드가 제공해야 하는 하드웨어 성능의 기대 최소 기준으로 명시적으로 정의되어야 한다.

클러스터의 대부분의 속성은 생성 이후 수정할 수 없다. 이는 클러스터가 공개적으로 사용 가능하며, 다른 MXE들이 해당 속성에 의존할 수 있기 때문이다. 다만 예외적으로 최대 연산 부하 제한은 클러스터 생성자에 의해 상향 조정될 수 있다. 단, 증가된 부하를 클러스터 내 모든 Arx 노드가 지원할 수 있어야 한다.

8.1.1 클러스터 할당 및 승인

클러스터 생성 시, 새 클러스터에 포함되는 각 Arx 노드는 해당 클러스터 참여를 개별적으로 승인해야 한다. Arx 노드가 클러스터 참여를 거부하는 이유는 다양할 수 있으며, 예를 들어 클러스터의 연산 요구사항이나 클러스터 내 다른 노드의 평판 또는 권한권 문제가 포함될 수 있다. 실제 운영에서는 Arx 노드들이 자신들이 함께 작업하기를 거부하는 Arx 노드 운영자 목록(오프체인 블랙리스트)을 유지하거나, 자신들이 참여 가능한 클러스터의 연산 요구사항 기준을 자체적으로 관리할 가능성이 높다. 이를 통해 클러스터 참여 요청에 대한 실시간 분석이 가능해진다.

비허가형 클러스터의 경우, 클러스터 활성화 이전에 네트워크에서 무작위로 선택된 하나의 Arx 노드가 클러스터 노드 집합에 추가된다. 이 무작위 노드는 연산 고객이 직접 지정하는 것이 아니라 독립적으로 선택된다. 클러스터에 할당된 모든 Arx 노드(무작위 노드 포함)가 참여를 승인하면 클러스터는 즉시 활성화된다. 그렇지 않을 경우 연산 고객은 최소 한 개의 전체 예폭이 지난 후 클러스터 생성 절차를 취소할 수 있다. 만약 단 하나의 Arx 노드라도 참여를 거부할 경우, 클러스터 생성은 실패한다.

이와 동일한 할당 및 거부 메커니즘은, 마이그레이션이 필요한 경우 기존 클러스터에 새로운 Arx 노드를 추가할 때에도 적용된다.

8.2 MXE 승인

클러스터가 MXE를 사용할 수 있도록 허용하는 과정(MXE 승인)에는 다양한 구성 가능한 동작 방식이 존재한다.

기본적으로, MXE가 특정 클러스터를 사용하기 위해서는 클러스터 내 모든 Arx 노드의 만장일치 승인이 필요하다. 실제 운영에서는 Arx 노드들이 자신들이 작업을 거부하는 MXE 또는 연산 고객(즉 MXE 생성자)에 대한 오프체인 블랙리스트를 유지하고, 나머지 모든 MXE 승인 요청을 자동 승인하는 방식이 일반적일 수 있다. 이 구조에서는 단 하나의 Arx 노드라도 특정 MXE 승인을 거부하면, 해당 MXE는 그 클러스터를 사용할 수 없게 된다(연산 고객은 다른 클러스터를 찾아야 한다).

대안적으로, 클러스터는 모든 MXE 승인 요청을 자동 승인하도록 구성될 수도 있다. 이 경우 개별 Arx 노드의 만장일치 승인은 요구되지 않으며, 노드가 해당 클러스터에 참여하는 순간 향후 모든 MXE에 대한 포괄적 승인을 제공한 것으로 간주된다. 마지막으로, 클러스터의 MXE 승인 권한을 특정 주체(예: 클러스터 내 특정 노드 하나)에 위임하는 방식도 가능하다.

이러한 동작 방식은 단일 사용 및 지속형 MXE 모두에 동일하게 적용된다. 빠르게 새로운 MXE를 사용하고자 하는 연산 고객은 자동 승인 기능이 활성화된 클러스터를 선택하거나, 높은 평판을 가진 Arx 노드들로 구성된 만장일치 승인형 클러스터를 선택할 수 있다. 전문적인 Arx 노드 운영자는 일반적으로 자체적인 동적 MXE 승인 스크립트를 운영하므로, 승인 요청에 즉시 반응할 가능성이 높다.

승인되지 않은 MXE 요청은 한 개의 전체 예폭이 지난 이후 연산 고객이 철회할 수 있다.

8.3 키세어 분배 및 보안

아르키움 네트워크에서 클러스터는 분산 키 생성(DKG, Distributed Key Generation) 과정을 통해 암호학적 키 관리를 수행한다. 이 과정에서 MXE 클러스터 내 각 Arx 노드는 전체 암호학적 키의 일부 조각인 키셰어를 받는다. 이러한 키셰어는 집합적으로 MXE의 키를 구성하며, 개별 입력 값을 공개하지 않고도 안전한 공동 연산을 가능하게 한다.

아르키움 네트워크의 핵심 MPC 프로토콜인 케르베로스는 모든 노드가 MXE 데이터 복호화에 필요하도록 설계되어 있다. 따라서 특정 클러스터 내 최소 하나의 노드만 항상 안전하고 신뢰 가능하다면, 연산의 무결성은 보호된다. 결과적으로, 애플리케이션 요구사항에 따라 클러스터 크기를 조정함으로써 기밀 연산의 보안 수준을 제어할 수 있다. 추가적인 보안 강화를 위해, 클러스터 내 노드는 선택적으로 TEE를 사용할 수도 있다.

8.4 클러스터 포크

클러스터 포크는 기존에 특정 MXE의 사용을 만장일치로 승인했던 클러스터의 일부 Arx 노드가 이후 해당 MXE를 클러스터에서 추방하기로 결정할 때 발생한다. 클러스터 포크는 MXE가 클러스터의 정책, 서비스 약관 또는 관할권 규제 요건을 위반한다고 판단될 경우 해당 MXE를 제거하기 위해 사용된다.

특정 Arx 노드가 MXE 추방을 요청하면, 동일 클러스터 내 다른 노드들은 다음 에폭 시작 전까지 해당 추방에 동참할지 여부를 결정해야 한다. 다음 에폭이 시작되면 클러스터는 포크되며, MXE 추방에 참여한 노드들은 해당 MXE를 계속 지원하기로 한 노드들만 포함하는 새로운 클러스터를 생성하기 위한 비용을 공동 부담해야 한다(모든 참여 노드 간 균등 분배). 또한 MXE를 새 클러스터로 마이그레이션하는 비용 역시 공동 부담한다.

포크 이후, 추방된 MXE에 대한 연산 책임은 기존 클러스터에서 새롭게 생성된 클러스터로 이전된다. 이를 통해 기존 클러스터는 포크와 무관한 다른 MXE들에 대한 서비스를 중단 없이 지속할 수 있다.

8.5 클러스터 마이그레이션

클러스터 마이그레이션은 특정 클러스터 내 하나 이상의 노드가 더 이상 참여할 수 없게 되는 상황에서 클러스터를 복구하는 과정이다. 마이그레이션은 다양한 이유로 발생할 수 있으며, 아래에서 각각 설명된다.

8.5.1 강제 마이그레이션

슬래싱이나 스테이킹 감소로 인해 발생하는 마이그레이션 비용은 Arx 노드 참여자 전체(운영자의 자기 위임 및 모든 제3자 위임)에 비례적으로 분산된다. 이는 모든 위임자가 노드 운영자 평가 및 스테이킹 수준 모니터링에 공동 책임을 가지기 때문이다. 실제로 마이그레이션 비용은 매우 작을 가능성이 높으며, 일반적으로는 해당 에폭의 보상 수익이 약간 감소하는 수준에 그칠 것이다.

8.5.2 계획된 마이그레이션

Arx 노드가 네트워크 종료를 원할 경우, 슬래싱을 피하기 위해 최소 한 개의 전체 에폭 전에 종료 의사를 공개해야 한다. 또한 새로운 클러스터 구성으로의 마이그레이션 비용은 자기 위임 스테이킹에서 차감된 이후 반환된다. MINIMUM_SELF_DELEGATION_PER_1000_CLUSTERS라는 네트워크 상수는 모든 Arx 노드가 자신이 속한 클러스터의 마이그레이션 비용을 충분히 부담할 수 있도록 강제한다. 따라서 노드의 자기 위임은 항상 잠재적 마이그레이션 비용을 감당할 만큼 충분한 규모를 유지해야 한다.

마이그레이션 이후 새롭게 구성된 클러스터는 기존 클러스터의 사전 설정 파라미터를 기준으로 재구성된다.

참고로, 클러스터 마이그레이션 시 새 노드 집합의 최대 연산 부하가 기존보다 낮아질 수 있으며, 이 경우 멤버에서 실행 대기 중이던 일부 연산이 취소될 수 있다. 그러나 이는 연산 고객이 클러스터를 생성 또는 선택할 때 백업 노드 포함 전체 하드웨어 사양을 이미 알고 있다는 점에서 고객 책임에 해당한다.

8.5.3 급격한 스테이킹 감소

특정 노드에 위임된 스테이킹 감소로 인해 Arx 노드의 활성 하드웨어 규모가 줄어들고, 그 결과 노드가 속한 클러스터의 최소 연산 용량 기준을 충족하지 못하게 되면, 해당 노드는 클러스터에서 제거되고 강제 마이그레이션이 발생한다.

클러스터의 연산 용량 기준은 공개되어 있으므로, 위임 철회나 재위임이 강제 마이그레이션을 유발할 수 있는 상황을 모니터링하는 도구가 개발될 가능성이 높다. 따라서 노드 참여자는 자신의 Arx 노드 총 스테이킹 규모 변화를 지속적으로 모니터링하고, 필요 시 현재 에폭 내에서 적절한 조치를 취해야 한다.

8.6 기존 클러스터 재사용

클러스터는 최초 연산 고객에 의해 생성되지만, 아르키움 네트워크 내 모든 연산 고객이 자유롭게 공유 및 재사용할 수 있는 공개 자원이다. 새로운 MXE 생성 시 기존 클러스터를 재사용할 경우, 새로운 클러스터 생성에 필요한 추가 온체인 트랜잭션 비용을 절약할 수 있다는 장점도 존재한다. 클러스터는 대부분 불변 구조이기 때문에(예외적으로 최대 연산 부하 상향 조정 및 노드셋 마이그레이션만 가능), 공유 구조 역시 안정적으로 작동한다.

8.7 허가형 클러스터

아르키움 네트워크는 공개형(비허가형)부터 부분 허가형, 완전 허가형(자체 운영형)에 이르기까지 다양한 권한 모델을 지원한다.

8.7.1 완전 허가형 클러스터

일부 조직이나 기관은 클러스터를 포함한 전체 네트워크 스택을 자체 운영하기를 원할 수 있다. 이러한 폐쇄형 구조 역시 아르키움 네트워크 위에서 동작할 유인을 가진다. 이는 네트워크가 중개 수수료를 부과하지 않기 때문에 가치 손실이 없으며, 광범위한 분쟁 해결 메커니즘의 이점을 활용할 수 있기 때문이다. 또한 이러한 구조 역시 네트워크 전반의 기본 가격 발견 메커니즘으로부터 혜택을 받는다.

8.7.2 부분 허가형 클러스터

부분 허가형 클러스터는 조직 또는 기관이 일부 Arx 노드를 직접 운영하면서도, 동시에 외부 Arx 노드를 포함하여 노드셋을 다변화하는 구조이다. 이러한 하이브리드 모델은 외부 노드가 모두 손상되더라도 악의적 다수 MPC 규칙 하에서 여전히 안전하게 동작할 수 있는 장점을 가진다. 동시에 외부 감시 기능 역시 제공된다.

8.7.3 데이터 협업

허가형 클러스터의 대표적 활용 사례는 여러 조직이 각자 보유한 데이터에 대해 공동 분석 또는 AI 모델 학습을 수행하는 것이다. 아르키움 네트워크에서는 각 조직이 서로의 민감한 데이터에 접근하지 않으면서도 공동 분석 결과로부터 상호 이익을 얻을 수 있다.

8.8 시빌 저항성

분산형 아르키움 네트워크에서 무결성과 보안을 유지하기 위해서는 시빌 공격에 대한 강력한 대응 메커니즘이 필요하다. 시빌 공격은 하나의 주체가 다수의 가짜 정체성을 생성해 네트워크에 과도한 영향력을 행사하는 공격이다. 기본적인 저항성은 각 노드가 일정 규모의 스테이킹을 요구받는 전형적인 지분증명 메커니즘을 통해 제공되지만, 아르키움 네트워크는 추가적인 대응 전략도 함께 적용한다.

8.8.1 클러스터 내부 시빌 저항성

클러스터 내부 시빌 저항성은 동일 클러스터 내 노드 간 공모를 방지하는 데 초점을 둔다. 네트워크 전체 노드셋이 다양하더라도, 특정 공개 클러스터가 지나치게 중앙화된 노드 구성을 가진다면 클러스터 내부 시빌 리스크가 발생할 수 있기 때문이다. 아르키움 클러스터는 최소 하나의 정직한 노드만 존재해도 안전성이 유지된다. 따라서 아르키움 네트워크는 모든 비허가형 클러스터에 최소 하나의 무작위 노드를 포함하도록 강제한다. 이 무작위 노드는 네트워크 전체에서 선택되며, 클러스터 내부 시빌 공격 가능성을 크게 줄인다.

8.8.2 네트워크 전반 시빌 저항성

네트워크 전체 수준의 시빌 공격은 기본 가격 투표 및 비참여 탐지 메커니즘을 포함한 프로토콜 전체 합의를 훼손할 위험을 가진다. 아르키움 네트워크는 클러스터 노드셋 규모 증가와 무작위 노드 포함 메커니즘을 통해 이러한 리스크를 완화한다. 또한 노드 운영자 평판 시스템, 커뮤니티 참여, 전략적 처벌 메커니즘 역시 시빌 저항성을 강화한다. 오프체인 평판 시스템은 노드 운영자의 과거 성과, 커뮤니티 활동, 타 네트워크·프로토콜에서의 검증 서비스 경험 등을 기반으로 신뢰도를 형성한다. 또한 커뮤니티 참여자는 의심스러운 활동을 적극적으로 감시하고 보고함으로써, 사회적 방어 계층을 형성하게 된다.

9 Arx 노드

9.1 노드 구성 및 설정

Arx 노드를 출시할 때는 해당 노드에 대한 일부 메타데이터 세부 정보를 제공해야 하며, 노드는 반드시 노드 운영자와 연결되어야 한다. 노드 운영자는 해당 노드를 운영하는 주체를 나타내는 메타데이터 엔터티이다.

9.1.1 노드 운영자

노드 운영자는 Arx 노드를 운영하는 주체를 나타내는 메타데이터 엔터티이다. 이들은 노드와 일대다 관계를 가지며, 즉 하나의 노드 운영자가 여러 노드를 운영할 수 있다. 노드 운영자와 관련된 메타데이터 필드는 다음과 같다.

- 관할권(Jurisdiction): 관할권에 대해 ISO 3166-1 alpha-2 코드 표준을 따른다(예: 독일은 DE, 프랑스는 FR 등).
- URL: 노드 운영자를 나타내는 웹페이지의 URL이다(예: 운영자의 서비스와 평판을 설명하는 소셜 미디어 또는 웹페이지).

노드 운영자의 관할권은 등록 시 운영자가 선언하며, 이후 연산 고객 및 클러스터 공동 구성원에 의해 표준 검증 방법을 통해 추가로 확인될 수도 있다.

9.2 키세어 관리 및 보안

Arx 노드는 자신이 속한 클러스터와 연결된 각 MXE에 대한 키세어를 저장하고 관리한다. 이러한 키세어의 저장과 보안은 노드가 아크키움 네트워크의 MPC 프로토콜에 적극적으로 참여하기 위해 필수적이다.

보다 정교한 노드 구성은 키세어를 TEE 내부에도 저장할 수 있다.

더 넓은 클러스터 맥락에서 키세어 분배 및 관리가 어떻게 이루어지는지에 대한 자세한 내용은 섹션 8.3을 참조하라.

9.3 운영 인센티브

노드 운영자는 계산 관련 수익원을 두 가지 가진다: 자기 위임에서 발생하는 보상과 자신의 Arx 노드에 대한 제3자 위임에서 수취하는 수수료이다. 일반적으로 네트워크가 성장할수록 위임에서 발생하는 수수료가 노드 수익의 대부분을 차지하게 될 가능성이 높다.

9.3.1 자기 위임

Arx 노드를 운영하기 위해서는, 노드 운영자가 최소한 `MINIMUM_SELF_DELEGATION_PER_1000_CLUSTERS` 상수만큼의 스테이킹을 자기 위임해야 하며, 그래야 해당 노드가 활성화된다. 이 상수는 자기 위임 스테이킹의 최소 요구 기준으로 정의되며, 최대 1000개의 클러스터에 대해 Arx 노드가 구성원으로 참여할 수 있도록 한다. 이는 해당 스테이킹 규모가 최대 1000개 클러스터에 대한 잠재적 마이그레이션 비용을 충당하기 때문이다.

노드 운영자는 `MINIMUM_SELF_DELEGATION_PER_1000_CLUSTERS`라는 네트워크 전역 상수 이상의 스테이킹을 추가로 자기 위임할 수 있으며, 이러한 추가 자기 위임은 노드가 1000개를 초과하는 클러스터에도 참여할 수 있도록 한다. 어떤 노드의 현재 총 자기 위임량을 z 로 나타낸다면, 해당 노드가 현재 구성원으로 속할 수 있는 최대 클러스터 수 x 는 다음과 같이 계산할 수 있다.

$$x = \frac{1000z}{\text{MINIMUM_SELF_DELEGATION_PER_1000_CLUSTERS}} \quad (1)$$

9.3.2 수수료

노드 운영자는 자신의 노드에 위임한 위임자들에게 지급되는 보상에서 수취할 수수료를 설정할 수 있다(곡선이 적용되는 경우, 곡선 조정 후 금액에서 차감된다). 이 선택적 수수료는 순수한 위임만이 아니라 노드 운영 자체를 유도하기 위해 필요하다.

Arx 노드가 생성될 때, 수취 수수료는 해당 노드의 위임자 보상에 적용될 `fee_rate` 파라미터로 정의된다. 또한 노드 생성 시 `max_fee_rate` 파라미터도 함께 정의되며, 이는 향후 `fee_rate`가 가질 수 있는 변경 불가능한 최대값을 의미한다(생성 이후 이 최대값은 절대 변경할 수 없다). 노드 운영자가 수수료율을 수정해도 실제 반영은 두 개의 완전한 에폭 이후에만 이루어진다. 즉, 현재 에폭에서 수수료율 변경이 요청되면, 해당 에폭이 끝나고 두 개의 완전한 에폭이 지난 후에야 새로운 수수료율이 활성화된다. 이는 Arx 노드가 에폭 종료 시점에 수수료를 변경하더라도 위임자들이 다음 에폭 동안 재위임하거나 위임 회수하여 새 수수료율에 맞춰 스테이킹을 조정할 시간을 확보하도록 하기 위함이다. 재위임 및 위임 회수 작업 자체도 1 에폭 지연을 적용하므로, 수수료율 변경 시에는 2에폭 지연이 필요하다. `fee_rate`와 `max_fee` 파라미터는 모두 베이스스 포인트 단위로 정의된다(예를 들어, 5% 수수료율은 500으로 표현된다).

9.4 성능 및 신뢰성 추적

Arx 노드 운영자에게 평판은 매우 중요하다. 이는 이들이 평가받는 핵심 기준이기 때문이다. 연산 고객과 제3자 스테이킹 위임자 모두 Arx 노드의 평판을 비판적으로 평가해야 한다. 연산 고객은 신뢰 가능하고 효율적인 연산을 위해 노드에 의존하며, 제3자 위임자는 슬래시 패널티를 피하기 위해 가장 신뢰할 수 있는 Arx 노드를 선택해 위임하기 때문이다.

오프체인 평판은 팀 평판, 회사의 투명성, 인프라 구성 등 다양한 기준을 바탕으로 평가될 수 있다. 반면 온체인 평판은 주로 Arx 노드의 과거 슬래시 기록을 기준으로 평가되며, 이는 비참여 및 치팅에 대한 모든 과거 슬래시 사례를 포함한다. 슬래시에 대한 자세한 내용은 섹션 5.5.1을 참조하라. 물론 새롭게 출시된 Arx 노드는 아직 충분한 이력이 없으므로, 슬래시 기록 평가는 일반적으로 Arx 노드가 아크키움 네트워크에 참여해 온 지속 기간도 함께 반영한다.

9.5 신뢰 실행 환경(TEEs, Trusted Execution Environments)

신뢰 실행 환경(TEE, Trusted Execution Environment)은 Arx 노드 프로세서 내부에 존재하는 안전한 엔클레이브로, 키세어와 데이터를 해당 TEE 내부에만 보관하고 머신의 다른 프로세스로부터 격리함으로써 노드의 보안 보장을 향상시킨다.

9.6 보안 고려사항

Arx 노드를 아크리움 네트워크에서 성공적으로 운영하기 위해서는, 노드 소프트웨어가 매우 신뢰성 높은 인프라 구성에서 실행되어야 하며, 비참여에 대한 처벌을 피하기 위해 최대한의 가동 시간과 장애 조치 보호가 보장되어야 한다. 이러한 신뢰성 요구를 충족하기 위해 Arx 노드 운영자는 네트워크 부하 분산, 물리적 데이터센터와 클라우드 컴퓨팅 서비스를 결합한 솔루션, 그리고 퍼블릭 인터넷을 우회하여 격리된 Arx 노드와 클라우드 사이에 전용 네트워크 경로를 제공하는 직접 클라우드 연결과 같은 정교한 시스템을 구현하려 할 것이다. 이는 지연과 패킷 손실을 크게 줄이는 동시에 보안성과 신뢰성을 향상시킨다.

10 연산

10.1 연산 작업의 유형

아크리움 네트워크에는 크게 두 가지 유형의 연산이 존재한다: 시스템 연산과 일반 연산이다. 시스템 연산은 아크리움 네트워크 자체의 운영과 관련된 프로세스에서 생성되며, 그 외의 모든 연산은 연산 고객이 요청하는 작업이다.

10.1.1 시스템 연산

시스템 연산은 아크리움 네트워크 자체 내의 필수 프로세스를 관리하며, 다음 세 가지 유형으로 구성된다.

- **DKGSystemComputation**: 새로운 MXE가 새로운 클러스터에 합류할 때 실행되는 분산 키 생성(Distributed Key Generation, DKG) 연산이다. 이 연산의 우선순위 수수료는 연산 고객이 MXE를 클러스터에 추가하도록 요청할 때 설정한다.
- **MigrationSystemComputation**: 클러스터 마이그레이션 연산으로, 계획된 마이그레이션이든 강제 마이그레이션이든 모두 포함된다. 이 경우 우선순위 수수료는 마이그레이션되는 클러스터 내 모든 Arx 노드가 기여할 수 있으며, 즉 각 노드가 부담하는 수수료 비율은 동일하지 않다. 각 노드는 자신에게 필요한 실행 속도 중요도에 따라 우선순위 수수료를 얼마든지, 혹은 최소한만 기여할 수 있다.
- **NonParticipationDetectionSystemComputation**: 클러스터 내에서 비참여가 탐지되었고, 클러스터 내부 합의만으로는 해결되지 않을 때 발생하는 연산이다(더 넓은 Arx 노드 합의가 필요함). 이 경우 우선순위 수수료는 비참여 분쟁을 유발한 원래 연산에서 직접 승계된다. 분쟁이 슬래싱으로 이어지면, 이 중복된 우선순위 수수료 비용은 시스템 연산을 수행한 노드들에게 환급된다. 그렇지 않을 경우, 이 중복 우선순위 수수료는 관련 노드들의 소규모 수익 손실을 의미한다.

10.1.2 표준 연산

아크리움 네트워크의 표준 연산은 기밀 컴퓨팅을 수행하기 위해 연산 고객이 제출한다. 모든 연산은 항상 연관된 MXE의 맥락 안에서 정의된다.

10.2 연산 정의하기

연산은 실행되기 전에 반드시 연산 정의(Computation Definition)로 정의되어야 한다. 연산 정의는 항상 연관된 MXE(단일 사용 또는 지속형)의 맥락 안에 존재한다. 각 연산 정의는 연산의 출력 세부사항과 더불어 정의 버전 업그레이드를 추적하고 연산 실행 인스턴스가 항상 의도된 정의 버전에서 수행되도록 보장하기 위한 버전 필드를 포함한다.

연산 정의에는 실행 권한도 부여되어야 한다. 즉, 해당 정의를 사용해 연산을 실행할 수 있는 주체는 다음 중 하나로 설정된다.

- **None**: 현재 이 정의를 사용해 연산을 실행할 수 있는 주체가 없다.
- **Private**: 단일 주체만 이 정의를 사용해 연산을 실행할 수 있다.
- **Restricted**: 이 정의를 사용해 연산을 실행할 수 있는 특정 주체들의 목록이 존재한다.
- **Public**: 누구나 이 정의를 사용해 연산을 실행할 수 있다.

연산 정의의 실제 실행 로직은 회로로 정의된다. 회로는 오프체인에 게시되며, 온체인에는 해시와 함께 노드가 이를 가져올 수 있는 원격 버킷 참조만 저장된다. 향후에는 공개적으로 보이지 않는 회로도 지원하도록 이 메커니즘이 확장될 수 있다.

케르베로스 프로토콜은 산술 회로 위에서 작동하므로, 그 기본 단위는 필드 원소와 타원곡선 점이다. 따라서 완전 동형 암호(FHE)처럼 이러한 연산을 모방할 필요 없이, 이들 유형에 직접 적용되는 많은 암호학적 연산을 네이티브하게 처리할 수 있다. 또한 조건문과 비교 연산 같은 논리 게이트, 그리고 룩업(lookup)과 같은 보다 복잡한 게이트도 지원할 수 있으므로, 진정한 범용 컴퓨팅도 가능하다.

10.2.1 Arcis 회로

연산 정의 내 회로는 Arcis를 사용해 작성된다. Arcis는 기밀성 보존 MPC 로직을 개발하기 위한 Rust 기반 프레임워크이다. Arcis는 표준 Rust 타입(예: `u8`, `bool` 등)을 시크릿 셰어드(secret-shared) 데이터 위에서 동작하는 마스크된 변형으로 대체함으로써 암호학적 복잡성을 추상화한다. 주요 기능은 다음과 같다.

- **타입 안전 기밀성(Type-Safe Confidentiality)**: 명시적으로 공개되지 않는 한 모든 연산은 데이터 비밀성을 유지한다.

- 고정소수점 연산(Fixed-Point Arithmetic): 52개의 소수 비트와 75개의 정수 비트를 가진 고정소수점 표현을 통해 정밀도가 중요한 연산(예: 금융 모델)을 지원하며, 유한체 내에서 유리수 처리를 가능하게 한다. 향후에는 더 동적인 범위가 지원될 예정이다.
- 사이드 채널 저항성(Side-Channel Resistance): 비밀 값에 따라 조건문 분기가 발생할 경우, 양쪽 경로를 모두 실행함으로써 타이밍 정보 유출을 제거한다.

Arcis로 생성된 회로는 모든 아크리움 MPC 프로토콜과 호환되므로, 개발자는 로직을 다시 작성하지 않고도 보안과 성능 사이에서 균형을 조정할 수 있다.

10.2.2 데이터 파라미터

연산 정의와 연관된 데이터 파라미터는 다음 두 가지 중 하나이다.

- 온체인 계정(On-Chain Accounts): 연산 정의는 온체인 블록체인 기반 데이터를 처리하도록 설정될 수 있다.
- 직접 입력(Direct Input): 연산 정의는 트랜잭션 호출 데이터를 통해 직접 제출된 데이터를 받을 수 있다.

향후 아크리움에서 MPC 프로토콜 지원 범위가 확장됨에 따라, 이러한 데이터 제공 옵션도 추가적으로 확장될 가능성이 높다.

10.2.3 실행 비용

연산 정의와 연결된 회로에는 회로 내 산술 연산 수와 입력 및 출력 수에 대한 메타데이터도 포함된다. 이 메타데이터는 포함된 각 연산 유형의 정확한 개수에 기반해 회로의 실행 비용을 정밀하게 계산할 수 있도록 한다.

10.3 연산 발주하기

연산이 정의되면, 개별 연산을 실행하도록 발주할 수 있다. 먼저, 인자(arguments), 이후 실행될 콜백, 지불 입찰 정보를 포함한 실행의 정확한 세부사항을 제공해야 한다. 또한 연산 고객은 연산 실행이 유효한 시간 창을 지정할 수 있다. 이후 해당 연산은 아크리움 네트워크의 온체인 메모로 전송된다.

연산을 발주할 때, 연산 고객은 해당 연산에 대해 지불할 우선순위 수수료를 명시적으로 정의해야 한다.

발주 시 연산에 전달되는 인자는 연산 정의 단계에서 지정된 파라미터와 자연스럽게 정확히 대응된다.

10.3.1 콜백

콜백은 연산 실행 이후 수행할 맞춤형 동작을 가능하게 한다. 콜백은 정의 자체가 아니라 발주 시점에 정의되는데, 이는 콜백 동작이 기본적인 연산 정의의 로직과 회로에는 영향을 주지 않으면서 특정 인스턴스에만 고유하게 적용될 수 있도록 하기 위함이다. 콜백은 성공 및 실패 사례 모두에 대해 생성될 수 있으며, 이는 연산 실행 결과에 따라 서로 다른 지시가 수행됨을 의미한다.

성공 사례에 대한 콜백 지시는 정적일 수도 있고 동적일 수도 있다. 정적 지시는 미리 정의된 온체인 동작을 수행할 수 있게 하며, 동적 지시는 성공한 연산의 출력값을 사용해 다른 프로그램 위에서 보다 복잡한 맞춤형 온체인 동작을 수행할 수 있게 한다.

실패 사례에 대한 콜백 지시는 출력이 생성되지 않기 때문에 항상 정적 온체인 동작이다.

10.4 연산 생명주기

연산 정의는 MXE에 연결된 논리 연산의 명세이므로 만료되지 않는다. 그러나 개별 연산 인스턴스(즉, 발주된 연산)는 실행 이후 암묵적으로 만료된다. 또한 연산은 실행이 유효한 특정 시간 창을 명시할 수 있다. 이 시간 창은 연산이 실행될 수 있는 시점을 각각 이후와 이전으로 지정하는 두 개의 타임스탬프로 구성된다.

연산의 “실행 가능 시작 시점”이 미래인 경우, 해당 연산은 메모의 별도 대기 영역에서 보류된다. 반대로 해당 시점이 현재보다 이전이라면, 연산은 즉시 실행 가능한 상태가 된다. 동시에 “실행 가능 종료 시점”이 미래에 있는 동안에는 연산 실행이 가능하지만, 해당 시점이 지나면 연산은 만료되어 더 이상 실행될 수 없다. 기본적으로 이러한 유효 시간 창이 별도로 지정되지 않은 경우, “실행 가능 시작 시점”은 0으로, “실행 가능 종료 시점”은 무한대로 설정된다.

11 네트워크 성숙도 및 지속 가능성

아크리움 네트워크의 장기적 지속 가능성은 연산 고객 수요에 달려 있다. Arx 노드는 실행된 작업에 대해 기본 가격과 고객이 정의한 우선순위 수수료를 통해 보상받으며, 우선순위 수수료는 전액 실행 노드에 귀속된다. 기본 가격은 매 에폭마다 스테이킹 가중치 노드 운영자 투표를 통해 설정되며, 이를 통해 노드들은 다양한 시장 상황에서도 비용을 충당할 수 있는 최소 요율을 유지할 수 있다.

락업 기간, 슬래싱 패널티, 그리고 계획된 종료에 수반되는 비용은 Arx 노드가 단기 변동성에 반응하여 네트워크에 빠르게 진입하거나 이탈하는 것을 추가로 억제한다. 이는 장기적으로 클러스터 신뢰성과 연산 고객 경험을 유지하는 데 기여한다.

11.1 거버넌스

아르키움의 거버넌스 구조는 두 개의 병렬 시스템을 통해 통제를 분산한다. 토큰 보유자는 온체인 거버넌스를 통해 프로토콜 수준의 경제 파라미터에 대해 투표할 수 있으며, 노드 운영자는 네트워크 운영 파라미터에 대해 투표한다. 어느 집단도 사전 공개되고 결정론적인 프로토콜 로직을 벗어나 토큰 발행을 지시하거나, 시장 활동을 수행하거나, 수혜자 지급을 승인할 수 없다. 이러한 분리는 핵심 파라미터에 대한 견제를 보장하는 동시에, 프로토콜 조정에 필요한 기민성을 유지한다.

토큰 보유자는 스테이킹 활성화 임계값, 수수료 분배 및 기타 경제 상수를 규율하는 파라미터를 포함해, 스테이킹 가중 투표를 통해 프로토콜 수준의 경제 파라미터에 대해 투표할 수 있다.

노드 운영자는 지원되는 MPC 프로토콜, 슬래싱 조건, 연산 기본 가격 등을 포함한 프로토콜 파라미터에 대해 스테이킹 가중 투표를 통해 거버넌스를 수행한다. MINIMUM_SELF_DELEGATION_PER_1000_CLUSTERS와 같은 핵심 상수는 변화하는 마이그레이션 비용을 반영하도록 거버넌스를 통해 조정될 수 있으며, 이를 통해 노드가 클러스터 신뢰성을 유지할 만큼 충분한 담보를 보유하도록 한다.

네트워크는 단계적 거버넌스를 통해 점진적으로 탈중앙화된다. 초기에는 업그레이드 가능한 컨트랙트가 채택이 안정화됨에 따라 점차 불변 구조로 전환되며, 아르키움 팀의 감독 역할은 커뮤니티 주도의 제안으로 체계적으로 축소된다. 그러나 출시 시점부터 ARX 토큰 컨트랙트는 특별 다수결 (supermajority)의 스테이킹 가중 투표를 통해서만 업그레이드될 수 있으며, 프로토콜 측 컨트랙트는 초기에는 아르키움 팀에 의해 업그레이드 가능하지만 시간이 지나면서 커뮤니티 거버넌스로 전환된다.

모든 거버넌스는 완전한 투명성을 바탕으로 온체인에서 수행된다. 핵심 스마트 컨트랙트는 오픈소스이며, Arx 노드 소프트웨어와 같은 독점적인 프로토콜 측 구성 요소는 경쟁 우위를 보호하기 위해 초기에는 비공개 소스 상태를 유지한다(그러나 궁극적으로는 이들 역시 오픈소스로 전환될 예정이다). 투표 결과와 파라미터 변경 사항은 변경 불가능한 형태로 온체인에 기록되어, 실시간 책임성을 가능하게 한다.

12 비교 분석

12.1 차별화 요소

아르키움은 하드웨어 기반 신뢰 실행 환경(TEE)에 의존하는 네트워크(예: Oasis, Secret Network) 및 완전 동형 암호(FHE) 중심 플랫폼과 근본적으로 다른 접근 방식을 취한다. 아르키움은 하드웨어 의존보다 암호학적 신뢰 불필요성을 우선시한다. 섹션 7.4에서 설명된 바와 같이, 사이드 채널 공격에 취약한 TEE 솔루션과 달리 아르키움의 MPC 클러스터는 독점 엔클레이브 없이 BFT를 구현하며, 동시에 병렬화된 MXE 실행을 통해 FHE의 과도한 지연 문제를 회피한다. 또한 중앙화된 게이트웨이를 운영하는 엔터프라이즈 MPC 서비스(예: Fireblocks)와 비교했을 때, 아르키움의 탈중앙화 아키텍처는 무허가 노드 참여와 스테이킹 기반 토큰노믹스를 결합함으로써 기밀 연산을 폐쇄형 SaaS에서 유동적인 상품으로 전환한다. 이러한 하이브리드 모델은 Web3의 인센티브 구조와 엔터프라이즈급 기밀성 보장을 독창적으로 연결한다.

13 향후 개발

초기에는 아르키움 네트워크의 온체인 구성 요소가 솔라나에 배포되지만, 향후 아르키움은 통합된 토큰노믹스를 유지하면서 여러 생태계 간 기밀성 보존 연산을 가능하게 하는 멀티체인 기밀 연산 계층으로 확장될 예정이다. 네트워크는 또한 추가적인 모듈형 MPC 프로토콜을 지속적으로 도입할 것이며, 이를 통해 개발자는 저지연 디파이부터 규제 준수형 엔터프라이즈 워크플로우에 이르기까지 특정 활용 사례에 맞는 암호학적 프리미티브를 동적으로 선택할 수 있게 된다.

13.1 멀티체인

아르키움 네트워크는 설계 단계부터 블록체인 불가지론적 구조를 채택하고 있으며, 따라서 멀티체인으로 전환될 예정이다. 아르키움 네트워크는 초기부터 체인 다양성을 경제 구조와 인센티브 설계에 통합함으로써, 멀티체인으로 전환된 이후에도 Arx 노드가 서로 다른 블록체인 배포 환경 간 자원 할당을 별도로 조정할 필요 없이 여러 체인을 동시에 지원할 수 있도록 설계되어 있다.

결론

아르키움 네트워크는 탈중앙화 기밀 컴퓨팅을 위한 새로운 패러다임을 제시하며, 여기서 ARX 토큰은 생태계의 경제적 엔진이자 신뢰의 기반 역할을 수행한다. ARX는 고정 총 공급량을 가지며, 노드 하드웨어 활성화를 위한 스테이킹 수요를 기반으로 하기 때문에, 토큰 유틸리티는 암호화된 연산에 대한 실제 수요와 직접적으로 연결된다. 이러한 설계는 노드 운영자, 위임자, 고객 간 인센티브를 정렬시키며, 보안성과 확장성을 강화하는 자기 강화적 순환 구조를 형성한다.

아рки움의 핵심은 기밀성을 특정 분야의 기능이 아닌 보편적 유틸리티로 전환하는 데 있다. 아рки움은 암호학적 프리미티브와 하드웨어 검증 기반 자원 클레임, 그리고 BFT 클러스터를 결합함으로써 Web3 규모에서의 안전한 연산을 가능하게 한다. 기업과 개발자들이 점점 더 신뢰 불필요한 사용 중 데이터 보호를 요구함에 따라, 아рки움의 토크노믹스와 아키텍처는 단순한 도구를 넘어 새로운 시대의 기밀 애플리케이션을 위한 기반 계층으로 자리매김하게 된다. 이 새로운 시대에서는 모든 연산이 기밀성을 유지하면서도 동시에 검증 가능하다.

A FAQ

- ARX 공급량은 고정되어 있는가?
그렇다. ARX의 총 공급량은 제네시스 시점에 고정된다. 프로토콜에는 추가 발행 또는 소각 메커니즘이 존재하지 않는다. Arx 노드는 연산 고객이 지불하는 기본 가격 및 우선순위 수수료를 통해서만 보상받는다(해당 비용은 체인 네이티브 토큰으로 지불되며, 자세한 내용은 섹션 6.1 참조). 또한 모든 우선순위 수수료는 해당 연산을 수행한 노드에 전액 귀속된다.
- 락업되었거나 베스팅되지 않은 토큰도 Arx 노드에 위임할 수 있는가?
아니다. 락업 상태이거나 베스팅되지 않은 토큰은 노드에 위임할 수 없다. 이러한 위임을 허용할 경우, 스테이킹된 ARX와 실제 노드 하드웨어 제공 간의 연결성이 희석되며, 토큰 유틸리티가 실제 네트워크 사용량과 연결되는 신호가 약화된다.

B 기술 참고 자료

- 솔라나 문서:
<https://docs.solana.com/>
- MPC 프로토콜:
 - BDOZ 프로토콜: Bendlin et al., “Semi-homomorphic Encryption and Multiparty Computation”
 - SGX 취약점: van Schaik et al., “SGX.Fail: Epic Fail of SGX Security”
- 아рки움 네트워크 문서:
<https://docs.arcium.com/>
- FHE 기초 이론:
Gentry, C. A Fully Homomorphic Encryption Scheme.
<https://crypto.stanford.edu/craig/craig-thesis.pdf>

C 용어집

- ARX: 아рки움 네트워크의 네이티브 유틸리티 토큰으로, 스테이킹, 거버넌스 및 프로토콜 인센티브에 사용된다.
- ArxOS: 아рки움의 오프체인 연산 인프라(노드, 클러스터, MXE)를 관리하는 운영체제 계층이다.
- Arcis: 타입 안전(type-safe) 기밀성 보장을 갖춘 MPC 회로 개발을 위한 Rust 기반 프레임워크이다.
- CU (Computation Unit): 네트워크 내 연산 작업량을 표준화하여 측정하는 단위이다.
- MXE (Multi-Party eXecution Environment): 구성 가능한 신뢰 모델을 갖춘 MPC 연산용 격리 가상 환경이다.
- 클러스터(Cluster): Byzantine fault tolerance 보장을 기반으로 연산을 수행하는 Arx 노드 그룹이다.
- 우선순위 수수료 시장(Priority Fee Market): 클러스터 연합 간 연산 스케줄링 우선순위를 결정하는 온체인 메모 메커니즘이다.
- TEE (Trusted Execution Environment): 향상된 키세어 보호를 위한 선택적 보안 엔클레이브이다(예: Intel SGX).
- DKG (Distributed Key Generation): 클러스터 노드 간 안전한 암호학적 키 공유를 위한 프로토콜이다.
- Byzantine Fault Tolerance: 분산 네트워크에서 악의적이거나 오류가 있는 노드에 대한 시스템 복원력을 의미한다.