# 인터폴드(The Interfold)

> 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.

# Interfold: 암호화 실행 환경(E3)을 위한 프로토콜

Auryn Macmillan / Nathan Ginnever / Marvin Lin

> **참고:** 이 백서는 원래 Interfold 프로토콜 설계를 설명합니다.
> Interfold는 이후 The Interfold로 발전했습니다. 본 문서는 아카이브용 기술 참조 자료로 보존되어 있습니다.

## 초록

본 백서는 암호화 실행 환경(E3)을 위한 오픈소스 프로토콜인 Interfold를 소개합니다. Interfold는 완전동형암호(FHE)를 영지식 증명(ZKP) 및 분산 임계값 암호화(DTC)의 새로운 조합과 통합함으로써, 요청된 FHE 프로그램에서 검증된 출력만 복호화하는 탈중앙화 노드 네트워크를 사용하여 여러 비공개 입력에 대한 안전한 연산을 가능하게 합니다. 이 프라이버시 보호 프레임워크는 이해관계자 주도 합의 메커니즘에 의존하며, 데이터 프라이버시, 데이터 무결성, 올바른 실행에 대해 강력한 경제적 및 암호학적 보장을 제공합니다.

Interfold의 신뢰 보장은 책임 없는 접근, 협업형 데이터 분석, 증명 가능하게 정당한 결과로 특징지어지는 새로운 유형의 애플리케이션을 가능하게 합니다. 마지막 사용 사례는 Interfold의 대표 애플리케이션인 CRISP(Coercion-Resistant Impartial Selection Protocol)를 통해 입증되며, 이는 강압을 방지하고 디지털 의사결정의 취약성을 완화하는 것을 목표로 하는 비밀 투표 프로토콜입니다. 본 논문은 Interfold가 금융, 헬스케어, AI 등 다른 분야에 어떻게 구현될 수 있는지 탐구하며 마무리됩니다.

## 소개

점점 더 데이터 중심적으로 변하는 환경에서, 견고한 데이터 보안과 연산 무결성의 부족은 중대한 과제를 제기합니다. 조직은 민감한 데이터를 활용할 때 프라이버시 침해 또는 법적 책임의 위험에 직면하며, 데이터 프라이버시와 올바른 실행에 대한 보장을 필요로 하는 애플리케이션은 신뢰에 의해 제약되거나 아예 존재하지 않습니다. 현재 솔루션은 전통적으로 신뢰된 운영자 또는 신뢰된 하드웨어에 의존하며(신뢰 의존적 공급망이라는 문제를 야기함), 많은 애플리케이션에 감당하기 어려운 거래상대방 리스크를 도입합니다. 이는 데이터 유용성과 규제 준수 사이의 절충을 강요할 수 있으며, 가치 있는 데이터를 취약하거나, 과소 활용되거나, 사용되지 않는 상태로 남겨둡니다.

Interfold는 탈중앙화 노드 네트워크에 의해 보호되는 여러 비공개 입력에 대한 안전한 연산을 가능하게 하는 암호화 실행 환경(E3)을 통해 이러한 문제를 해결합니다. Interfold의 노드는 완전동형암호(FHE)를 영지식 증명(ZKP) 및 분산 임계값 암호화(DTC)의 새로운 조합과 통합함으로써, 요청된 FHE 프로그램에서 검증된 출력만 복호화하도록 설계되어 있으며, 데이터 프라이버시, 데이터 무결성, 올바른 실행에 대해 강력한 암호학적 보장을 제공합니다. 이러한 연산 인프라는 조직이 민감한 데이터를 보호하면서 동시에 활용할 수 있게 하여, 프라이버시를 보존하고 규정 준수를 보장하는 새로운 혁신 기회를 엽니다.

E3는 다양한 출처의 여러 입력을 집계하거나 이에 대해 연산하는 것이 가치 있는 모든 사용 사례를 처리하도록 설계되어 있으며, 입력과 중간 상태가 비공개로 유지되고, 연산이 안전하며, 출력이 검증 가능함을 보장합니다. Interfold의 모듈형 및 조합 가능한 설계로 인해, E3는 매우 다양한 암호화 방식과 호환되며 암호화된 데이터에 대해 임의의 연산을 안전하게 실행할 수 있습니다. 프로토콜의 일시적 실행은 리스크를 더욱 최소화하며, 일회용 키는 단일 주체가 통제하는 전역 키에 존재하는 근본적인 취약성과 경제적 비효율성을 제거합니다. 블록체인 기술을 활용하여 일부 바람직한 속성(검열 저항성, 검증 가능성, 데이터 가용성)을 계승하지만, Interfold의 인프라는 web3 생태계 안팎 모두에 폭넓게 적용될 수 있습니다.

Interfold의 사용 사례는 세 가지 주요 유형으로 분류될 수 있습니다:

1. 증명 가능하게 정당한 결과: Interfold는 여러 비공개 입력에 대해 연산하기 위해 신뢰된 운영자(예: 경매인) 또는 신뢰된 하드웨어(Trusted Execution Environment)가 필요하지 않도록 함으로써 결과의 정당성을 높입니다. 예시로는 비밀 투표와 봉인입찰 경매가 있습니다.
2. 책임 없는 접근: Interfold는 FHE를 사용하여 민감한 데이터에 대한 안전한 연산을 촉진함으로써 책임을 줄이고 데이터 접근성을 높입니다. 이는 프라이버시와 규정 준수가 중요한 의료 연구 및 금융 데이터 분석과 같은 애플리케이션에 필수적입니다.
3. 협업형 데이터 분석: Interfold는 여러 주체가 원시 데이터셋을 공유하지 않고 암호화된 데이터에 대해 공동으로 연산할 수 있게 하여, 민감하거나 독점적인 정보의 프라이버시를 보존합니다. 이러한 혁신은 유전체학 및 기후 과학과 같은 분야가 입력의 완전한 기밀성을 유지하면서 데이터에 대해 안전하게 협업할 수 있도록 합니다.

Interfold 운영 모델의 핵심에는 데이터 처리를 감독하는 스테이킹된 노드의 탈중앙화 네트워크가 있습니다. 다단계 프로세스를 통해 이러한 데이터 처리 interfold는 이해관계자 참여에 의해 구동되는 합의 메커니즘을 통해 연산의 올바른 실행과 비공개 데이터의 기밀성을 보장합니다. 비례 수수료, 처벌 조치, 스테이킹 메커니즘의 전략적 사용은 투명성과 책임성을 높이며, 프라이버시, 활성성, 무결성에 대한 강력한 암호학적 보장과 경제적 보장을 결합합니다.

본 백서에서는 Interfold의 핵심 원칙, 기술 세부사항, 기능을 살펴보고, E3를 활용하여 검증 가능한 비밀 투표를 가능하게 하는 대표 애플리케이션 CRISP(Coercion-Resistant Impartial Selection Protocol)를 중점적으로 다룹니다. CRISP의 설계를 살펴봄으로써, Interfold의 모듈형 프레임워크가 민감한 데이터의 가치를 극대화하면서 프라이버시, 무결성, 신뢰를 강화하기 위해 다양한 사용 사례와 요구에 어떻게 맞게 조정될 수 있는지 보여줍니다.

# 개요

Interfold는 암호화 실행 환경(E3)을 만들기 위해 필요한 다양한 행위자를 조율하고 다양한 구성 요소를 조합하기 위한 오픈소스 프로토콜입니다. 이러한 새로운 실행 환경은 완전동형암호(FHE)와 영지식 증명(ZKP), 분산 임계값 암호화(DTC)와 같은 기타 고급 암호학 기법을 사용하여 연산을 보호합니다. 신뢰된 하드웨어에 의존하고 사이드채널

공격 및 검증 취약성에 노출될 수 있는 신뢰 실행 환경(TEE)과 달리, Interfold의 탈중앙화 아키텍처, 고급 암호학, 프라이버시 보호 메커니즘 설계는 데이터 프라이버시, 데이터 무결성, 올바른 실행에 대해 강력한 경제적 및 암호학적 보장을 제공합니다.

이러한 보장의 효력은 대부분 Interfold의 독특한 모듈형 아키텍처에서 비롯되며, 이는 다양한 사용 사례 전반에서 적응 가능하고 회복력 있는 연산을 가능하게 합니다. 다단계 연산 프로세스에 참여하는 스테이킹된 노드 네트워크를 활용함으로써, Interfold는 올바른 행동을 장려하기 위해 스테이킹 메커니즘, 비례 수수료, 처벌 조치를 특징으로 하는 정교한 경제 모델을 통합합니다. 이 설계는 신뢰 기반 시스템과 관련된 리스크를 완화할 뿐만 아니라, 다양한 분야에 맞춰진 프라이버시 보호 애플리케이션을 구축하기 위한 다목적 기반을 제공하여, 조직이 보안이나 규정 준수를 훼손하지 않고 민감한 데이터를 활용할 수 있도록 보장합니다.

# Interfold의 행위자

Interfold는 프로토콜 내 세 가지 구별되는 행위자의 조율을 통해 암호화된 데이터에 대한 안전하고 프라이버시를 보호하는 연산을 가능하게 합니다. 다음은 각 역할과 해당 프로세스에 대한 기여의 설명입니다:

## 1. 요청자

요청자는 프로세스를 시작하고 안전한 연산을 위한 파라미터를 정의합니다. 모든 개인 또는 조직은 해당 스마트 컨트랙트 진입점을 호출하고 요청된 E3의 노드 수, 임계값 수준, 기간에 비례하는 보증금을 예치함으로써 Interfold 네트워크에 E3를 요청할 수 있습니다.

## 2. 데이터 제공자

데이터 제공자는 요청된 E3에 대한 데이터 입력을 제공하는 개인 또는 시스템입니다. 이들의 평문 데이터는 요청된 E3가 생성하고 온체인에 게시한 공개 임계값 키를 사용하여 암호화되며, 이를 통해 프로세스 전반에서 데이터가 안전하고 비공개로 유지됩니다.

## 3. 암호화 실행 환경(E3)

E3는 Interfold 프로토콜에서 요청되고 이를 통해 조율되는, 암호화된 데이터에 대해 안전한 연산을 실행하는 핵심 단위입니다. E3는 탈중앙화 노드 네트워크를 사용하여 요청된 FHE 프로그램에서 검증된 출력만 복호화합니다. 이는 데이터가 비공개이고 안전하게 유지되는 동안 연산이 수행되도록 보장합니다.

# E3 아키텍처

E3는 암호화된 데이터에 대해 안전하고 검증 가능한 연산을 보장하는 여러 모듈형 구성 요소로 이루어져 있습니다. 이러한 구성 요소는 데이터 프라이버시와 연산 무결성을 유지하도록 설계되어 있으며, 신뢰 기반 솔루션과 관련된 리스크를 완화합니다.

각 E3는 다음 핵심 구성 요소로 구성됩니다:

## Ciphernode Committee(CiCo)

Compute Provider가 게시한 암호문 출력을 복호화하고 분산 키 생성을 담당하는 선택된 Ciphernode 그룹입니다. 노드는 무작위 sortition을 사용하여 위원회에 선정되며, 신뢰를 분산함으로써 보안을 강화합니다. 이러한 탈중앙화 접근 방식은 어떤 단일 노드도 완전한 통제권을 갖지 않도록 보장하여, 데이터 침해 또는 변조 리스크를 제거합니다.

## E3 Program(E3P)

수행될 연산을 정의하는 구체적인 명령어 집합입니다. E3 Program은 온체인에 게시된 암호문을 입력으로 받아 평문 출력을 생성하며, 데이터와 결과 모두의 기밀성을 보장합니다. E3P는 보안을 유지하면서 다양한 연산을 지원하도록 다목적으로 설계되었습니다.

## Secure Process(SP)

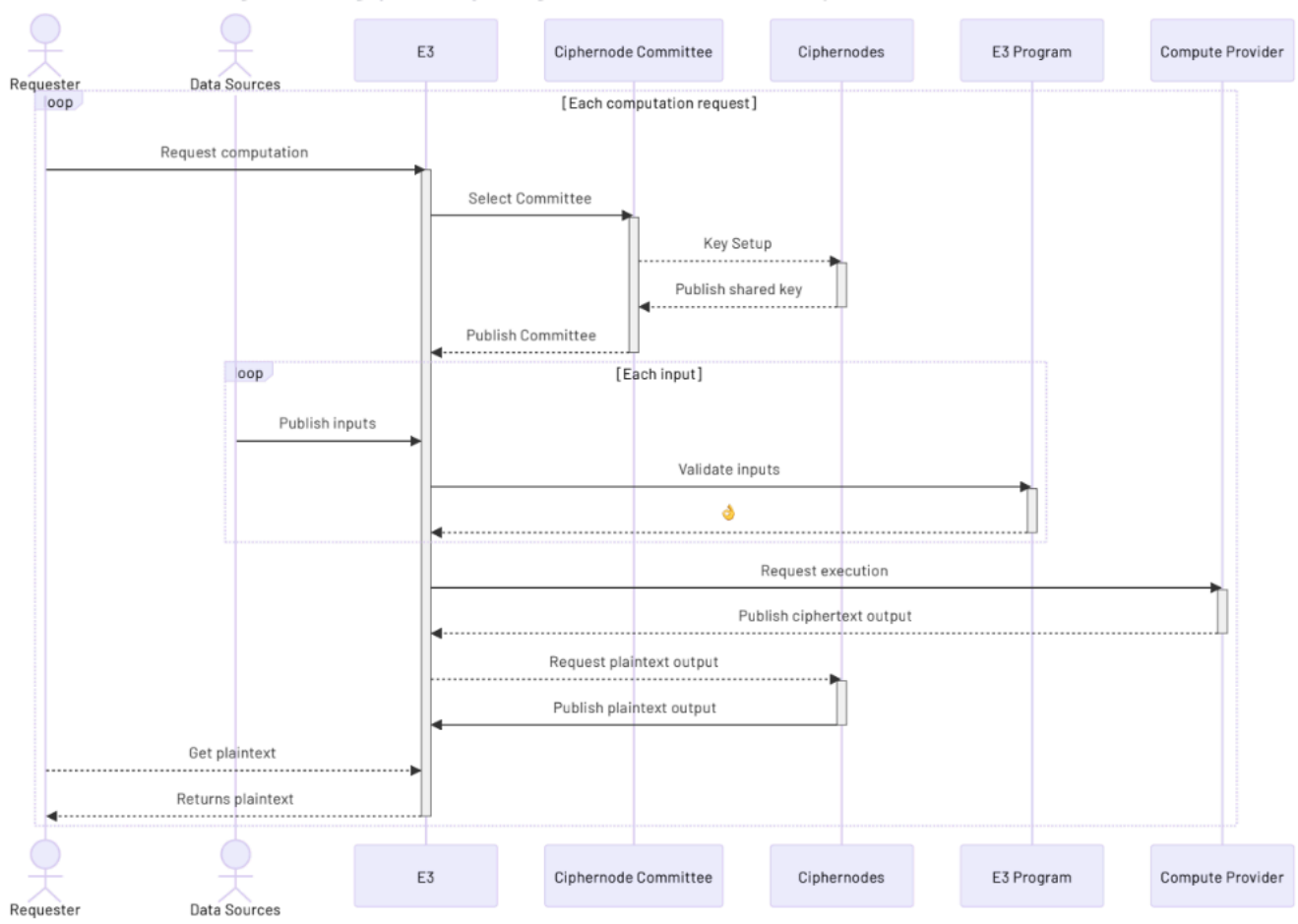
FHE 연산 실행에 전념하며 Compute Provider가 실행하는 E3P 구성 요소입니다. 프로토콜, Ciphernode 또는 Data Provider가 관리하는 다른 시스템 구성 요소와 달리, Secure Process는 Compute Provider의 검증 가능한 구성 요소에서만 실행되며, 이는 SP의 올바른 실행에 대한 온체인 증명 또는 검증을 제공합니다.

## Compute Provider(CP)

E3에서 Secure Process를 실행하는 플랫폼입니다. Compute Provider는 특정 E3에 대해 게시된 입력에 대해 Secure Process를 실행한 결과로 올바른 암호문 출력을 다시 보고할 것이라고 암묵적으로 신뢰됩니다. CP는 증명 기반 시스템과 오라클 기반 시스템으로 분류될 수 있습니다: 증명 기반 제공자(RISC Zero[1], Succinct의 SP1[2], Jolt[3], Nexus[4])는 올바른 실행에 대한 암호학적 보장을 제공하는 반면, 오라클 기반 제공자(Multisigs, Reality.eth[5], Kleros Court[6], API3[7])는 검증을 위해 합의 또는 경제적 메커니즘을 사용합니다.

# Interfold의 운영 방식

Interfold는 다양한 행위자와 E3 구성 요소가 관여하는 구조화된 다단계 프로세스를 통해 안전하고 탈중앙화된 연산을 촉진합니다. 이 프로세스는 프라이버시를 보호하면서 올바른 실행을 보장합니다. Figure 1은 이러한 상호작용을 시각적으로 나타낸 것입니다.




## Phase 1: 요청

누구나 언제든지 Interfold에 연산 요청을 시작할 수 있습니다. 이 요청에는 다음 사항이 명시되어야 합니다:

1. 수행될 FHE 연산.
2. Compute Provider의 선택 및 관련 파라미터.
3. 필요한 Ciphernode의 수.
4. 출력 검증에 필요한 각 유형의 노드에 대한 임계값.
5. Input Deadline의 타임스탬프. 이 시점 이후에는 새로운 입력이 허용되지 않습니다.
6. 노드가 이용 가능해야 하는 기간.

요청을 시작하려면 요청된 노드가 자신의 의무를 수행하는 데 대해 최소 보상을 보장하기 위한 비례 보증금이 필요합니다. 보증금 금액은 최종 단계인 Decryption 전까지 다른 참여자에 의해 증가될 수 있습니다.

## Phase 2: 노드 선정

연산이 요청되면, 필요한 수의 Ciphernode가 sortition이라는 무작위 프로세스를 통해 Ciphernode Committee를 구성하도록 선정됩니다. 선정된 Ciphernode는 요청된 임계값을 가진 공유 공개 키를 즉시 생성하고 게시합니다. 이를 통해 연산을 위한 암호학적 프레임워크가 안전하게 설정됩니다.

## Phase 3: 입력 기간

선정된 Ciphernode Committee가 공유 공개 키를 게시하면 Input Window가 열립니다. 이 단계에서 Data Provider는 연산을 위한 입력을 제출할 수 있으며, 해당 입력은 기밀성을 보장하기 위해 공개 키로 암호화됩니다. 각 입력에는 Data Provider가 자신의 입력에 대한 preimage를 알고 있음을 보장하고, 입력이 올바르게 형성되었으며, 다른 맥락에서 사용될 수 없고, 선택된 연산에서 요구하는 기타 검증 로직을 통과함을 보장하기 위한 해당 영지식 증명(ZKP)이 필요합니다.

## Phase 4: 실행

Input Window가 닫힌 후, 선정된 Compute Provider는 암호화된 입력을 사용하여 FHE 연산을 실행하는 Secure Process를 포함한 사전 정의된 E3 Program을 실행합니다. 그런 다음 Compute Provider는 암호문 출력을 생성하고 게시하여, 프로세스 전반에서 연산의 무결성이 유지되도록 보장합니다.

## Phase 5: 복호화

선정된 Compute Provider가 요청된 연산의 암호문 출력을 게시하면, Ciphernode의 임계값 이상이 Compute Provider가 보고한 연산의 출력을 공동으로 복호화하고 게시해야 합니다. 연산 출력의 평문이 게시된 후, Ciphernode의 의무는 완료됩니다. 이후 각 노드는 보증금의 비례 지분을 청구할 수 있으며, 이 위원회에 사용된 키를 toxic waste로 취급하여 폐기해야 합니다.

# 인센티브 및 페널티

무결성을 유지하고 올바른 행동을 장려하기 위해, Interfold는 인센티브와 페널티 시스템을 활용합니다.

## 스테이킹

누구나 Interfold 토큰 일정 수량[8]을 스테이킹하여 새로운 Ciphernode를 등록할 수 있습니다. 등록 후, 새로운 노드는 Ciphernode 의무에 선정될 수 있기 전에 등록 지연 기간[9]을 기다려야 합니다. 등록 지연 기간은 Requester가 연산을 요청할 때 현재 Ciphernode 세트를 합리적으로 예측할 수 있게 합니다. 이러한 경제적 지분은 이들의 이해관계를 시스템 보안과 정렬합니다.

노드는 언제든지 decommission을 요청할 수 있습니다. Decommission되기 전에, 노드는 decommission 지연 기간[10] 동안 활성 상태를 유지해야 하며, 이후에는 더 이상 의무에 선정되지 않습니다. 노드는 또한 이미 선정된 모든 의무를 완료하기 위해 활성 상태를 유지해야 합니다. Decommission 지연이 지나고 할당된 모든 의무가 완료되면, 노드는 decommission될 수 있으며 스테이킹 보증금이 반환됩니다.

## 수수료

Interfold에 연산을 요청할 때, Requester는 요청된 Ciphernode의 수와 임계값, 해당 노드가 이용 가능해야 하는 기간, 그리고 선택된 Compute Provider가 요구하는 추가 비용에 비례하는 수수료를 예치해야 합니다. 이 보증금은 요청된 Ciphernode를 예약하며, 이들이 요청된 연산의 출력을 복호화하기 위해 온라인 상태를 유지할 것이라는 경

제적 보장을 제공합니다. 그러나 보증금은 연산 자체에 대한 대가를 지불하지 않습니다. 서로 다른 Compute Provider는 Interfold 프로토콜 외부에서 부과되는 서로 다른 수수료 구조를 갖게 됩니다.

Ciphernode는 자신이 위원회 구성원이었던 연산의 입력 또는 중간 상태로부터 복호화 데이터의 자신의 지분을 제공한 것으로 입증되는 경우 페널티를 받고 강제로 decommission됩니다. Decommission된 Ciphernode에 대해서도 중간 복호화에 대한 장기적인 억제 요인을 보장하기 위해, Ciphernode 보상의 일부는 Decryption 직후 즉시 지급되며, 나머지 부분은 slashing 조건과 함께 cliff 및 vesting 일정[11]의 적용을 받습니다.

## 페널티

악의적 행동 또는 프로토콜 규칙 미준수는 해당 의무 라운드에 대한 보상 손실부터 강제 decommission에 이르는 페널티를 초래합니다. 이는 데이터 보안 또는 연산 무결성을 손상시킬 수 있는 행동을 억제합니다. 사례와 페널티는 다음과 같습니다:

- **비활성 노드:** 요청된 연산을 복호화하는 데 필요한 데이터의 자신의 지분을 제공하지 못한 Ciphernode는 복호화 수수료에 대한 자신의 지분을 몰수당합니다. 또한 누락된 서명으로 인해 발생한 추가 가스 비용을 충당하는 데 사용되는 소액 페널티[12]의 적용을 받습니다.
- **결과 보고 실패:** 선정된 Compute Provider가 요청된 연산의 암호문 출력을 적시에[13] 제공하지 않는 경우, 연산 요청은 취소됩니다. 이 경우 보증금의 일부[14]가 선정된 Ciphernode Committee에 지급되고, 연산에 대한 나머지 보증금은 Requester에게 반환됩니다.
- **복호화 실패:** 선정된 Ciphernode Committee가 선정된 Compute Provider가 동의한 암호문 출력으로부터 복호화된 평문을 적시에[15] 제공하지 않는 경우, 연산 요청은 취소됩니다. 이 경우 연산에 대한 보증금은 Requester에게 반환됩니다. 복호화 데이터의 자신의 지분을 제공하지 않은 모든 노드는 페널티의 적용을 받고 강제로 decommission됩니다. 징수된 페널티는 선정된 Ciphernode Committee의 나머지 구성원에게 비례적으로 분배됩니다.
- **비결정론적 연산:** 노드가 출력에 대해 합의에 도달하기 위해서는 요청된 연산이 결정론적이어야 합니다[16]. 선정된 Compute Provider가 요청된 연산이 비결정론적 출력을 갖는다고 보고하는 경우, 해당 출력에 대한 보증금은 일부가 선정된 노드들에게 분배되고 일부는 토큰을 매입 및 소각하는 데 사용됩니다.
- **중간 복호화:** 모든 암호화된 입력과 중간 상태의 프라이버시를 유지하는 것은 Interfold의 핵심 기능입니다. 요청된 연산의 합의된 출력을 제외하고 Ciphernode Committee에 암호화된 모든 것을 복호화하려는 시도는 강제 decommission과 함께 위반 노드 지분의 일부에 대한 slashing[17]으로 처벌됩니다. Slashing된 지분의 일부는 소각되고, 나머지 부분은 slash 가능한 위반을 네트워크에 알린 계정에 할당됩니다.
- **강제 Decommissioning:** 노드의 유효 지분이 최소 지분의 절반으로 감소하는 경우, 해당 노드는 즉시 decommission되며 더 이상 의무에 선정되지 않습니다. 그러나 노드는 이미 선정되었던 모든 의무를 완료하기 위해 계속 활성 상태를 유지해야 합니다. 할당된 모든 의무가 완료되면, 강제로 decommission된 노드는 스테이킹 보증금의 나머지를 청구할 수 있습니다.

# Interfold 거버넌스

Interfold는 최상위 의사결정 기구 역할을 하는 토큰 보유자에 의해 거버넌스됩니다. 이들은 프로토콜 파라미터 설정, 프로토콜 업그레이드 감독, 분쟁 해결 촉진 등을 담당합니다. Interfold에는 프로토콜의 공정성, 성능, 변화하는

요구사항에 대한 대응력을 유지하기 위해 주기적인 조정이 필요할 수 있는 여러 변수가 있습니다.

Interfold 토큰 보유자의 책임은 다음과 같습니다:

- 프로토콜 관리
- 프로토콜 업그레이드
- 분쟁 해결
- 파라미터 설정
- 취소 소각 비율
- 스테이킹 예치 수량
- 등록 지연 기간
- Decommission 지연 기간
- 비활성 노드 페널티 수량
- 중간 Slashing 페널티 및 소각 비율

# 대표 사용 사례: 비밀 투표

현대 의사결정 시스템은 강압, 데이터 침해, 프라이버시 침해와 같은 문제로 인해 종종 훼손됩니다. 이러한 취약성은 거버넌스 및 의사결정 시스템의 무결성과 효과성을 약화시키며, 안전하고 신뢰할 수 있는 솔루션에 대한 필요를 만듭니다.

Coercion-Resistant Impartial Selection Protocol(CRISP)은 Interfold의 고급 암호학을 통해 의사결정 시스템이 안전하고 공정하게 유지되도록 보장함으로써 이러한 문제를 해결합니다. 비밀 투표의 현대적 구현체 역할을 하는 CRISP는 디지털 의사결정을 위한 안전한 프로토콜로, FHE와 임계값 암호화를 활용하여 검증 가능한 비밀 투표를 가능하게 합니다. 이는 민주주의와 기타 많은 의사결정 애플리케이션에 중요한 구성 요소입니다. 이는 외부 및 내부 위협 모두로부터 의사결정 프로세스를 보호하기 위한 견고한 프레임워크를 구축하며, 참여자 간 신뢰의 필요성을 제거할 뿐만 아니라 민주적 의사결정 환경에 필수적인 기밀성과 무결성을 보존합니다.

# CRISP 비밀 투표의 구조

이 섹션은 현대적이고 강압에 저항하는 투표를 가능하게 하기 위해 Interfold의 범용 core 위에 구축된 구성 요소를 상세히 설명합니다. 여기에는 Requester 및 Data Provider와 Interfold 간의 상호작용을 중재하는 스마트 컨트랙트와 Interfold를 통해 실행될 투표 집계 연산이 포함됩니다.

CRISP 구현에는 Zodiac[18] 호환 모듈이 포함되어 있으며, 이는 Zodiac IAvatar 인터페이스를 따르는 모든 컨트랙트 계정을 제어하는 데 사용될 수 있습니다. 예를 들어 Safe[19]가 있습니다.

## 설정

호환 계정을 CRISP 비밀 투표에 의해 제어될 수 있도록 하려면, CRISP 모듈을 배포하고 제어될 계정에서 이를 활성화해야 합니다. 이 프로세스에는 다음 사항을 지정하는 것이 포함됩니다:

1. 모듈에 의해 제어될 계정.

2. 지정된 투표 집계 연산의 ID.
3. 지정된 실행 환경 및 추가 파라미터.
4. 필요한 Ciphernode의 수와 임계값.
5. 각 poll의 기간.
6. voter registry 컨트랙트의 주소.

스마트 컨트랙트 계정에서 모듈을 활성화하는 구체적인 단계는 구현마다 다를 수 있으며, 본 문서에서는 자세히 설명하지 않습니다. CRISP 모듈이 계정에서 활성화되면, 해당 계정이 궁극적으로 임의의 호출을 수행하도록 트리거할 수 있는 poll을 생성하는 데 이를 사용할 수 있습니다.

## Poll 생성

누구나 CRISP 모듈이 활성화된 스마트 컨트랙트 계정에 의해 실행될 트랜잭션을 제안할 수 있습니다. 이를 위해서는 CRISP 모듈의 함수를 호출하고 다음을 제공해야 합니다: 제안 설명의 해시, 트랜잭션 payload의 해시, 그리고 CRISP 네트워크가 요구하는 연산을 위한 보증금.

이는 CRISP 모듈에 제안을 등록하고, CRISP 모듈 설정에서 정의된 임계값 및 기간 파라미터를 사용하여 CRISP 네트워크에 연산을 요청합니다.

## 유권자 등록

각 poll에서 모든 유권자는 투표를 하기 위한 암호학적 keypair를 보유해야 하며, 이는 Voter Registry 스마트 컨트랙트에 등록되어야 합니다. 이 컨트랙트는 다음을 담당합니다:

- FHE 연산에서 메시지를 검증하기 위해 필요한 증명을 지정합니다.
- Poll에서 요구하는 모든 유권자 자격 기준을 시행합니다.
- 각 등록 유권자의 투표 가중치를 결정합니다.

예를 들어, Voter Registry는 유권자가 최소 수량의 토큰을 보유하거나 사전에 정의된 적격 유권자 목록의 구성원임을 증명 가능하도록 요구할 수 있습니다. 중요한 점은 투표 키가 유권자가 온체인에서 다른 방식으로 상호작용할 때 사용할 수 있는 keypair와 구별된다는 점입니다. 투표 키는 현재 poll에 특화된 일회용 키이며, poll이 종료되면 폐기 가능하고 다른 poll이나 맥락에서 재사용되어서는 안 됩니다.

## 투표

투표는 선정된 Ciphernode가 제공한 공유 키로 암호화된 암호문으로 CRISP 스마트 컨트랙트에 직접 제출되며, 암호화된 메시지가 등록 유권자가 서명한 유효한 투표 형식을 나타내고 다른 제안에서 재사용될 수 없다는 영지식 증명과 함께 제출됩니다. 이 설정은 모든 계정이 모든 사용자를 대신하여 투표 메시지를 제출할 수 있게 합니다.

유권자는 Input Deadline 이전 언제든지 CRISP 스마트 컨트랙트에 암호화된 메시지를 제출하여 자신의 투표를 변경할 수 있습니다. 가장 최근의 메시지만 집계 결과에 반영됩니다.

### 키 전환

유권자는 Input Deadline 이전 언제든지 선정된 Ciphernode가 제공한 private key로 암호화된 올바른 형식의 키 변경 메시지를 CRISP 스마트 컨트랙트에 제출하여 자신의 투표 키를 변경할 수 있습니다. 유권자의 가장 최근 유효 키로 서명된 메시지만 집계 결과에 반영됩니다.

### 결과 계산 및 게시

Input Deadline이 지나면, 선정된 Compute Provider가 집계 암호문을 계산하고, 선정된 Ciphernode는 출력을 복호화하는 동시에 평문 결과의 해시를 온체인에 게시합니다.

### 온체인 작업 트리거

선정된 Ciphernode가 평문 결과의 해시를 온체인에 게시하면, 누구나 첨부된 트랜잭션 payload를 실행하는 함수를 호출할 수 있습니다. 이 실행에는 선정된 Ciphernode가 제공한 결과의 증명을 제공해야 합니다.

## 일반적인 공격의 완화

CRISP 위에 구축된 당사의 투표 시스템은 여러 실무적 및 이론적 공격에 대해 회복력을 갖도록 설계되었습니다. 이 섹션은 이러한 공격 벡터 중 일부와 당사의 투표 구현에서 이에 대응하는 완화 방안을 상세히 설명합니다.

각 시나리오에서 Alice와 Bob은 모두 등록 유권자입니다. Alice와 Bob은 모두 Banana Party의 공개적인 지지자입니다.

Chuck은 유권자가 poll에서 투표를 제출하는 웹 서비스를 운영하며 Durian Party를 지지합니다. 다른 인물도 추가로 등장할 수 있습니다.

### 검열

Alice와 Bob이 각각 Banana Party에 투표할 가능성이 높다는 것을 알고, Chuck은 이들이 투표 애플리케이션을 통해 자신의 웹 서비스에 제출한 Alice와 Bob의 투표를 무시하고 해당 투표를 온체인에 게시하지 않기로 선택합니다. Chuck은 Alice와 Bob에게 투표가 온체인에 제출되었음을 보여주는 유효한 트랜잭션 영수증을 제공할 수 없었기 때문에, Alice와 Bob은 Chuck과 같은 중개자의 허가나 검열 없이 자신의 투표를 CRISP 컨트랙트에 직접 게시하기로 선택할 수 있습니다. Alice와 Bob은 자신의 투표를 직접 제출하거나 대체 relaying 서비스를 통해 제출함으로써 Chuck의 투표 검열 시도를 우회할 수 있습니다.

Chuck은 특정 poll에 대한 유권자 등록을 거부하려는 유사한 공격을 시도할 수도 있지만, 완화 방식은 유사합니다.

### 영수증 공유

Alice와 Bob이 각각 Banana Party에 투표할 가능성이 높다는 것을 알고, Chuck은 Alice에게 원래 선호가 아닌 Durian Party에 투표하도록 뇌물을 제공하기로 결정합니다.

Alice의 최적 행동은 Chuck의 뇌물을 받고, 뇌물이 없었을 때와 같은 방식으로 투표한 다음, 합법적인 투표와 구별할 수 없는 가짜 증명을 Chuck에게 제공하는 것입니다. 이를 위해 Alice는 Durian party에 투표한 다음,

keychange 메시지를 제출하여 자신의 투표 키를 변경하고 이전 투표를 무효화한 뒤, 마지막으로 새로운 키로 Banana party에 투표할 수 있습니다. Alice는 첫 번째 투표의 영수증을 Chuck과 공유할 수 있습니다. 그러나 Chuck은 투표에 사용된 키가 Alice의 유효한 투표 키였는지 보장할 방법이 없습니다. 따라서 Chuck은 Alice의 말을 믿을 수밖에 없습니다.

### 대리 투표

Alice와 Bob이 각각 Banana Party에 투표할 가능성이 높다는 것을 알고, Chuck은 Alice에게 원래 선호가 아닌 Durian Party에 투표하도록 뇌물을 제공하기로 결정합니다. Chuck은 Alice의 투표를 자신이 행사할 권한을 부여받는 조건으로 Alice에게 뇌물을 지급하겠다고 제안합니다. 이에 따르기 위해 Alice는 자신의 투표 키를 Chuck이 제공한 키로 전환하는 key change 메시지를 제출해야 합니다.

영수증 공유의 경우와 마찬가지로, Alice의 최적 행동은 Chuck의 뇌물을 받고 해당 key change 메시지의 영수증을 Chuck과 공유하되, 그 전에 이미 Chuck이 모르는 다른 키로 투표 키를 전환해두는 것입니다. Chuck에게 불행하게도, Alice가 이전에 다른 투표 키를 등록했거나 다른 투표 키로 전환하지 않았음을 보장할 방법은 없으며, Chuck의 투표 키로 키를 변경하는 메시지가 실제로 무효인지 판단할 방법도 없습니다.

### 강제 기권

Alice와 Bob이 각각 Banana Party에 투표할 가능성이 높다는 것을 알고, Chuck은 Alice와 Bob 모두에게 투표를 기권하도록 강제하기로 결정합니다(뇌물 또는 더 강압적인 형태의 담합을 통해). Chuck의 요구에 따르려면, Alice와 Bob은 poll에 등록하거나 투표한 것이 적발되어서는 안 됩니다.

Alice와 Bob의 최적 행동은 식별 가능한 흔적을 남기지 않도록 주의하면서 정상적으로 투표하는 것입니다. 온체인에서 등록하고 투표를 제출할 때, Alice와 Bob은 Chuck이 통제하는 relayer가 아닌 다른 relayer를 통해 메시지를 제출하거나, 자신의 신원과 연결될 수 없는 새로운 주소에서 제출할 수 있습니다. 등록에는 유권자가 registry에 포함되어 있음을 증명하는 것이 포함되지만, 유권자가 자신을 평문으로 식별하거나 특정 계정에서 등록 메시지를 제출할 필요는 없습니다. 마찬가지로, 투표를 행사할 때도 식별 가능한 정보는 평문으로 게시되지 않으며, 결과를 제외한 어떠한 암호문도 Ciphernode에 의해 복호화되지 않습니다.

## 한계

### 손상된 Registry

다른 모든 투표 구현과 마찬가지로, CRISP는 정상적으로 작동하는 voter registry에 의존합니다. 유권자 등록 프로세스가 손상되어 공격자가 등록을 거부하거나 유권자가 자신의 투표 키를 선택하기 전에 유권자의 계정을 통제할 수 있게 되면, 공격자는 시스템을 성공적으로 손상시킬 수 있습니다.

후자의 유형의 공격은 특정 poll에 적합한 범위와 신뢰 가정에 따라 다양한 방식으로 완화될 수 있습니다. 그러나 일반적인 경험칙은 각 유권자의 계정에 충분한 다른 가치가 연결되어 있어, 해당 유권자가 제3자와 credentials를 공유하려 하지 않도록 보장하는 것입니다. 최소한 이는 그러한 공격의 비용을 높이고 확장성을 낮춥니다.

### MACI와의 비교

CRISP의 설계는 영지식 증명(ZKP)을 활용한 강압 저항 투표를 위해 Vitalik Buterin[21]이 처음 제안한 프로토콜인 MACI[20]에서 많은 영감을 받았습니다. CRISP는 주로 완전동형암호(FHE)와 임계값 암호화를 사용한다는 점에서 MACI와 다릅니다. 이 접근 방식은 CRISP가 신뢰 분산을 위해 임의로 큰 노드 네트워크를 구축할 수 있게 하며, 단일 신뢰된 coordinator에 의존하는 MACI와 대비됩니다.

MACI는 프라이버시에 대해 정직한 coordinator 가정에 의존합니다. 이는 coordinator가 모든 입력과 중간 상태에 제한 없이 접근할 수 있으며 이를 공개하지 않을 것이라고 신뢰된다는 의미입니다. 반면, CRISP는 누구든 Ciphernodes가 연산의 입력 또는 중간 암호문을 복호화하려 시도했음을 증명할 수 있는 경우 해당 Ciphernodes가 slashing 대상이 되므로, 프라이버시에 대해 강력한 경제적 보장을 제공합니다.

# 잠재적 사용 사례 탐색

다음 목록은 Interfold의 잠재적 사용 사례에 대한 탐색입니다.

## 봉인입찰 경매

모든 참여자가 다른 참여자의 입찰을 알지 못한 채 입찰을 제출하는 경매 유형입니다. 최고 입찰자가 낙찰되지만, 낙찰 금액은 모든 입찰이 제출된 후에만 공개됩니다.

혜택: 공정성과 기밀성을 보장하고, 강압과 입찰 조작을 방지합니다

## 영수증 없는 위임

유권자가 자신의 선택을 공개하지 않고 투표를 위임할 수 있는 투표 시스템입니다.

혜택: 유권자 프라이버시를 보호하고 사람들이 민주적 절차에 참여하도록 장려합니다

## 비공개 의료 데이터에 대한 연구

연구 목적으로 여러 출처의 민감한 의료 기록과 건강 데이터를 분석하는 방법입니다.

혜택: 의료 연구를 안전하게 발전시키고, 규제를 준수하며, 환자 프라이버시를 보존합니다

## 여러 비공개 데이터셋을 활용한 AI 모델 학습

원시 데이터를 노출하지 않고 여러 출처의 민감한 데이터셋을 사용하여 AI 모델을 학습할 수 있게 하는 시스템입니다.

혜택: AI 역량을 강화하고 프라이버시 및 규제 준수를 보장합니다

## 협업형 데이터 분석

독점적이거나 민감하거나 비공개인 데이터를 공유하지 않고 여러 조직의 데이터셋을 공동으로 분석하는 방법입니다.

혜택: 민간/공공 기관 간 새로운 협업 가능성을 열고, 경쟁 우위를 유지하며, 데이터 프라이버시를 보존합니다

## 프라이버시 보호 설문조사

응답자의 익명성과 응답의 기밀성을 보장하는 설문조사 수행 방식입니다.

혜택: 프라이버시 보장을 통해 참여를 늘리고, 보복에 대한 두려움 없이 솔직함을 장려합니다

## 집단 기후 모델링

조직, NGO, 정부가 기후 모델에 대해 협업할 수 있는 방법이며, 특히 독점 모델이나 경쟁 데이터와 같은 민감한 내부 데이터셋을 어떠한 이유로든 비공개로 유지해야 하는 경우에 해당합니다.

혜택: 국경 간 협업을 가능하게 하고, 독점 모델과 경쟁 정보를 보호하며, 기후 모델링의 무결성과 정확성을 유지합니다.

## 점수 집계 시스템

신용 점수, 제품 리뷰, Michelin stars 또는 web-of-trust 네트워크와 같은 맥락에서 중앙 운영자에 의존하거나 개별 입력을 노출하지 않고 점수 또는 평점을 집계하는 시스템입니다.

혜택: 프라이버시를 보장하고, 강압과 조작을 방지하며, 점수 및 평점 시스템에서 결과의 정당성을 높입니다.

## 여러 서버 전반의 **Federated Learning**

로컬 데이터 샘플을 보유한 탈중앙화 기기 또는 서버 전반에서 AI 모델을 학습하는 방법은 연산 프로세스가 암호화되고 탈중앙화되도록 보장합니다.

혜택: 더 포괄적인 모델로 이어지며, Federated Learning에 남아 있는 프라이버시/보안 우려를 해결하고, 데이터 프라이버시를 유지합니다

# 기여자

Interfold 백서에 대한 초기 기여와 검토를 해주신 다음 분들께 특별히 감사드립니다: Alex Espinosa, Anthony Leutenegger, Casey Gardiner, Daniel Ospina, Disruption Joe, Gabriel Shapiro, Giacomo Corrias, Koh Wei Jie, Mike Chan, Pandy Marino, Vitalik Buterin, Yuet Loo Wong

1. RISC Zero는 범용 영지식 가상 머신입니다. 자세히 알아보기 ↩
2. SP1은 Succinc1. RISC Zero는 범용 영지식 가상 머신입니다. ↩
3. SP1은 Succinct의 영지식 가상 머신입니다. ↩

4. Jolt는 a16z의 확장 가능한 영지식 가상 머신입니다. ↩
5. Nexus는 모듈형, 확장형, 오픈소스, 고도로 병렬화된, 증명자 최적화 zkVM입니다. ↩
6. Reality.eth는 Reality Keys의 크라우드소싱 온체인 스마트 컨트랙트 오라클 시스템입니다. ↩
7. Kleros Oracle은 실제 세계 이벤트를 온체인에서 검증하기 위해 Kleros의 분쟁 해결 시스템과 Reality.eth의 암호경제적 메커니즘을 결합한 제품입니다. ↩
8. API3는 web3를 위한 탈중앙화 API를 만듭니다. ↩
9. 스테이킹 예치 수량은 Interfold 거버넌스에 의해 설정됩니다. ↩
10. 등록 지연 기간은 Interfold 거버넌스에 의해 설정됩니다. ↩
11. Decommission 지연 기간은 Interfold 거버넌스에 의해 설정됩니다. ↩
12. Ciphernode cliff 및 vesting 일정은 Interfold 거버넌스에 의해 설정됩니다. ↩
13. 비활성 노드 페널티 수량은 Interfold 거버넌스에 의해 설정됩니다. ↩
14. 암호문 출력을 제공하는 데 필요한 시간은 연산의 복잡도, 연산이 확장되는 방식, 입력 수에 의해 제약됩니다. ↩
15. 결과 보고 실패 비율은 Interfold 거버넌스에 의해 설정됩니다. ↩
16. Ciphernode는 선택된 Coordinator Node가 출력 암호문을 게시한 후 고정된 시간 창 내에 복호화된 출력을 제공해야 합니다. ↩
17. 동일한 입력은 항상 동일한 출력을 생성해야 합니다. ↩
18. 중간 slashing 페널티와 그 소각 비율은 Interfold 거버넌스에 의해 설정됩니다. ↩
19. Zodiac은 ERC-5005에 명시된 모듈형 및 조합 가능한 스마트 컨트랙트 계정의 표준입니다. 라이브러리와 구현은 GitHub에서 확인할 수 있습니다. ↩
20. Safe는 가장 널리 사용되는 스마트 컨트랙트 계정 구현체이며, 해당 코드는 GitHub에서 확인할 수 있습니다. ↩
21. Minimal Anti-Collusion Infrastructure ↩
22. https://ethresear.ch/t/minimal-anti-collusion-infrastructure/5413 ↩

*2026년 7월 18일에 마지막으로 업데이트됨*