



(19) 대한민국특허청(KR)
(12) 공개특허공보(A)

(11) 공개번호 10-2025-0090689
(43) 공개일자 2025년06월20일

(51) 국제특허분류(Int. Cl.)
H04L 9/32 (2006.01) G06Q 20/36 (2012.01)
H04L 9/00 (2022.01) H04L 9/08 (2006.01)
(52) CPC특허분류
H04L 9/3297 (2013.01)
G06Q 20/3678 (2013.01)
(21) 출원번호 10-2023-0180615
(22) 출원일자 2023년12월13일
심사청구일자 없음

(71) 출원인
고려대학교 산학협력단
서울특별시 성북구 안암로 145, 고려대학교 (안암동5가)
(72) 발명자
이중희
서울특별시 성북구 길음로9길 50, 907동 202호
김범중
경기도 군포시 수리산로 244, 996동 1301호
이동준
경기도 김포시 사우중로74번길 66, 129동 1302호
(74) 대리인
김준석, 박민욱

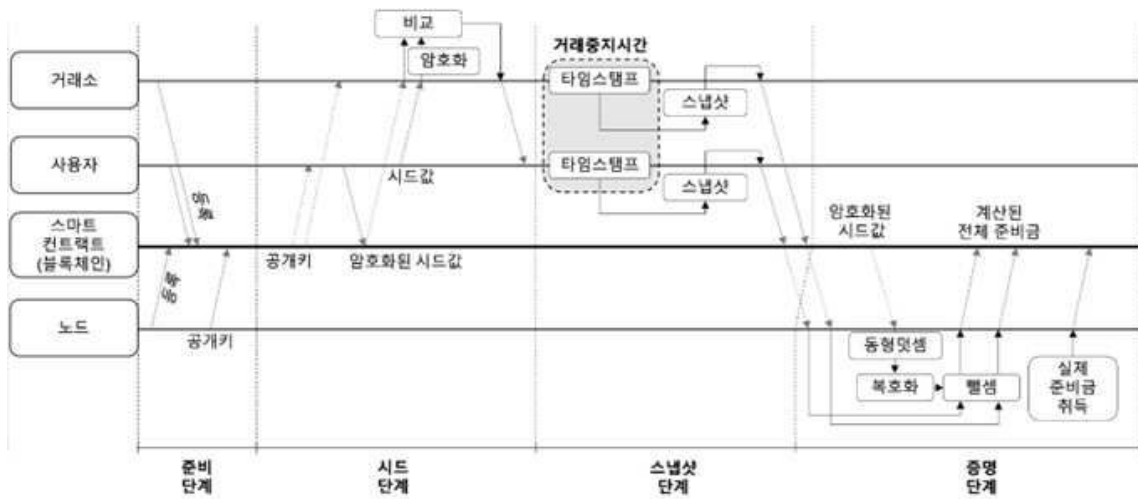
전체 청구항 수 : 총 4 항

(54) 발명의 명칭 중앙화 암호자산 거래소의 준비금 증명을 위한 양방향 재정 건전성 감사 프로토콜

(57) 요약

본 발명의 일 실시예에 의한 양방향 재정 건전성 감사 프로토콜은, 노드들이 ADKG 프로토콜을 이용하여 암호화 키 쌍을 생성하고, 공개키를 블록체인에 공개하는 준비 단계, 사용자가 생성한 유사난수 시드값이 공개키로 암호화된 시드값을 기초로 유효한 사용자들을 선별하는 시드 단계, 거래소는 거래중지시간을 발생시키고, 사용자 및 거래소는 상기 거래중지시간 발생에 따라 각자 기록한 시드값이 더해진 스냅샷을 블록체인에 공개하는 스냅샷 단계 및 스냅샷 및 암호화된 시드값을 기초로 준비금 증명을 확인할 수 있는 증명 단계를 포함할 수 있다.

대표도 - 도1



(52) CPC특허분류

H04L 9/008 (2013.01)

H04L 9/0825 (2013.01)

H04L 9/0869 (2013.01)

H04L 9/50 (2022.05)

이 발명을 지원한 국가연구개발사업

과제고유번호	1711194271
과제번호	2021-0-00518-003
부처명	과학기술정보통신부
과제관리(전문)기관명	정보통신기획평가원
연구사업명	데이터경제를위한블록체인기술개발
연구과제명	블록체인 데이터 암호화 기반의 프라이버시 보호 기술개발
기 여 율	100/100
과제수행기관명	한양대학교산학협력단
연구기간	2023.01.01 ~ 2023.12.31

명세서

청구범위

청구항 1

노드들이 ADKG 프로토콜을 이용하여 암호화 키 쌍을 생성하고, 공개키를 블록체인에 공개하는 준비 단계;

사용자가 생성한 유사난수 시드값이 상기 공개키로 암호화된 시드값을 기초로 유효한 사용자들을 선별하는 시드 단계;

거래소는 거래중지시간을 발생시키고, 사용자 및 거래소는 상기 거래중지시간 발생에 따라 각자 기록한 시드값이 더해진 스냅샷을 블록체인에 공개하는 스냅샷 단계; 및

상기 스냅샷 및 상기 암호화된 시드값을 기초로 준비금 증명을 확인할 수 있는 증명 단계; 를 포함하는

양방향 재정 건전성 감사 프로토콜.

청구항 2

제1항에 있어서,

상기 시드 단계는,

사용자가 생성한 유사난수 시드값을 상기 공개키로 암호화하여 블록체인에 공개하고, 거래소는 전달받은 상기 시드값을 직접 암호화한 뒤 상기 암호화된 시드값과 비교한 후, 비교 결과를 기초로 유효한 사용자들을 선별하는 것을 특징으로 하는

양방향 재정 건전성 감사 프로토콜.

청구항 3

제1항에 있어서,

상기 스냅샷 단계는,

거래소는 거래중지시간을 발생시키고, 사용자는 상기 거래중지시간이 발생했음을 고지받은 즉시 타임스탬프를 기록하고, 상기 타임스탬프가 가리키는 본인의 자산 총액을 기록하여 상기 시드값을 더해 제1 스냅샷으로 남기고, 거래소는 모든 유저에 대한 상기 타임스탬프를 거래중지시간 내에 기록하고, 상기 타임스탬프가 가리키는 유저에 대한 스냅샷을 기록하여 상기 시드값을 더하고, 사용자와 거래소는 각자 기록한 시드값이 더해진 제2 스냅샷을 블록체인에 공개하는 것을 특징으로 하는

양방향 재정 건전성 감사 프로토콜.

청구항 4

제3항에 있어서,

상기 증명 단계는,

노드들은 상기 암호화된 시드값에 대해 모든 유저들에 대하여 동형 덧셈을 수행하고, 덧셈의 결과값을 ADKG 복호화 프로토콜로 복호화하고, 복호화된 시드값들의 합을 상기 제1 스냅샷의 합 및 상기 제2 스냅샷의 합 각각에 대해 뺄셈을 수행하여 이를 기초로 준비금 증명을 확인하는 것을 특징으로 하는

양방향 재정 건전성 감사 프로토콜.

발명의 설명

기술분야

[0001] 본 발명은 암호자산(가상자산)에 관한 것이며, 보다 상세하게는 중앙화 암호자산 거래소(가상자산 거래소)의 준비금 증명을 위해 사용자와 거래소 간의 양방향 프로토콜을 통해 재정 건전성을 감사하는 기술에 관한 것이다.

배경 기술

[0002] 중앙화 암호자산 거래소가 사용자들이 거래소 내에서 보유중인 자산을 실제로 보유하고 있음을 증명해야 할 필요성이 대두됨에 따라, 해외에 위치한 크라켄(Kraken)과 바이낸스(Binance) 같은 거래소는 준비금 증명을 공개하기 시작하였다.

[0003] 준비금 증명은 거래소를 이용중인 각 사용자의 자산 및 전체 사용자들의 자산을 보유하고 있음을 머클트리(Merkle tree)와 같은 기술을 통해 수학적으로 증명하는 것으로, 거래소는 이를 공개함으로써 사용자들이 본인들의 자산에 대해 침해가 없음을 보장받을 수 있게 된다.

[0004] 위와 같은 노력에도 불구하고, 현존하는 준비금 증명 기술은 거래소의 사용자들이 그 결과를 완전히 신뢰하기에 부족함이 존재한다.

[0005] 바이낸스, 크라켄 등의 거래소에서 머클 트리(Merkle tree)를 활용한 준비금 증명 프로토콜을 개발, 이용 중이나, 이들에는 ‘스냅샷 체리피킹(snapshot cherry-picking)’이 가능하다는 한계점이 존재한다.

[0006] 예를 들어, 준비금 증명을 위해서는 특정 시점에 각 사용자에게 대한 스냅샷(자산내역의 기록)을 기록한 뒤, 모든 사용자의 스냅샷 상 자산의 합과 해당 시점에 거래소가 실제로 보유중인 블록체인 상 자산을 비교한다. 위의 자산 합과 실제 자산의 비교 결과, 실제 자산이 사용자들의 자산 합보다 3퍼센트 클 경우, “103 퍼센트”와 같은 증명 결과를 사용자들에게 공개한다. 그러나, 스냅샷을 찍는 시점이 0시 0분 0초일 경우, 해당 시점에도 사용자들 간의 거래가 오고갈 수 있다. 예를 들어, 스냅샷을 찍는 시점에 사용자 A가 사용자 B에게 특정 자산을 100만큼 판매하였을 경우, 사용자 A와 B에 대한 두 개의 스냅샷은 사용자 A, B의 거래 이전 혹은 거래 이후를 완벽히 반영해야 한다. 그러나, 준비금 증명을 수행하는 주체(거래소 혹은 외부 감사단체)가 스냅샷을 체리피킹(원하는 것만 고르는 행위)하여 사용자 A의 판매 후 시점, 사용자 B의 판매 전 시점에 대해 기록할 경우, 사용자들의 자산 합은 실제 사용자들이 인지하고 있는 것 대비 100만큼이 부족하게 된다. 거래소의 준비금이 부족한 경우 스냅샷 체리피킹을 수행, 결과로 사용자들의 자산 합은 실제 대비 100만큼이 부족한 수치를 보이게 되며, 준비금이 부족한 것을 감출 수 있다. (예를 들어, 거래소의 준비금이 70만큼 부족한 경우에도 위와 같은 조작을 통해 30만큼 초과한 준비금을 보유중인 것으로 보일 수 있음)

발명의 내용

해결하려는 과제

[0007] 본 발명은 상기 문제점을 해결하고자, 블록체인을 활용한 양방향 재무 건전성 감사 프로토콜(Bidirectional Financial Soundness Audit Protocol)을 제공하는 것을 목표로 한다.

과제의 해결 수단

[0008] 본 발명의 일 실시예에 의한 양방향 재정 건전성 감사 프로토콜은, 노드들이 ADKG 프로토콜을 이용하여 암호화 키 쌍을 생성하고, 공개키를 블록체인에 공개하는 준비 단계, 사용자가 생성한 유사난수 시드값이 공개키로 암호화된 시드값을 기초로 유효한 사용자들을 선별하는 시드 단계, 거래소는 거래중지시간을 발생시키고, 사용자 및 거래소는 상기 거래중지시간 발생에 따라 각자 기록한 시드값이 더해진 스냅샷을 블록체인에 공개하는 스냅샷 단계 및 스냅샷 및 암호화된 시드값을 기초로 준비금 증명을 확인할 수 있는 증명 단계를 포함할 수 있다.

[0009] 또한, 일 실시예에 따라서, 시드 단계는, 사용자가 생성한 유사난수 시드값을 공개키로 암호화하여 블록체인에 공개하고, 거래소는 전달받은 시드값을 직접 암호화한 뒤 암호화된 시드값과 비교한 후, 비교 결과를 기초로 유효한 사용자들을 선별하는 것을 특징으로 할 수 있다.

[0010] 또한, 일 실시예에 따라서, 스냅샷 단계는, 거래소는 거래중지시간을 발생시키고, 사용자는 거래중지시간이 발생했음을 고지받은 즉시 타임스탬프를 기록하고, 타임스탬프가 가리키는 본인의 자산 총액을 기록하여 시드값을 더해 제1 스냅샷으로 남기고, 거래소는 모든 유저에 대한 타임스탬프를 거래중지시간 내에 기록하고, 타임스탬프가 가리키는 유저에 대한 스냅샷을 기록하여 시드값을 더하고, 사용자와 거래소는 각자 기록한 시드값이 더해진 제2 스냅샷을 블록체인에 공개하는 것을 특징으로 할 수 있다.

[0011] 또한, 일 실시예에 따라서 증명 단계는, 노드들은 암호화된 시드값에 대해 모든 유저들에 대하여 동형 덧셈을 수행하고, 덧셈의 결과값을 ADKG 복호화 프로토콜로 복호화하고, 복호화된 시드값들의 합을 제1 스냅샷의 합 및 제2 스냅샷의 합 각각에 대해 뺄셈을 수행하여 이를 기초로 준비금 증명을 확인하는 것을 특징으로 할 수 있다.

발명의 효과

[0012] 본 발명에서 제안하는 양방향 재무 건전성 감사 프로토콜(Bidirectional Financial Soundness Audit Protocol)은 블록체인을 활용, 전략적인 거래 정지 시간을 도입하고 감사 과정에 유저들이 직접 참여하도록 함으로써 기존의 방법에서 드러나는 한계점을 보완할 수 있다.

도면의 간단한 설명

[0013] 도 1은 일 실시예에 의한 전체 흐름도이다.
 도 2는 일 실시예에 의한 시드 단계 및 스냅샷 단계의 흐름도이다.
 도 3은 일 실시예에 의한 EDH 방식을 통한 거래중시시간 흐름도이다.
 도 4는 일 실시예에 의한 UDH 방식을 통한 거래중시시간 흐름도이다.
 도 5 내지 도 17은 본 발명의 양방향 재무 건전성 감사 프로토콜 관련 논문이다.

발명을 실시하기 위한 구체적인 내용

[0014] 본 발명의 이점 및 특징, 그리고 그것들을 달성하는 방법은 첨부되는 도면과 함께 후술되어 있는 실시예들을 참조하면 명확해질 것이다. 그러나 본 발명은 이하에서 개시되는 실시예들에 한정되는 것이 아니라 서로 다른 다양한 형태로 구현될 수 있으며, 단지 본 실시예들은 본 발명의 개시가 완전하도록 하고, 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자에게 발명의 범주를 완전하게 알려주기 위해 제공되는 것이며, 본 발명은 청구항의 범주에 의해 정의될 뿐이다.

[0015] 본 명세서에서 사용되는 용어에 대해 간략히 설명하고, 본 발명에 대해 구체적으로 설명하기로 한다.

[0016] 본 발명에서 사용되는 용어는 본 발명에서의 기능을 고려하면서 가능한 현재 널리 사용되는 일반적인 용어들을 선택하였으나, 이는 당 분야에 종사하는 기술자의 의도 또는 관례, 새로운 기술의 출현 등에 따라 달라질 수 있다. 또한, 특정한 경우는 출원인이 임의로 선정한 용어도 있으며, 이 경우 해당되는 발명의 설명 부분에서 상세히 그 의미를 기재할 것이다. 따라서 본 발명에서 사용되는 용어는 단순한 용어의 명칭이 아닌, 그 용어가 가지는 의미와 본 발명의 전반에 걸친 내용을 토대로 정의되어야 한다.

[0017] 명세서 전체에서 어떤 부분이 어떤 구성요소를 "포함"한다고 할 때, 이는 특별히 반대되는 기재가 없는 한 다른 구성요소를 제외하는 것이 아니라 다른 구성요소를 더 포함할 수 있음을 의미한다. 또한, 명세서에서 사용되는 "부", "모듈", "유닛" 등의 용어는 적어도 하나의 기능 또는 동작을 처리하는 단위를 의미하며, 소프트웨어, FPGA 또는 ASIC과 같은 하드웨어 구성요소, 또는 소프트웨어와 하드웨어의 결합으로 구현될 수 있다. 그렇지만 "부", "모듈", "유닛" 등의 용어가 소프트웨어 또는 하드웨어에 한정되는 의미는 아니다. "부", "모듈", "유닛" 등은 어드레싱할 수 있는 저장 매체에 있도록 구성될 수도 있고 하나 또는 그 이상의 프로세서들을 재생시키도록 구성될 수도 있다. 따라서, 일 예로서 "부", "모듈", "유닛" 등의 용어는 소프트웨어 구성요소들, 객체지향 소프트웨어 구성요소들, 클래스 구성요소들 및 태스크 구성요소들과 같은 구성요소들과, 프로세스들, 함수들, 속성들, 프로시저들, 서브루틴들, 프로그램 코드의 세그먼트들, 드라이버들, 펌웨어, 마이크로 코드, 회로, 데이터, 데이터베이스, 데이터 구조들, 테이블들, 어레이들 및 변수들을 포함한다.

[0018] 아래에서는 첨부한 도면을 참고하여 본 발명의 실시예에 대하여 본 발명이 속하는 기술 분야에서 통상의 지식을 가진 자가 용이하게 실시할 수 있도록 상세히 설명한다. 그리고 도면에서 본 발명을 명확하게 설명하기 위해서 설명과 관계없는 부분은 생략한다. "제1", "제2" 등과 같이 서수를 포함하는 용어는 다양한 구성 요소들을 설명하는데 사용될 수 있지만, 구성 요소들은 용어들에 의해 한정되지는 않는다. 용어들은 하나의 구성요소를 다른 구성요소로부터 구별하는 목적으로만 사용된다. 예를 들어, 본 발명의 권리 범위를 벗어나지 않으면서 제1 구성요소는 제2 구성요소로 명명될 수 있고, 유사하게 제2 구성요소도 제1 구성요소로 명명될 수 있다. "및/또는" 이라는 용어는 복수의 관련된 항목들의 조합 또는 복수의 관련된 항목들 중의 어느 하나의 항목을 포함한다.

- [0019] 본 명세서에 설명된 다양한 시스템, 장치, 저장 매체, 모듈 및 기구는 전용 연산 디바이스, 또는 하나 이상의 전용 연산 디바이스의 하나 이상의 연산 칩에서 구현될 수도 있다.
- [0021] 이하, 본 발명의 양방향 재정 건전성 감사 프로토콜에 관하여 설명하도록 한다.
- [0022] 본 발명에는 ADKG(asynchronous distributed key generation)를 기반으로 암호화 시스템을 구성할 수 있다.
- [0023] 분산 키 생성(DKG; distributed key generation)은 쓰레시홀드 암호화 시스템(threshold cryptosystem)의 시작을 용이하게 하는 동시에 중앙화된 신뢰가 필요한 주체의 필요성을 회피하는 근본적인 기술로 부상하고 있으며, 쓰레시홀드 서명과 다자간 계산을 포함한 탈중앙화 프로토콜의 기반에까지 그 중요성이 확대되고 있다.
- [0024] 비동기 분산 키 생성(ADKG)은 동기화를 필요로하지 않는 분산 키 생성 방식으로, 다자 간(노드들 간) 저수준의 신뢰만을 요구하는 암호화 시스템이다. 비동기 분산 키 생성(ADKG)은 Das et al.의 연구(Das et al., Practical Asynchronous Distributed Key Generation, in IEEE Symp. Secur. Priv., 2022.) 등에서 제안된 바 있으며, 해당 연구는 ElGamal 암호화와 같은 이산 로그 기반 암호화 시스템에 기반을 둔다. 특히 복원력이 뛰어난 이 프로토콜은 악성 노드의 임계값 수인 m 과 노드 수인 n 을 가정했을 때 최대 $m < n/3$ 의 악성 노드가 존재해도 견딜 수 있으며, 생성된 공유 무작위성과 최종 키를 구성하는 개별 공유 계산의 무결성을 보호할 수 있다.
- [0025] 본 발명은 ADKG를 채택함으로써, 분산 키 생성 기능을 활용하여 네트워크 내의 여러 노드가 공동으로 암호화 키를 구성할 수 있도록 지원할 수 있다. 사용자가 제공한 민감한 정보를 안전하게 암호화하는 데 사용되는 이 키는 동형 덧셈의 탈중앙화 계산을 위한 기반이 될 수 있다. 신뢰할 수 있는 제3자에 의존하지 않기 때문에 프레임워크의 신뢰 없는 아키텍처가 강화될 수 있다.
- [0027] <수식 정의>
- [0028] 본 발명의 양방향 재정 건전성 감사 프로토콜의 효과를 기술하기 위해 다음과 같은 정의를 도입한다.
- [0029] tsp^i : i 번째 사용자의 디바이스에서 기록하는 타임스탬프 값으로, 거래중지시간 내의 특정 시각을 가리킴
- [0030] tsp^{exch} : 거래소 서버에서 기록하는 타임스탬프 값으로, 거래중지시간 내의 특정 시각을 가리킴
- [0031] $b_k^i |_{tsp^i}$: 사용자 디바이스에서 기록된 타임스탬프 tsp^i 가 가리키는 i 번째 사용자의 자산 k (예를 들어, 비트코인)에 대한 자산액. i 번째 사용자가 인지하고 있는 본인의 자산 k 에 대한 자산액을 의미함.
- [0032] $b_k^i |_{tsp^{exch}}$: 거래소 서버에서 기록된 타임스탬프 tsp^{exch} 가 가리키는 i 번째 사용자의 자산 k (예를 들어, 비트코인)에 대한 자산액. 거래소가 인지하고 있는 i 번째 사용자의 자산 k 에 대한 자산액을 의미함.
- [0033] B_k : 거래소가 실제로 보유중인 자산 k 에 대한 자산액. 해당 금액은 거래소의 소유로 밝혀진 자산 k 지급의 잔액을 확인해봄으로써 누구나 쉽게 계산 가능함.
- [0034] 위와 같은 정의를 이용, 본 발명을 통해 거래소의 사용자들은 다음의 등식 및 부등식의 유효함을 확인해볼 수 있다.
- [0035] $b_k^i |_{tsp^i} = b_k^i |_{tsp^{exch}}, \quad \forall k$: 모든 k 에 대하여, i 번째 사용자의 자산 k 에 대한 자산액에 대해, 거래소가 인지하고 있는 값과 사용자가 인지하고 있는 값이 같음을 의미한다.
- [0036] $\sum_i b_k^i |_{tsp^i} \leq B_k, \quad \forall k$: 모든 k 에 대하여, 거래소가 실제로 보유중인 자산 k 에 대한 자산액이 각 사용자들이 인지하고 있는 자산 k 에 대한 자산액의 총합보다 같거나 크음을 의미한다.
- [0038] 일 실시예에 의하면, 본 발명의 양방향 재무 건전성 감사 프로토콜은 준비 단계, 시드 단계, 스냅샷 단계 및 증명 단계를 포함할 수 있다.

[0039] 이하 각 단계에 관하여 설명하도록 한다.

[0041] <준비 단계>

[0042] 준비 단계는 노드들이 ADKG 프로토콜을 이용하여 암호화 키 쌍을 생성하고, 공개키를 블록체인에 공개할 수 있다.

[0043] 본 프로토콜에서는 감사를 담당하는 노드(node)들이 존재하며, 노드들은 ADKG(asynchronous distributed key generation) 프로토콜을 이용해 중개자 없이 암호화 키 쌍을 생성, 공개키를 블록체인에 공개해 사용자 및 거래소가 이후 과정에서 활용할 수 있도록 할 수 있다.

[0045] <시드 단계>

[0046] 시드 단계는 사용자가 생성한 유사난수 시드값이 공개키로 암호화된 시드값을 기초로 유효한 사용자들을 선별할 수 있다.

[0047] 사용자는 본인의 자산 금액을 대중으로부터 감추기 위해 유사난수 시드값(pseudo-random seed value)를 생성 후 노드가 제공한 공개키로 암호화하여 블록체인에 공개할 수 있다.

[0048] i번째 사용자의 k번째 자산에 대한 시드값을 s_k^i 라고 정의 시, 모든 자산(최대 k개)에 대한 시드값은 다음과 같은 벡터 형태로 표기할 수 있다.

[0049]
$$\vec{s}^i = (s_1^i, s_2^i, \dots, s_K^i)$$

[0050] i번째 사용자에 대해 시드값을 암호화한 값은 다음과 같은 벡터 형태로 표기할 수 있다.

[0051]
$$\vec{E} = (enc(s_1^i), enc(s_2^i), \dots, enc(s_K^i))$$

[0052] 사용자는 거래소에게 암호화되지 않은 시드값을 전달하고, 거래소는 전달받은 시드값이 유효한지 확인하기 위해 전달받은 시드값을 직접 암호화한 뒤 블록체인에 공개된 해당 유저의 암호화된 시드값과 비교해 볼 수 있다.

[0053] 거래소는 시드값 유효성 검사의 결과를 블록체인에 공개하며, 유효한 사용자들은 악의적이지 않은 사용자로 간주해 이후의 감사 과정을 수행할 수 있도록 할 수 있다.

[0055] <스냅샷 단계>

[0056] 스냅샷 단계에서 거래소는 거래중지시간을 발생시키고, 사용자 및 거래소는 거래중지시간 발생에 따라 각자 기록한 시드값이 더해진 스냅샷을 블록체인에 공개할 수 있다.

[0057] 거래소는 전략적 거래중지시간을 발생시킬 수 있다.

[0058] 거래중지시간을 구성하는 방법으로 거래소가 직접 모든 사용자들에게 거래중지의 사실을 고지하는, “거래소에 의한 핸드셰이크(EDH; exchange-driven handshake)”와 거래소가 일부 사용자들에게 사실을 고지 후 유저들이 다른 사용자들에게 전파하는 “사용자에 의한 핸드셰이크(UDH; user-driven handshake)” 두 가지 방식을 제안한다.

[0059] EDH는 거래소 서버에서 각 사용자들에게 SYN 패킷(거래중지의 사실을 고지하는 역할을 수행)을 전송하고, 그에 따라 각 사용자는 SYN-ACK 패킷 - (거래중지의 사실을 고지받았음을 알리는 역할을 수행)을 거래소 서버에 전송하고, 거래소 서버는 ACK 패킷(SYN-ACK 패킷을 올바르게 수신하였음을 고지하는 역할을 수행)을 각 사용자에게 전송하는 구조를 가질 수 있다.

[0060] EDH에서 거래중지시간은 거래중지를 시행한 뒤 거래소가 모든 사용자들에게 SYN 패킷을 전송한 뒤 모든 사용자들로부터 SYN-ACK 패킷을 수신하기까지 걸리는 시간만큼 지속할 수 있다.

[0061] UDH는 거래소 서버에서 정해진 수만큼의 이웃하는 사용자들(예를 들어, 10명의 사용자)에게 SYN 패킷을 전송하고, SYN 패킷을 수신한 사용자들은 SYN-ACK 패킷으로 거래소 서버에게 응답하고, 응답 직후 또 다른 이웃하는 사용자들에게 SYN 패킷을 전송하는 구조를 가질 수 있다. 이와 같은 전달이 정상적으로 이루어졌음을 고지하기 위해 사용자는 본인에게 SYN 패킷을 전송한 당사자(이웃하는 사용자 혹은 거래소 서버)에게 ACK 패킷을 전송할 수 있다.

[0062] UDH에서 거래중지시간은 거래중지를 시행한 뒤 거래소 서버가 모든 사용자들로부터 SYN-ACK 패킷을 전달받기까지의 시간만큼 지속할 수 있다.

[0063] 거래의 중지로 인해 거래의 발생 자체가 불가능한 이 시점, 사용자들은 거래중지시간이 발생했음을 고지받은 즉시(SYN 패킷을 받은 즉시) 타임스탬프(시간값)를 기록할 수 있다.

[0064] 사용자들은 거래중지시간이 진행되고 있는지 여부와 무관하게, 타임스탬프가 가리키는 본인의 자산 총액을 기록, 시드값을 더해 스냅샷으로 남길 수 있다.

[0065] 거래중지시간이 종료된 이후일지라도 타임스탬프는 거래중지시간 내의 시간을 가리키므로 이때를 기준으로 한 스냅샷은 스냅샷 체리피킹 문제가 발생하지 않을 수 있다.

[0066] i 번째 사용자의 k 번째 자산에 대한 자산액값을 b_k^i 라고 정의 시, 모든 자산(최대 k 개)에 대한 자산액은 다음과 같은 벡터 형태로 표기할 수 있다.

[0067]
$$\vec{b}^i = (b_1^i, b_2^i, \dots, b_K^i)$$

[0068] 사용자들이 타임스탬프를 기록한 것과 같은 방법으로 거래소 또한 모든 유저에 대한 타임스탬프를 거래중지시간 내에 기록, 이 타임스탬프들이 가리키는 각 유저에 대한 스냅샷을 기록, 시드값을 더할 수 있다.

[0069] i 번째 사용자의 자산액에 시드값이 더해진 스냅샷의 값은 다음과 같은 벡터 형태로 표기할 수 있다.

[0070]
$$\vec{S}^i = \vec{b}^i + \vec{s}^i$$

[0071] 사용자들과 거래소는 각자가 기록한, 시드값이 더해진 스냅샷을 블록체인에 공개할 수 있다.

[0073] <증명 단계>

[0074] 증명 단계는 스냅샷 및 암호화된 시드값을 기초로 준비금 증명을 확인할 수 있다.

[0075] 감사자이자 블록체인을 구동하는 역할을 수행하는 노드(node)들은 스냅샷 단계가 종료되는 대로, 블록체인에 공개된 각 사용자에 대한 암호화된 시드값과 사용자들 및 거래소가 공개한 스냅샷 정보들을 블록체인으로부터 내려받을 수 있다.

[0076] 각 노드는 암호화된 시드값 $(\vec{E}^i = (enc(s_1^i), enc(s_2^i), \dots, enc(s_K^i)))$ 에 대해 모든 유저들에 대해 다음 수학적 1의 동형 덧셈(homomorphic addition)을 수행, 덧셈의 결과값을 노드들간의 ADKG 복호화 프로토콜을 통해 복호화할 수 있다.

수학적 1

[0077]
$$(\sum_i \vec{E}^i = (\sum_i enc(s_1^i), \sum_i enc(s_2^i), \dots, \sum_i enc(s_K^i)))$$

[0078] 노드는 복호화된 시드값들의 합을 스냅샷 값의 합에서 뺄셈을 수행(사용자들이 공개한 스냅샷의 합 및 거래소가 공개한 스냅샷의 합 각각에 대해서), 각 값을 ‘사용자들이 주장하는 자산의 합’ 및 ‘거래소가 주장하는 자산의 합’으로 간주할 수 있다.

[0079] 노드들은 거래소가 보유중인 블록체인상의 실제 자산 총액을 구한 뒤, 이 실제 자산 총액(B_k)과 앞에서 구한 사용자들이 주장하는 자산의 합($\sum_i b_k^i |_{tsP^i}$) 및 거래소가 주장하는 자산의 합($\sum_i b_k^i |_{tsP^{exh}}$)을 비교해 봄으로써 전체 사용자에게 대한 준비금 증명을 확인해 볼 수 있다.

[0080] $\sum_i b_k^i |_{tsP^i} \leq B_k, \forall k$: 모든 k 에 대하여, 거래소가 실제로 보유중인 자산 k 에 대한 자산액이 각 사용자들이 인지하고 있는 자산 k 에 대한 자산액의 총합보다 같거나 큰지 확인해 볼 수 있다.

[0081] 각 사용자는 블록체인에 공개된 본인의 자산에 대한 두 개의 스냅샷(i 번째 사용자에게, 본인이 주장하는 스냅샷 $\vec{S}_{user}^i$, 그리고 거래소가 주장하는 스냅샷 $\vec{S}_{exch}^i$)을 비교해 봄으로써 사용자 개인에 대한 준비금 증명을 확인해 볼 수 있다.

[0082] 두 개 스냅샷의 동일성을 확인해보는 것($\vec{S}_{user}^i = \vec{S}_{exch}^i$ 인지 확인)은 개인의 자산 총액에 대한 사용자 주장 값과 거래소 주장 값의 비교($b_k^i |_{tsP^i} = b_k^i |_{tsP^{exh}}, \forall k$)와 동등할 수 있다.

[0083] 각 사용자는 거래소가 주장하는 자산의 합($\sum_i b_k^i |_{tsP^{exh}}$)을 거래소와 실제 자산 총액(B_k)과 비교해 봄으로써 거래소의 주장과 실제의 괴리를 확인해볼 수 있다.

[0085] <실시예>

[0086] 제안한 두 가지 방식에 대해, 3억 명의 사용자 및 사실고지를 위한 패킷의 크기로 512비트, 10Gbps의 네트워크 처리량을 실험값으로 사용해 시뮬레이션(OMNet++를 이용함)을 이용한 실험을 수행하여, 거래중지시간이 거래소의 생태계에 부정적인 영향을 미칠 정도가 아닌지를 확인해 보았다.

[0087] -EDH에 대한 실험 값 설명

[0088] EDH를 통한 거래중지시간은 ‘사용자 수’를 ‘패킷의 크기’와 곱한 값에서 ‘초당 네트워크 처리량’을 나눈 값에 ‘1명의 사용자에게 대한 응답(SYN-ACK 패킷을 통한) 대기 시간’과 같다. 여기서 사용자 수를 패킷의 크기와 곱한 값은 전송해야 할 총 데이터의 크기를 의미하며, 거래소의 서버가 감당가능한 초당 네트워크 처리량으로 이것을 나누면 거래소 서버가 SYN 패킷을 전부 전송하기까지 걸리는 시간을 구할 수 있다. 한편, 사용자들로부터 SYN-ACK 패킷을 전부 수신할 시 거래중지시간이 중단되는데, 거래소가 SYN 패킷을 보내는 동안 해당 패킷을 수신한 사용자들은 수신 즉시 SYN-ACK 패킷을 거래소 서버로 전송할 것이다. 따라서 거래소 서버는 SYN 패킷을 전송함과 동시에 사용자들의 SYN-ACK 패킷을 수신할 것이며 SYN 패킷 전송이 끝날 때 1명의 사용자에게 대한 응답(SYN-ACK 패킷)이 추가적으로 수신되어야 하기에 이에 대한 대기시간을 더하면 EDH를 통한 거래중지시간을 계산할 수 있게 되는 것이다.

[0089] 3억 명의 사용자, 512비트의 패킷 크기, 10Gbps의 초당 네트워크 처리량을 가정 시 15.36초로 계산이 가능하다.

[0090] 물론, 위와 같은 패킷의 전송에 더해 네트워크 지연시간이 계산되어야 할 것이다. 네트워크 지연시간은 transmission delay, processing delay, propagation delay, queueing delay 총 네 가지 delay에 대한 합으로 표현 가능한데, queueing delay는 통상적으로 마이크로초에서 밀리초 단위, propagation delay는 통상적으로 밀리초 단위임이 알려져있다. 이와 같은 통상 값에 더해, OMNet++를 통해 142,642개의 패킷을 전송해보는 실험을 해본 결과로 processing delay 및 transmission delay를 계산해본 결과(평균값을 채택) 각각 1.244 밀리초와 0.068 밀리초임을 확인하였다. 통상값 및 실험값을 전부 더하면 네트워크 지연시간을 구할 수 있으며, 통상값을 최대치로 가정 시 네트워크 지연시간은 3.312 밀리초임을 확인해볼 수 있다.

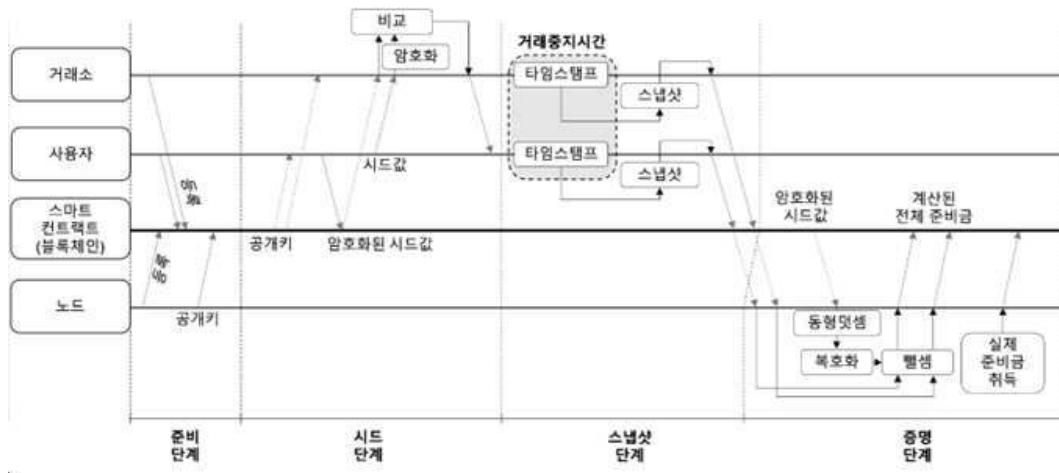
[0091] 네트워크 지연시간 값은 위에서 계산한 실질적 거래중지시간 대비 무시가 가능한 수준이므로, EDH를 통한 거래중지에 필요 시간은 약 15.36초라고 결론지을 수 있다.

[0092] -UDH에 대한 실험 값 설명

- [0093] EDH의 수행에 필요한 시간과는 반대로, UDH에서는 네트워크 지연시간 대비 패킷 전송시간이 무시 가능한 수준이다. UDH는 패킷의 전파가 사용자들에 의해 이루어지며, 한명의 사용자당 이웃하는 10명에게의 전파가 이루어진다고 가정할 시, 총 9번의 전파 과정이 반복될 경우 모든 사용자 3억 명에 대한 패킷 전파가 완료되기 때문이다 (3억은 10의 8제곱보다 크며 10의 9제곱보다 작기 때문).
- [0094] 실제로, EDH에서 사용된 레퍼런스 값(512 비트의 패킷 크기, 10Gbps의 초당 네트워크 처리량)을 가정시 10명의 사용자에게 패킷을 전송하는 데에 걸리는 시간은 0.512 마이크로초 수준이다.
- [0095] 반면, 총 10번의 전파과정이 반복되는데 여기에는 각각 EDH에서 계산한 것과 같은 네트워크 지연시간이 반복되어 소요되므로 총 33.12 밀리초의 네트워크 지연시간이 발생한다. 따라서, 패킷 전송에 걸리는 시간은 네트워크 지연시간에 비해 무시가 가능하다.
- [0096] 따라서, 실험을 통해 얻은 UDH를 통한 거래중지 필요 시간은 약 33.12 밀리초라고 결론지을 수 있다.
- [0097] -EDH 및 UDH에 대한 실험 값 해석
- [0098] EDH를 통한 거래중지시간으로도 시중 은행, 주식거래소 등에서 활용하는 거래중지시간보다 현저히 낮아 거래소 생태계에 악영향을 끼치지 않을 것으로 판단되며, UDH는 이보다도 훨씬 낮은 시간을 필요로 해 UDH를 활용하는 것을 권장할 수 있다.
- [0100] 위를 통해 사용자들은 본인이 인지하고 있는 본인들의 자산 총액과 거래소가 주장하는 자산 총액을 개인적으로 (위의 등식), 그리고 모든 사용자들의 값 총합에 대해(위의 부등식) 확인해볼 수 있다.
- [0101] 준비금 증명에 있어 거래소가 일방적으로 제시하는 값만을 신뢰하는 것이 아니라, 사용자들이 인지하고 있는 본인들의 자산 총액을 직접 반영해 거래소가 주장하는 값과 비교해볼 수 있으므로, 거래소에 대한 신뢰 문제를 해결할 수 있다.
- [0102] 이와 같은 양방향 프로토콜을 위해 전략적 거래중지시간을 통해 각 사용자들이 인지하고 있는 본인의 자산과 거래소가 주장하는 자산이 같을 수밖에 없는 환경을 조성하였고, 이를 통해 스냅샷 채리피킹(snapshot cherry-picking)을 방지할 수 있다.
- [0103] 거래중지시간은 충분히 짧은 시간 이내에 수행될 수 있음을 실험을 통해 보였으며, 이를 통해 거래소의 거래 생태계에 미칠 부정적인 영향을 최소화할 수 있다.
- [0104] 거래중지시간은 수많은 사용자들이 존재하는(예를 들어, 3억 명) 환경에서도 네트워크 지연시간과 같은 문제가 생기지 않도록 하는 한편 거래소-사용자 간의 양방향 프로토콜을 통해 거래소의 재정 건전성 감사가 이루어질 수 있도록 할 수 있다.
- [0106] 본원 발명의 실시예 등과 관련된 기술 분야에서 통상의 지식을 가진 자는 상기 기재의 본질적인 특성에서 벗어나지 않는 범위에서 변형된 형태로 구현될 수 있음을 이해할 수 있을 것이다. 그러므로, 개시된 방법들은 한정적인 관점이 아닌 설명적 관점에서 고려되어야 한다. 본 발명의 범위는 발명의 상세한 설명이 아닌 특허청구 범위에 나타나며, 그와 동등한 범위 내에 있는 모든 차이점은 본 발명의 범위에 포함되는 것으로 해석되어야 한다.

도면

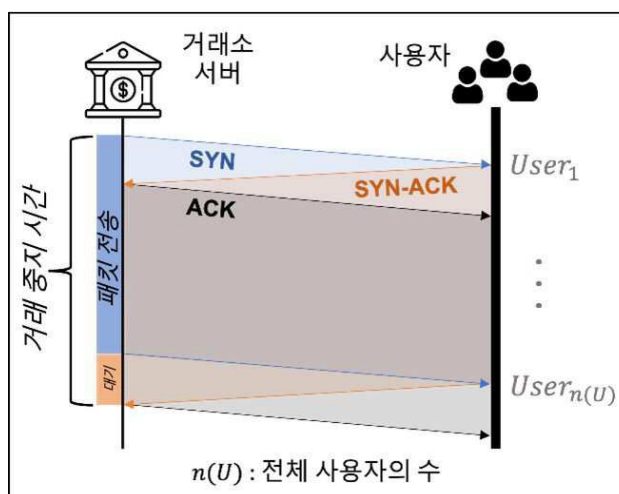
도면1



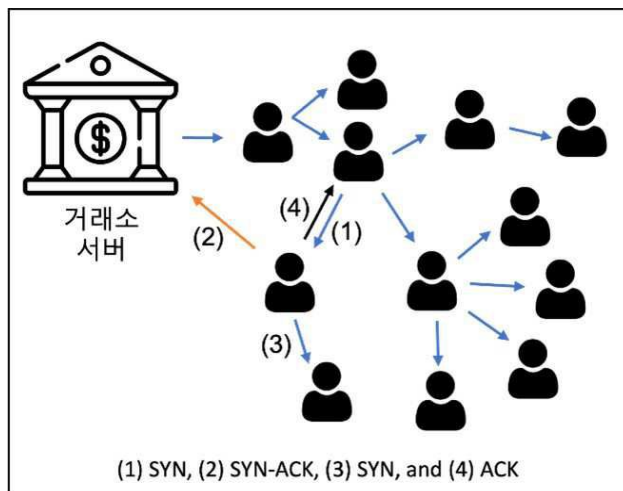
도면2



도면3



도면4



Bi-FiSA: Bidirectional Financial Soundness Audit Protocol for The Proof of Reserves of Centralized Crypto Asset Exchanges

Abstract

The crypto asset industry's rapid expansion has heightened the need for transparency and reliability within centralized crypto asset exchanges (CEXs). While some proof of reserves (PoR) protocols are introduced, reliability issues with the auditing process still exist. Specifically, if the CEX is malicious, it may cherry-pick the snapshots of user accounts to inflate its reserves. In this paper, we present bidirectional financial soundness audit (Bi-FiSA) protocol, a novel approach that users of CEXs participate proactively in the auditing process for mitigating trust issues. Bi-FiSA leverages asynchronous distributed key generation (ADKG), unique random seed values, and strategic pause periods to provide user privacy, bidirectional audit, and minimize trading disruptions. We evaluate Bi-FiSA's applicability regarding the minimization of trading disruptions with a reference user number of 300 million, by implementing exchange-driven handshakes (EDH) and user-driven handshakes (UDH) pause period network models. We also assess privacy measures with the ADKG protocol and unique random seed values, demonstrating that Bi-FiSA does not compromise user privacy. In summary, Bi-FiSA offers a promising solution to CEX auditing, ensuring financial integrity, fostering user trust, and enhancing transparency in the crypto asset industry.

1 Introduction

In December 2022, the CEO of a centralized crypto asset exchange was arrested in the Bahamas¹ with criminal charges including wire fraud, wire fraud conspiracy, securities fraud, securities fraud conspiracy, and money laundering. The exchange has been suspected of using clients' assets without their consent ever since it was first questioned about the balance sheet. Due to its severity, the incident served as a wake-up call to the crypto asset industry of the importance of exchange auditing.

¹<https://www.nytimes.com/2022/12/12/business/ftx-sam-bankman-fried-bahamas.html>, last accessed July 1, 2023

The blockchain is also known as distributed ledger technology (DLT). All user transactions within the blockchain platform are recorded in a ledger, after being verified by the nodes that comprise the blockchain. Meanwhile, users that wish to sell or buy assets usually trade assets through marketplaces called exchanges. The types of exchange are usually divided into decentralized exchanges (DEXs) and centralized exchanges (CEXs). While the trades on DEX are utilized by blockchain-native transactions that are verified by the nodes, the trades done with the use of CEXs do not rely on such blockchain-native transactions. CEX is a type of crypto asset exchange operated by a centralized entity, that provides a marketplace that supports trades of assets between users. The trades on CEX are recorded only on the ledgers that are kept by the operating party and not recorded on the blockchain. Thus, the trading process is faster and more flexible than DEX in that the users are also allowed to trade fiat money (e.g., U.S. Dollar) with crypto assets [27]. Due to such advantages in usability, CEX has become a dominant type of crypto asset exchange [4].

While the structure of CEX, where the trades are recorded on a ledger possessed only by a centralized party, has benefits for usability, it becomes a critical drawback when it comes to protecting users' own assets from the operating party. While stock exchanges like Nasdaq and NYSE (New York Stock Exchange) have a structure that is somewhat similar to CEX in that a centralized organization records and maintains ledgers, they are subject to government regulation and audits, which sets them apart from CEX in terms of reliability. Reliability issues have already surfaced in relation to improper CEX audits and regulations [11]. It is also highlighted in the case of the arrest incident that there always exists a risk of embezzlement.

In order to address the reliability issues, several research works [12–14] and services [5, 24] propose proof of reserves (PoR) protocols. The MProve protocol [12, 13] and Revelio protocol [14] focus on proving reserves of a particular type of crypto asset that supports anonymous transactions, Monero and MimbleWimble, respectively. Although the protocols mentioned above improve the financial soundness of crypto

도면6

asset exchanges, they are not suitable for proving the entire reserves of CEXs. On the other hand, protocols proposed by Kraken [24] and Binance [5] provide audits of multiple crypto assets that are traded within their exchange platforms by utilizing a Merkle tree [26]. The protocols allow auditors to generate snapshots for each user's balance, form a Merkle tree with the snapshots, and provide the tree information to users to validate whether their snapshots are included in the tree or not. The users are given their snapshot values by the exchanges.

While the structures of such third-party auditing with a Merkle tree seem plausible, they also possess a potential risk of manipulation regarding the question of trust in the exchanges and their auditors. For example, auditors may cherry-pick snapshots of user balance with different timestamps, resulting in a manipulation of the total reserves of the exchange. This scenario is described in detail in Section 2.

In this paper, we propose a structure for a bidirectional financial soundness audit (Bi-FiSA), where a PoR protocol for exchange platforms is carried out bidirectionally. With Bi-FiSA, as the audit process is carried out bidirectionally, account holders can verify their own balance in their own hands. The audit process is done in a decentralized manner by utilizing asynchronous distributed key generation (ADKG) and the blockchain publications of processes. Bi-FiSA is applicable for audits on various types of crypto assets, while not revealing each user's balance. Bi-FiSA utilizes the concept of a 'pause period' to prevent cherry-picking snapshots, and it is demonstrated that it can be minimized in order to avoid negative impacts on the trading ecosystem of the exchange platform.

The rest of this paper is organized as follows. In Section 2, we explain related works with descriptions of their mechanisms and risks. Section 3 explains the theoretical backgrounds of the proposed method. In Section 4, we introduce the key ideas of Bi-FiSA informally, and Section 5 explains the structure formally. Section 6 describes experiments regarding lengths of the pause period and a round of Bi-FiSA, and analyzes the results. Section 7 analyzes the privacy of Bi-FiSA. We conclude this paper in Section 8 with potential future research agenda.

2 Background

2.1 Related Work

While PoR is a relatively new concept in blockchain, audits of real-world asset reserves have a long history. For example, the major stock exchange NYSE has a robust system for order processing, trade reporting, quote reporting, and audit trails, with a strong focus on recording and auditing [20]. These exchanges, regulated by the SEC (U.S. Securities and Exchange Commission), have maintained audit trail accuracy at 98 percent. Even for challenging-to-value assets like oil and

gas, government and regulatory bodies have used probabilistic approaches for reserve evaluation [7]. Overall, institutions holding real-world asset reserves are subject to regulatory audits, a practice that blockchain PoR methodologies draw inspiration from.

In the blockchain world, trades within CEXs are blind spots in which the currency flows are not explicitly recorded on the blockchain. A user who wants to trade an asset on a CEX must deposit the asset into an account owned by the exchange, and any subsequent trades are only recorded on the exchange's ledger. Only when the user withdraws the asset is the point where the transaction is recorded on the blockchain. Thus, if exchanges are not obligated to report the financial soundness to users under appropriate audits, it is not difficult for the exchanges to utilize users' assets other than for contracted purposes.

In order to prove their financial soundness, Kraken [24] and Binance [5] introduced PoR methods that implement third-party audits with the use of a Merkle tree. Kraken and Binance perform PoR audits using Merkle trees and third-party auditors. These auditors verify asset existence, anonymize customer balances into snapshots, and provide publicly verifiable addresses. Clients can check if their balances exceed audit reserve amounts. Binance's Merkle tree, generated by a third party, allows users to verify asset holdings and inclusion in liability reports using the Merkle root hash.

Kraken and Binance utilize trusted third-party auditors to create a Merkle tree for verifying user funds. However, there are concerns that this method of fund verification through trusted third-party auditors may run counter to the decentralization philosophy of blockchain. It should also be noted that there is a possibility of malicious manipulation of the fund verification process between third-party auditors and exchanges, as well as the provision of inaccurate ledgers or misinformation in situations where verification is not carried out adequately. Such factors can impact the trustworthiness of the exchanges, amplifying concerns over users' investments and asset protection. Consequently, research on a financial soundness audit method for CEXs is needed, which aligns with the principles of decentralization and blockchain, while also ensuring transparency and privacy.

2.2 Snapshot Cherry-Picking

This section provides a scenario where a malicious CEX cherry-picks snapshot values in order to achieve the desired proof of total reserves.

The existing works of Kraken and Binance audits on the reserves at a specific time. In those works, the third-party auditor creates a snapshot hash for each user that includes information about the user's account and its balance. However, if the third-party auditor accidentally or intentionally picks certain users' snapshots with different timestamps, it can result in a huge difference in the result of the calculated

도면7

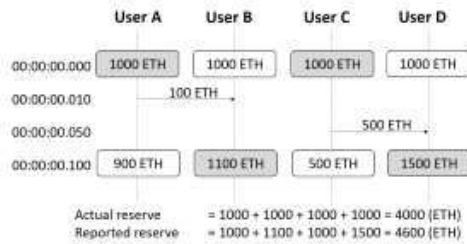


Figure 1: Cherry-picking snapshots to inflate the reserve.

total reserves.

Furthermore, existing works do not dictate the precise time of taking a snapshot. Even if it is specified, the resolution is limited because the discrepancy between the server time and local time of user devices is inevitable.

Let us assume there is a CEX with the existing PoR protocol, and the time for the audit is 00:00:00 on the first day of every month. Note that the time cannot be as precise as like 00:00:00.000000000 (in the order of nanosecond) because the server time and user device time cannot be synchronized in such a fine-grained manner. In the following scenario, as also illustrated in Figure 1, inflation of the exchange's total reserves is possible.

Users A and B trade assets *ten millisecond after* the audit time, where A sells 100 ETH (Ethereum) and B buys that 100 ETH. Similarly, let us suppose user C sells 500 ETH to user D at fifty millisecond after the audit time. The malicious third-party auditor may generate snapshots of users with different timestamps. In this example, the snapshot of users A and C is taken before their trade is made, while that of users B and D is taken after their trade. None of them can notice the discrepancy because (1) the difference in the timestamp is within the noise margin (shorter than a second), and (2) they cannot check out the balance of other users. Therefore, the total reserve reported in this example is 4,600 ETH, while the actual reserve is only 4,000 ETH.

Through the scenario above, the value of the exchange's total reserves is manipulated. If this is the case, the audit results can also be manipulated. Even if the exchange is 600 ETH short on the asset for whatever reason (accidental or intentional), the audit result will show users that the exchange has sufficient reserves. More importantly, the users have no choice but to accept the audit result because the audit process appears to be performed in accordance with the rule.

The scenario is entirely feasible, given that the trade frequency often surges in a very short period of time [2]. This scenario of manipulation, the timestamp cherry-picking, is achievable because the user does not actively participate in the audit process. If users participate in audits by ensuring their balance values the timestamp points are correctly reflected in

the audit, this kind of manipulation can be prevented.

3 Preliminaries

3.1 Asynchronous Distributed Key Generation

Distributed key generation (DKG) [16] emerges as a fundamental technique that facilitates the initiation of threshold cryptosystems, all the while circumventing the need for a centralized, trust-requiring entity. Its importance extends to decentralized protocols' foundations, including threshold signatures and multiparty computations.

These cryptographic paradigms place a significant emphasis on secret-shared private keys, which are then used in subsequent threshold cryptosystems to perform tasks like creating threshold signatures [16] or decrypting ciphertexts when using threshold encryption techniques. The assurance of security and integrity across these systems hinges on the robustness of the initial DKG process.

A practical embodiment of asynchronous distributed key generation (ADKG) is heralded by the work of Das et al. [10], a protocol grounded in discrete logarithm-based cryptosystems, exemplified by cryptographic schemes like ElGamal encryption. Notably resilient, this protocol endures the presence of up to $m < n/3$ malicious nodes, where m the threshold count of malicious nodes and n the number of nodes, safeguarding the integrity of the generated shared randomness and the computation of individual shares comprising the final key. This resilience underscores its suitability for contexts requiring stringent security assurances.

By adopting the ADKG protocol, the Bi-FiSA framework harnesses the capabilities of distributed key generation to empower multiple nodes within the network to collectively construct an encryption key. This key, which is used for the secure encryption of sensitive information provided by users, serves as the foundation for the decentralized computation of homomorphic additions. Importantly, this process does not rely on a trusted third party, which strengthens the framework's trustless architecture. Furthermore, the ADKG implementation within Bi-FiSA enforces a strict sequence of operations. Specifically, nodes are mandated to delay the decryption of users' confidential information until after the homomorphic addition computation has been executed, ensuring data privacy. This mechanism not only protects users' privacy but also the integrity of the overall computation, preventing sensitive information from being exposed prematurely in the absence of collusion exceeding the established threshold.

3.2 Pause of Trading

Crypto asset exchanges play a pivotal role in the digital economy by facilitating the exchange of various coins and tokens. As these exchanges grow in complexity and volume, ensuring the integrity of their operations becomes paramount. One

도면8

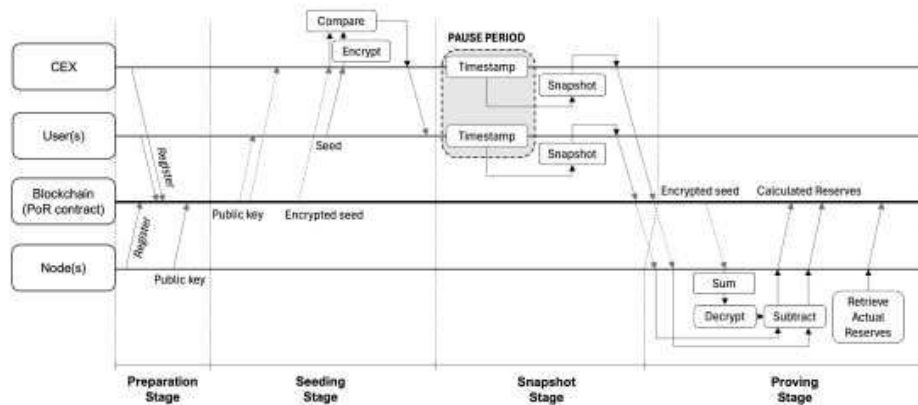


Figure 2: A diagram demonstrating an overview of a round of Bi-FiSA. Blue arrows with dotted lines represent retrievals of data from the PoR contract deployed on the blockchain.

crucial aspect often overlooked in the realm of crypto assets is the absence of a designated 'pause period'—a deliberate timeframe during which all trading activities are temporarily halted. This paper advocates for the implementation of such a pause period as an essential component of a new reserves auditing protocol for CEXs.

We draw inspiration from established financial markets, where trading halts are used for market stability and investor protection. For example, the NYSE implements halts (i.e., circuit breakers) during significant market fluctuations or concerns to prevent panic and ensure information dissemination [23]. Traditional stock exchanges also have fixed trading hours with scheduled pauses for reserve inspection and balance verification, promoting stability. Similarly, the banking sector allocates maintenance hours [22] and cut-off times [25] for system inspection and audits, emphasizing the importance of brief interruptions to maintain accuracy and transparency in financial activities.

In contrast, the crypto asset industry lacks a pause period structure. Currently, CEXs only pause for inspections in response to severe network issues or major irregularities, missing proactive reserve assessment. The proposed pause period aligns with financial practices, enhancing accountability, transparency, and resilience in digital finance. Trading suspensions at designated intervals enable comprehensive reserve audits, risk mitigation, and boost investor confidence.

Auditing protocols aim for accurate reserve assessments. Notably, exchanges like Binance and Kraken conduct audits without a comprehensive pause period, ensuring continuous trading. However, trust issues arise when exchanges audit unilaterally. To overcome this, a pause period becomes essential. Bi-FiSA fosters trust with a bidirectional audit involving

users and the exchange. It addresses concerns such as timestamp manipulation, allowing users an active role in verifying platform integrity.

While the pause period offers a potent solution to the synchronization challenges posed by snapshot disparities, it is imperative to acknowledge the potential implications of its implementation. CEXs often witness substantial trading volumes that can surge dramatically during periods of heightened market activity [2]. The introduction of a pause period has the potential to disrupt the dynamic trading ecosystem and raise concerns regarding market liquidity and accessibility, as seen in the case of stock exchanges [23]. Given that it has been found that the length of the trading pause and the market volatility are somewhat in a relationship of increasing tendency [21], minimizing the pause would lessen the disruption of the trading ecosystem.

Recognizing the delicate balance between maintaining the accuracy and security of audits and safeguarding the uninterrupted flow of trading activities, this paper dedicates Section 6 to a comprehensive analysis of the possible pitfalls associated with implementing the pause period. Through a meticulous analysis of potential drawbacks, we endeavor to ascertain whether the benefits of bidirectional auditing and snapshot synchronization outweigh the potential disruptions.

4 Overview

In this section, we explain key ideas of Bi-FiSA informally. Formal definition of Bi-FiSA is given in the following section.

When a single entity has full control over CEX audits, it can compromise the original intent, as seen in the CEX CEO

도면9

arrest and as demonstrated for existing works in Section 2. Even though a third party is chosen as a verifier, if the verifier is a single entity, it does not resolve the trust issue. To address this problem, we propose a model involving a *blockchain* network of verifiers, where multiple nodes participate. The blockchain is required to ensure that audits of individual and total reserves use identical values. The specifics of the audits are introduced in Section 5.2. The blockchain also allows for accurate data backtracking when users and the exchange disagree about the correctness of data.

A single round of the audit is composed of four sequential stages, as depicted in Figure 2. A round starts with the preparation stage where the nodes set up a public key. During the preparation stage, eligible nodes and users are registered. The nodes registered at Bi-FiSA derive synchronized public keys using the ADKG protocol. The ADKG protocol prevents a node from decrypting encrypted data outside of the established protocol, as the ADKG protocol tolerates malicious nodes up to one-third of the total nodes.

The next is the seeding stage, the stage where encrypted seed values are posted by each user and the exchange confirms the validity of the seeds. In order to improve user privacy, each user needs to generate seed values that are added to balance values to conceal the balance from the public. A user needs to generate a unique pseudorandom seed value for each asset. This method of generating seeds is to obfuscate the pattern of the sum of the user's balance and a seed. For instance, when a user owns assets A and B on the exchange, the balance of each asset may be significantly different from the other two due to the potential for a large price difference in fiat currency between them. The user balance difference between A and B may be exponential as a result of the price discrepancy. For example, the balance of A is 0.01 and the balance of B is 100 because A is much more expensive than B. Therefore, if the same seed value is added to the balance, there is a risk of a third party noticing the user's actual balance, or at least its approximate value. Users encrypt these seeds and publish them to the PoR contract. Then, users send the plain seed values to the exchange through a separate secure channel. The exchange encrypts the plain values and compares them to the corresponding encrypted values posted on the contract. Given that some users may fail on correct computations, the exchange grants users a certain number of such comparison attempts. Users who have failed the maximum number of attempts are classified as non-participating. During the snapshot stage, the CEX posts snapshots on behalf of non-participating users. Lastly, the exchange posts the result of the comparison to the contract.

The third is the snapshot stage in which the pause period is carried out to provide a short period of time for the users and the exchange to take snapshots. The pause period synchronizes time points among users and the exchange via a specialized three-way handshake, by each entity recording timestamps upon receiving the other entity's signal. The timestamp

values for a user's device and the exchange server recorded may differ due to the time difference between devices and network delays. However, the balance values displayed on the server and the user's device that the timestamp points are assured to be identical. This is because all the trades are paused at the moment when the timestamp is recorded. After this, balance values for each asset are captured. Users calculate and publish their snapshots, combining asset balances with pseudorandom seed values to the PoR contract. Simultaneously, the exchange computes and publishes sums for each user in the same way, completing the snapshot phase.

After the snapshot stage, the nodes perform the audits in order to prove financial soundness during the proving stage. In the proving stage, nodes perform individual audits, which are also verifiable by anyone via the PoR contract. They also calculate total reserves and publish the calculation with the exchange-owned on-chain balances to the contract concluding a round of financial soundness auditing.

5 Bi-FiSA

5.1 Balance and Seed Values

For a fine description of processes that handle users' balance values and seed values, we define an asset balance value and a random number seed value in a vector form as in Equation 1 and 2, in order to assign different values for each type of asset. Here, k is the type of an asset on the exchange, u^i is the i^{th} user, and $u^i \in U$ where U is the set of registered users. Also, b_k^i and s_k^i are a balance value and a seed value of the i^{th} user for the k^{th} asset type, respectively, where $k \leq K$, the number of asset types available on the exchange. Note that a balance value b_k^i for any user and any type of asset (i.e., $\forall k$ and $\forall i$) is a number value for easing calculations, represented in a multiple of the smallest unit the exchange can express the asset (e.g., 1 satoshi, the smallest of Bitcoin, is equal to the 10^{-8} Bitcoin²). $\vec{S}^i$ is used to denote the snapshot value for the i^{th} user, which is equal to $\vec{b}^i + \vec{s}^i$, as described in Equation 3.

$$\vec{b}^i := (b_1^i, b_2^i, \dots, b_K^i) \quad (1)$$

$$\vec{s}^i := (s_1^i, s_2^i, \dots, s_K^i) \quad (2)$$

$$\vec{S}^i := \vec{b}^i + \vec{s}^i \quad (3)$$

In addition, we define the actual total reserves (owned by the CEX) as $\vec{B}$, the vector of actual reserves value for each type of asset, B_k for the k^{th} asset, as described in Equation 4. The actual reserves for each type of asset can be retrieved from its corresponding blockchain network, by browsing the balance values of the addresses the exchange owns.

$$\vec{B} := (B_1, B_2, \dots, B_K) \quad (4)$$

²[https://en.bitcoin.it/wiki/Satoshi_\(unit\)](https://en.bitcoin.it/wiki/Satoshi_(unit)), last accessed August 15, 2023

도면10

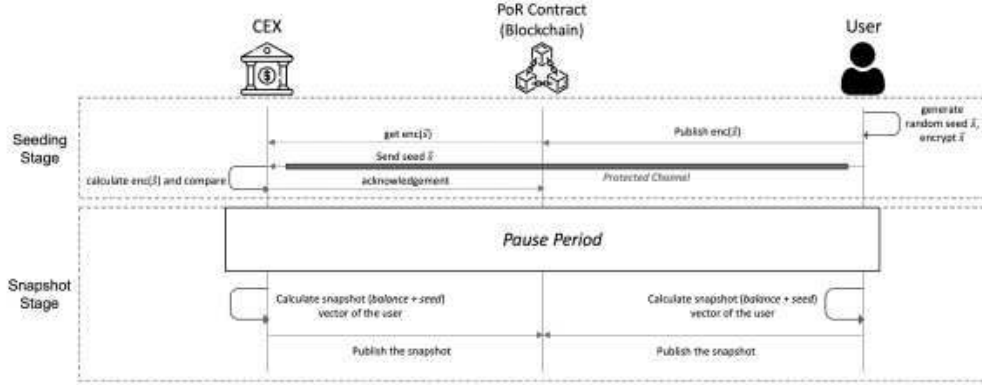


Figure 3: A diagram demonstrating the operation of a seeding stage and a snapshot stage

5.2 Design Goals

Throughout the following descriptions, notations are used as follows:

- tsp^i : A timestamp value the i^{th} user records from the user's device, which points a specific time within the pause period.
- tsp^{exch} : A timestamp value the exchange records from the server, which points the time the pause period started.
- $b_k|_{tsp^i}$: Asset k balance of the i^{th} user, pointed to by timestamp tsp^i on the user's device.
- $b_k|_{tsp^{exch}}$: Asset k balance of the i^{th} user, pointed to by timestamp tsp^{exch} on the exchange's server.

The purpose of Bi-FiSA is to verify that a CEX is holding the assets that users perceive themselves to possess. This can be divided into two distinct objectives: auditing individual reserves and total reserves. In order to audit individual reserves, $b_k|_{tsp^i}$ should be equal to $b_k|_{tsp^{exch}}$ for each type of asset. In terms of total reserves, the actual total reserves $\vec{B}$ should be greater than or equal to the sum of $b_k|_{tsp^i}$ for all users, for each type of asset. Audits on individual and total reserves can be expressed in Equation 5 and 6, respectively.

$$b_k|_{tsp^i} \stackrel{?}{=} b_k|_{tsp^{exch}} \quad \forall k \in [1, K] \quad (5)$$

$$\sum_i b_k|_{tsp^i} \stackrel{?}{\leq} B_k \quad \forall k \in [1, K] \quad (6)$$

Given that the previous works introduced in Section 2 lack the bidirectional structure (i.e., do not let users actively take part in the audit by computing $b_k|_{tsp^i}$), they cannot guarantee meeting Equations 5 and 6.

While satisfying the objectives, Bi-FiSA should also preserve users' privacy. In order to do so, $\vec{b}^i$ for the i^{th} user should

not be revealed to the public. In other words, an adversary should not be able to computationally identify $\vec{b}^i$ from the snapshot of the i^{th} user. Bi-FiSA meets this condition if seed values are pseudorandomly generated in the seeding stage. The validity of this privacy goal is described in Section 7.

To ensure Bi-FiSA operates smoothly, user and node participation is crucial. We assume users and nodes are adequately compensated, with rewards surpassing participation costs.

5.3 Preparation Stage

During the preparation stage, eligible nodes and users are registered. New nodes are registered with a vote by existing nodes, while new users are registered by the exchange.

Throughout the following descriptions, notations are used as follows:

- adr^{send} : An address of the sender, the one who calls the function
- adr^{self} : An address of self, the executor of the function
- adr^{exch} : An address of the exchange
- adr^i : An address of the i^{th} user
- thr : A threshold number of honest nodes, defined by ADKG
- pk : A public key a node computed through ADKG
- PK : Established public key that is to be used through a round

Each registered node at Bi-FiSA derives a key through $KEYGEN(adr^{self})$. Its pseudocode is shown in Algorithm 1. It first executes $ADKG_{gen}$, which generates a key pair through ADKG protocol. It outputs a public key pk and a node's private key (in ADKG, each node produces a different private key). The function $KEYGEN$ then generates a transaction $TX_{PostKey}$ with (adr^{self}, pk) sent to the PoR contract to call

도면11

POSTKEY function.

Node

o KEYGEN(adr^{self}):

- 1: Generate pk through ADKG_{gen}
- 2: $TX_{PostKey} \leftarrow (adr^{self}, pk)$
- 3: **return** $TX_{PostKey}$

PoR Smart Contract

o POSTKEY(adr^{send}, pk):

- 4: **Assert** adr^{send} in the list of registered nodes.
- 5: **Assert** The key is not yet established.
- 6: **if** Count(pk) = 0 **then**
- 7: Append pk to the list of candidates for PK
- 8: Count(pk) $\leftarrow$ Count(pk) + 1
- 9: Append adr^{send} to the list of nodes who posted a key.

o SETKEY:

- 10: **Assert** The length of two lists, the list of registered nodes and the list of nodes who posted a key, equals.
- 11: **Assert** The key is not yet established.
- 12: **for all** pk in the list of candidates for PK **do**
- 13: **if** Count(pk) $\geq$ thr **then**
- 14: PK $\leftarrow$ pk
- 15: **break**
- 16: **else** Roll back stored values before posting keys.

o RESETKEY(adr^{send}):

- 17: **Assert** $adr^{send} = adr^{self}$
- 18: **delete** PK

Algorithm 1: Scheme for the preparation stage.

Each node posts pk to the contract as a candidate for a round public key PK, by calling POSTKEY function at the PoR contract with $TX_{PostKey}$, where adr^{send} becomes the address of each node. The function first verifies if adr^{send} is the address of a registered node, and if the public key for the round is not yet established. If Count(pk) = 0, where Count(pk) is the integer value counter increasing from 0 that indicates the number of pk enrolled, pk is appended to the list of candidates for PK. Whether or not Count(pk) is 0, the counter Count(pk) is increased to indicate that the number of pk submitted as a candidate has increased. Also, adr^{send} is appended to the list of addresses of nodes that posted their pk, in order to denote that the sender, the node, posted a key.

The agreed public key, PK, is then established with SETKEY function at PoR contract, by verifying if any posted key is proposed by greater than or equal to thr number of nodes. If all the candidates do not meet the condition, it means that the nodes were unable to agree on an identical public key. In other words, the correctness of the computations made during the key derivation phase of the ADKG protocol is compromised. If this is the case, the contract rolls back contract states before the keys were posted by using delete function

that assigns variables initial values [15], such as assigning an integer value 0, and an array (i.e., a list) a dynamic array of length zero. SETKEY function can be called by any entity only if the numbers of the items included in the list of registered nodes and the list of nodes who posted a key are identical. It also should be ensured that the round key is not yet established.

5.4 Seeding Stage

User

o ENCSEED(PK, adr^{self}):

- 1: User u^i generates a vector of random seed values $\vec{s}^i = (s_1^i, s_2^i, \dots, s_k^i)$, where $s_k^i \in \mathbb{N}$
- 2: $\vec{E}^i \leftarrow (enc(s_1^i), enc(s_2^i), \dots, enc(s_k^i))$
- 3: $TX_{PostSeed} \leftarrow (adr^{self}, \vec{E}^i)$
- 4: **return** $\vec{s}^i, TX_{PostSeed}$

Exchange

o VERISEED(PK, adr^{self}):

- 5: Receive vectors of seeds ($\vec{s}^i$ for the i^{th} user) from each user through a protected channel
- 6: Let $List_{participate}$ the list of addresses of users with successful verification of seed values.
- 7: **for** j ≥ 0 lower than the maximum number of trials **do**
- 8: **upon** t_{seed}^j **ends do**
- 9: Get all available $\vec{E}^i$ from PoR contract.
- 10: **for** u^p that posted $\vec{E}^p$ on PoR contract for j^{th} seed posting period **do**
- 11: From $\vec{s}^p$ the user sent, compute $(enc(s_1^p), enc(s_2^p), \dots, enc(s_k^p))$ and check if it equals to $\vec{E}^p$.
- 12: **if** they are equal **then**
- 13: Append adr^p to $List_{participate}$.
- 14: **else**
- 15: Notify u^p to start seeding stage from the beginning.

16: $TX_{SeedAck} \leftarrow (adr^{self}, List_{participate})$

17: **return** $TX_{SeedAck}$

PoR Smart Contract

o POSTSEED($adr^{send}, \vec{E}^i$):

- 18: **Assert** adr^{send} in the list of registered users.
- 19: Parse $\vec{E}^i$ as a list of $enc(s_k^i)$ for all k, and store the list as the list of i^{th} user's encrypted seeds.

o SEEDACK($adr^{send}, List_{participate}$):

- 20: **Assert** $adr^{send} = adr^{exch}$
- 21: Store $List_{participate}$ as the list of participating users.

Algorithm 2: Scheme for the seeding stage.

도면12

The seeding stage, shown in Algorithm 2, begins after the end of the preparation stage. Throughout the following descriptions, notations are used as follows:

- $\text{enc}(s_k^i) : s_k^i$ encrypted with PK.
- $\vec{E}^i : A$ vector of $\text{enc}(s_k^i)$ for all k , which equals to $(\text{enc}(s_1^i), \text{enc}(s_2^i), \dots, \text{enc}(s_k^i))$.
- $t_{\text{seed}}^j : A$ time window for the j^{th} trial of a seeding stage.

As the seeding stage begins, each user generates seed values, s^i for the i^{th} user. After generating s^i , each user calculates $\vec{E}^i$ and publishes it to the contract by calling POSTSEED function with $\text{TX}_{\text{PostSeed}}$, a transaction with the user's address and encrypted seeds. As soon as the encrypted seeds are published, each user sends the seed values s^i to the exchange. Users are given a time window of t_{seed}^j to carry out the above processes. Because some users may fail on correct computations, they are given a certain number of trials of this time window.

After t_{seed}^j , the exchange then gets $\vec{E}^i$ for each i from PoR contract and compares it to the value drawn from the encryption of s^i each user sent. For each user, if the comparison outputs *true*, the exchange appends adr^i to $\text{List}_{\text{participate}}$, the list of addresses of users with successful verification. If the comparison outputs *false* for the maximum number of trials, the exchange considers the user faulty. After the maximum number of trials, the address of the exchange and $\text{List}_{\text{participate}}$ is parsed as $\text{TX}_{\text{SeedAck}}$. Then, it is published to PoR contract by calling SEEDACK, where the list of participated users (i.e., $\text{List}_{\text{participate}}$) is stored. SEEDACK can only be called by the exchange.

5.5 Pause Period

Although the pause period is part of the snapshot stage, it is described in advance in this subsection for clarity. Some notations for descriptions of the processes are used as follows:

- SYN : A packet of signal sent to a user to record a timestamp upon receiving.
- SYN-ACK : A packet of signal sent to the exchange as an acknowledgment that the sender (the user) recorded a timestamp.
- ACK : A packet of signal the exchange sends to the sender of SYN-ACK as an acknowledgment that the exchange well received the packet.

Following the seeding stage, the snapshot stage initiates with a pause period to synchronize snapshots between users and the exchange, addressing timing differences as mentioned in Section 4. This pause temporarily halts trades (sell, buy, withdraw, and deposit) to facilitate synchronization. During this period, traffic of signals is carried out between the exchange and users in the form of three-way handshakes. Unlike the sequence of traditional three-way handshake of transmission control protocol (TCP) [1] uses, the handshake for the pause period of Bi-FiSA is initiated by the server (i.e., the

exchange), sending an SYN packet as a timestamp recording signal. Before sending SYN, the exchange pauses trades and records tsp^{exch} . Upon receiving the packet, a user records a timestamp tsp^i (assuming the i^{th} user), then sends a packet with SYN-ACK as an acknowledgment that the user has successfully received the signal. As soon as receiving the user's packet, the server sends ACK to the user to notify that the server has successfully received the packet. The pause period ends after the exchange receives all the users' packets (SYN-ACK), and trades on the exchange become available.

This mechanism ensures that the balance in the exchange never changes from the start (tsp^{exch}) to the end of the pause period, and tsp^i is always in-between them. Thus, the snapshot taken by each user at his/her local time tsp^i must be the same with the snapshot taken by the exchange at tsp^{exch} .

By implementing the mentioned structure, we propose two different handshake models, exchange-driven handshakes (EDH) and user-driven handshakes (UDH).

5.5.1 Exchange-Driven Handshakes

The model of EDH is comprised of direct networking between the exchange and users. The exchange sends SYN as a timestamp signal to all the users consequently. Upon receiving SYN, each user sends back SYN-ACK to the exchange. The exchange replies to each user with ACK, but the pause period is needed for only the duration of the following processes take: (1) execution time of which the exchange sending SYN to users, and (2) standby time of which the exchange has to wait until users sending back SYN-ACK are done.

The execution time is where the time takes for the exchange to send multiple SYN packets, thus equals to the number of users times the size of a packet divided by the network throughput. The standby time, on the other hand, is equal to the single user delay, which is the length of time it takes for a server to receive a packet from a user. This is because the server can send multiple SYN packets and receive user packets concurrently, as depicted in Figure 4 (a).

5.5.2 User-Driven Handshakes

The UDH protocol differs from the EDH method in message distribution. In UDH, the server sends messages to neighbor users, who then relay them to their neighbors, resembling gossip protocols [3]. This decentralized approach outperforms a single entity broadcasting all messages.

The UDH protocol operates through a series of iterative steps that serve to synchronize the actions of participating users and the exchange server. First, the synchronization process commences with the server's initiation. The server sends a SYN signal to neighboring users. This SYN acts as a trigger, signaling the start of the synchronization procedure. Upon receiving the server's SYN signal, each user records a timestamp tsp^i and replies to the server with SYN-ACK. This step

도면13

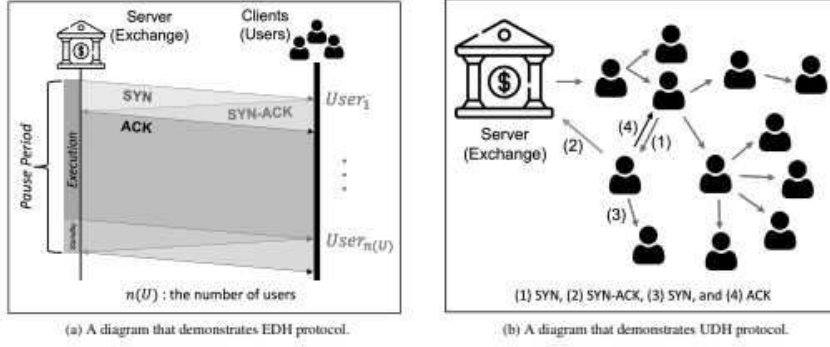


Figure 4: Overview of two different protocols for a pause period in Bi-FiSA.

represents their commitment to participate actively in the coordination. With the reception of SYN-ACK, the users take on the role of propagating the synchronization signal. They forward the SYN signal to other neighboring users, effectively extending the reach of the synchronization process throughout the network. This step ensures that all relevant users are brought into the synchronization loop.

The process described above—initiation by the server, user responses, and user-to-user propagation—is executed iteratively to ensure comprehensive synchronization. This cyclic repetition guarantees that all users within the network acknowledge and participate in the synchronization process, mitigating any potential discrepancies in timing. Additionally, a noteworthy aspect of the UDH protocol is the incorporation of ACK signals to validate the successful execution of the synchronization process. After the third step—where users receive SYN signals from other users, not the server—they send ACK signals back to the parent user who initiated SYN. These ACK signals serve as confirmations that the synchronization process has been effectively carried out, further bolstering the integrity of the coordination effort. The structure of the protocol is depicted in Figure 4 (b).

5.6 Snapshot Stage

Throughout the snapshot stage in Algorithm 3, $TX_{PostUserSnap}$ and $TX_{PostExchSnap}$ are used to describe transactions that call $POSTUSERSNAP$ and $POSTEXCHSNAP$ function for the i^{th} user's snapshots, respectively, of which the former is executed by a user and the latter by the exchange. Also, following notations are used for descriptions:

- $\vec{S}_{user}^i$: the vector of snapshot values the i^{th} user posted.
- $\vec{S}_{exch}^i$: the vector of snapshots the exchange has posted for the i^{th} user.

User

◦ $USERSNAP(adr^{self}, tsp^i, \vec{s}^i)$:

- 1: **for each** k **do**
- 2: $b_k^i \leftarrow$ (the balance value of k the tsp^i points)
- 3: $\vec{S}^i \leftarrow \vec{s}^i + (b_1^i, b_2^i, \dots, b_k^i)$
- 4: **return** $TX_{PostUserSnap} \leftarrow (adr^{self}, \vec{S}^i)$

Exchange

◦ $EXCHSNAP(adr^{self}, tsp^i, List_{Send}^{Exch}, List_{UserAddr})$:

- 5: **for each** adr^i in $List_{participate}$ **do**
- 6: **for each** k **do**
- 7: $b_k^i \leftarrow$ (the balance value of k the tsp^{exch} points for the i^{th} user)
- 8: $\vec{S}^i \leftarrow \vec{s}^i + (b_1^i, b_2^i, \dots, b_k^i)$
- 9: $TX_{PostExchSnap} \leftarrow (adr^{self}, \vec{S}^i)$
- 10: **return** $\{TX_{PostExchSnap}^i\}$ for each i

PoR Smart Contract

◦ $POSTUSERSNAP(adr^{send}, \vec{S}^i)$:

- 11: **Assert** $adr^{send} \in List_{participate}$
- 12: $\vec{S}_{user}^i \leftarrow \vec{S}^i$
- 13: Append $(adr^{send}, \vec{S}_{user}^i)$ to the list of user snapshots.

◦ $POSTEXCHSNAP(adr^{send}, (adr^i, \vec{S}^i))$:

- 14: **Assert** $adr^{send} = adr^{exch}$
- 15: $\vec{S}_{exch}^i \leftarrow \vec{S}^i$
- 16: Append $(adr^i, \vec{S}_{exch}^i)$ to the list of exchange snapshots.

Algorithm 3: Scheme for the snapshot stage.

Each user captures balance values after the pause period ends. A vector of numerical balance values for each type of crypto asset, b^i for the i^{th} user, is captured for the time where

도면14

tsp^i points.

After $\vec{b}^i$ is captured, each user calculates a vector of snapshots $\vec{S}^i$, the sum of $\vec{b}^i$ and $\vec{s}^i$. Then, each user publishes $\vec{S}^i$ to PoR contract by calling `POSTUSERSNAP`, where a tuple of $(adr^i, \vec{S}^i)$ is appended to the list of user snapshots. The exchange also computes $\vec{S}^i$ for each user, and publishes each to the contract where a tuple of $(adr^i, \vec{S}^i)$ is appended to the list of exchange snapshots.

5.7 Proving Stage

PoR Smart Contract

```

o INDIVAUDIT( $adr^i$ ):
1: if  $\vec{S}_{user}^i = \vec{S}_{exch}^i$  then
2:   return true
3: else
4:   return false

o CALCRESERVEUSER( $adr^{send}, \sum \vec{b}^i$ ):
5: Assert  $adr^{send}$  in the registered nodes.
6: Store  $\sum \vec{b}^i$  as user-snapshot-calculated reserves.

o CALCRESERVESEXCH( $adr^{send}, \sum \vec{b}^i$ ):
7: Assert  $adr^{send}$  in the registered nodes.
8: Store  $\sum \vec{b}^i$  as exchange-snapshot-calculated reserves.

o ACTUALRESERVES( $adr^{send}, \vec{B}$ ):
9: Assert  $adr^{send}$  in the registered nodes.
10: Store  $\vec{B}$  as the actual reserves the exchange owns.

```

Algorithm 4: PoR smart contract for the proving stage.

At the proving stage shown in Algorithm 4 and Figure 5, audits on the individual reserves are first carried out by nodes. Not only the nodes but anyone can check its validity through `INDIVAUDIT` function on PoR contract, which simply compares $\vec{S}_{user}^i$ and $\vec{S}_{exch}^i$. Here, if the comparison outputs *true*, it shows that the balance value that appears to the i^{th} user and the value the CEX claims to hold for the user is identical. Thus, the objective of Bi-FiSA to audit individual reserves is fulfilled by calling `INDIVAUDIT` function, as described in Equation 7.

$$\vec{S}_{user}^i \stackrel{?}{=} \vec{S}_{exch}^i \iff \vec{b}_k^i|_{tsp^i} \stackrel{?}{=} \vec{b}_k^i|_{tsp^{exch}} \quad \forall k \in [1, K] \quad (7)$$

After the individual audit, each node gets snapshots from PoR contract. Upon getting snapshots, each node calculates $\sum \vec{S}^i$, the sum of snapshots the users posted. The calculation is described in Equation 8. Then each node downloads all the encrypted seed values (for all users) from PoR contract, and

calculates their sum in a homomorphic way. Subsequently, the reconstruction phase of the ADKG protocol is carried out among the nodes to decrypt $\sum \vec{E}^i$. This decryption process prevents nodes from decrypting selected users' snapshots.

$$\sum \vec{S}^i = \sum (\vec{b}^i + \vec{s}^i) \quad (8)$$

The calculation of $\sum \vec{b}^i$, the calculated total reserves, is described in Equation 9. Here, `ADKGDecrypt` is a decryption algorithm through the ADKG protocol, and it decrypts the homomorphic sum of encrypted seed values. After the calculation of $\sum \vec{b}^i$ through the snapshots the users posted, each node publishes its outcome (i.e., $\sum \vec{b}_k^i|_{tsp^i}$) by calling `CALCRESERVESUSER` function at PoR contract. The nodes perform the identical process with the snapshots the exchange posted and publish the outcome (i.e., $\sum \vec{b}_k^i|_{tsp^{exch}}$) by calling `CALCRESERVESEXCH`.

$$\begin{aligned} \sum \vec{S}^i - \text{ADKG}_{\text{Decrypt}}(\sum \vec{E}^i) \\ = \sum \vec{b}^i \end{aligned} \quad (9)$$

Next, audits on the total reserves should also be made. It is fulfilled by comparing the calculated reserves with the actual reserves the exchange owns. In order to do so, nodes calculate $\vec{B}$, the actual reserves. By nodes publishing $\vec{B}$ by calling `ACTUALRESERVES` function at PoR contract, users become available to notice if there exists a gap between the actual on-chain reserves and the calculated reserves that are recorded on the ledger kept only by the exchange. Validation of Equation 6 is fulfilled by comparing the values stored by `CALCRESERVESUSER` and `ACTUALRESERVES`. Through these processes, a single round of a financial soundness audit ends.

6 Experiments and Analysis

In order to ensure Bi-FiSA's efficacy, we analyze the proposed model with two pivotal considerations: **pause period length** and **round length**.

Trading pauses, as discussed in Section 3, can disrupt high-frequency crypto asset markets, causing significant price fluctuations and imbalances. Evaluating whether Bi-FiSA's pause period can be minimized to reduce market impacts is crucial. Thus, we experiment and analyze the length of a pause period with EDH and UDH model, to maintain auditing precision while mitigating disruptions. The estimated pause period lengths for EDH and UDH are 15.36 *seconds* and 33.12 *milliseconds*, respectively, based on the experiment results.

Evaluating the round length is also important because the round length equals the length of an audit. If an audit is needed frequently, the round length should be minimized.

도면15

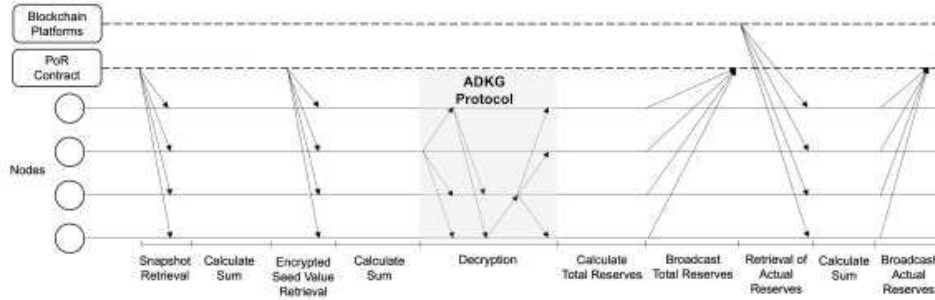


Figure 5: A diagram demonstrating the operation of the proving stage.

6.1 Pause Period Length

The calculation of the length of a pause period is carried out by experimenting on EDH and UDH models. Experiments were conducted with OMNeT++ discrete event simulator³, under the following reference values: 300 million users (which is more than three times as many as Binance, the world's largest CEX has [8]), a packet with the size 512 bits, and 10Gbps of the server's network throughput.

6.1.1 Exchange-Driven Handshakes

The relationship between pause period duration and its components for EDH is represented in Equation 10.

$$\begin{aligned}
 & \text{(EDH Pause Period Duration)} \\
 &= (\text{Execution Time}) + (\text{Standby Time}) \\
 &= \frac{(\text{Number of Users}) \times (\text{Size of A Packet})}{(\text{Network Throughput})} \quad (10) \\
 &+ (\text{Single User Delay})
 \end{aligned}$$

By calculating Equation 10 with the reference values, the execution time takes *15.36 seconds*. For the standby time, it is equal to the single user delay. The network delay between the exchange sending the packet to a single user and the user responding to the exchange is referred to as the single user delay. As a result, it is approximately twice the network delay for a packet. For the value of network delay, it has been researched that the end-to-end network delay can be calculated as the sum of transmission delay, processing delay, propagation delay, and queuing delay [6]. Also, since it has been found that queuing delays are usually on the order of microseconds to milliseconds, and propagation delays are usually on the order of milliseconds, their sum can be considered as *on the order of milliseconds*.

³<https://omnetpp.org/>, last accessed August 1, 2023

Both simulations for the calculation of processing delay and transmission delay took a second-long experiment (transmission of 142,642 packets). The packet length of 1,200 bytes and packet generation interval of 10 microseconds were taken for the experiment of the processing delay, and the packet length of 200 to 800 bytes with truncated normal distribution and 100 microseconds of packet generation interval were taken for the experiment of the transmission delay. The packet generation intervals for both experiments followed an exponential distribution. The mean values for the results showed that the processing delay and the transmission delay can be approximated as *1.244 milliseconds* and *0.068 milliseconds*. By summing up the delays, the end-to-end network delay is *3.312 milliseconds* if we consider the maximum value (1 millisecond) for the propagation and queuing delay. Thus, the standby time would be negligible compared to the execution time.

In summary, standby time is minimal compared to execution time. Execution time depends on the number of users and inversely on network throughput, as shown in Equation 10. Given these constraints, the EDH model's pause duration is significantly shorter than traditional stock exchange trading halts [21], even with values exceeding *300 million* users and *10 Gbps* network throughput. Thus, it would be safe to say that the pause period with the EDH model will not harm the trading ecosystem on a considerable scale, given that the experiment results with the reference values results *on the order of 10¹ seconds*.

6.1.2 User-Driven Handshakes

Experiments on the single user delay for UDH are identical to that of EDH, given that the experimental environments and variables are identical. Thus, the single user delay for UDH is *3.312 milliseconds*. Same with EDH, a server can send SYNs and receive user packets concurrently. However, unlike EDH, execution time for UDH depends on the single user delay

도면16

and the depth (the number of propagation rounds) for full propagation. Assuming a node propagates SYN to ten child nodes and 300 million users, the depth equals 9, because 300 million is less than 10^9 and greater than 10^8 . As the propagation for a single depth consumes a single user delay, a single depth takes 3.312 milliseconds. Thus, the execution time for the UDH with the reference values equals 9×3.312 milliseconds. Here, the time takes for a node to send a packet to child nodes is negligible because it equals 5.12×10^{-4} milliseconds when the reference values are used (10 users for a node, a 512-bit packet, and 10Gbps of network throughput). Consequently, utilizing the results of the experiment, the pause period length for UDH is approximately 33.12 milliseconds because a standby time equals a single user delay.

Overall, while the standby time of the EDH and UDH protocol is the same, the execution time takes much less for UDH. In conclusion, EDH is suitable enough for the implementation of a pause period, given that length would be on the order of 10^1 seconds. However, considering the difference in calculated results, implementing UDH would be more efficient for having a pause period in CEXs.

6.2 Round Length

It has been shown in the paper [10] that the introduced ADKG protocol is practical enough, and generating and encrypting random seed values are done concurrently by the users, stages without blockchain publications are not burdensome. The pause period during a snapshot stage is also shown to take little enough through Section 6.1.

However, blockchain network congestion is anticipated for the parts where blockchain publications are needed: (1) user publishing $\vec{E}^i$, (2) user publishing $\vec{S}_{user}^i$, and (3) exchange publishing $\vec{S}_{exch}^i$.

Let the length of an element that comprises two vectors, $\vec{E}^i$ and $\vec{S}^i$, is 256 bits, and PoR contract stores each vector as an array with 256-bit unsigned integers. Assuming 30 asset types to be audited ($K = 30$), publications of $\vec{E}^i$, $\vec{S}_{user}^i$, and $\vec{S}_{exch}^i$ are experimented to consume 0.7×10^6 . Thus, with 300 million users, 6.3×10^{14} gas is consumed for three blockchain publications where blockchain network congestion is anticipated. In other words, for the publication of 300 million user data, approximately 86.4 GB of data is stored.

The block capacity of Ethereum and Binance smart chain (BSC) is 30 million gas [9] and 60 million gas [19], respectively, with block times of 15 seconds and 3 seconds. Thus, for the round length of Bi-FiSA to fit within a desired audit frequency, a customized blockchain platform with the capacity that suits the desired audit frequency should be implemented. If Bi-FiSA is implemented with an existing public blockchain, scalability solutions such as rollups [28] should be considered.

7 Privacy and Threat Analysis

7.1 Privacy Analysis

As the proposed model utilizes a public blockchain platform where all the stored data are visible to anyone, it is important to avoid public disclosure of users' sensitive information. In the case of Bi-FiSA, snapshots of user balances, $\vec{S}^i$ for all i , and encrypted seed values $\vec{E}^i$ are published on the blockchain, thus user balances are the ones that should be protected. Thus, we define the users' balance values as privacy targets.

To address this paramount concern, Bi-FiSA implements a mechanism wherein participating users generate pseudorandom seeds. In order to argue that privacy is preserved, we will first argue that pseudorandom seeds prevent an adversary that does not run a node is not able to computationally identify $\vec{b}^i$ from $\vec{S}^i$ for the i^{th} user, and then argue it applies the same with an adversary that runs a node. Throughout the following descriptions, S_k^i is defined as $b_k^i + s_k^i$.

Lemma 1. *If the seed s_k^i is generated pseudorandomly, an adversary who does not run a node for Bi-FiSA is computationally impossible to identify $\vec{b}^i$ from $\vec{S}^i$.*

Proof. Let x be the element of X , and $\text{Prob}(x)$ the probability of identifying the value of x where $\sum_{x \in X} \text{Prob}(x) = 1$. Also, let $B^i = \{b_k^i \mid b_k^i \text{ for all } k\}$ be the set of balance values of each user for the i^{th} user.

Now, let us assume $B^i = \{b_{k1}^i, b_{k2}^i, \dots\}$. As $b_k^i = S_k^i - s_k^i$, $\text{Prob}(b_k^i) = \text{Prob}(S_k^i - s_k^i)$ holds. As S_k^i is a constant because it is a publicly-posted snapshot, $\text{Prob}(b_k^i) = \text{Prob}(s_k^i)$ holds. For the case of $k \in \{k1, k2, \dots\}$, if s_k^i is generated pseudorandomly, any $s_k^i \in \{s_{k1}^i, s_{k2}^i, \dots\}$ is computationally indistinguishable [17]. Thus, it is computationally impossible to identify b_k^i from S_k^i .

For the case where k is fixed and $i \in \{i1, i2, \dots\}$, $s_k^i \in \{s_{k1}^i, s_{k2}^i, \dots\}$ is also computationally indistinguishable. Thus, it is computationally impossible to identify $\vec{b}^i$ from $\vec{S}^i$. $\square$

Lemma 2. *If the seed s_k^i is generated pseudorandomly, an adversary who runs a node for Bi-FiSA is computationally impossible to identify $\vec{b}^i$ from $\vec{S}^i$, unless the number of nodes the adversary colludes with equals to or greater than thr .*

Proof. Lemma 1 implies that an adversary without the ability to decrypt seeds is computationally impossible to identify $\vec{b}^i$ from $\vec{S}^i$.

An adversary who runs a node, however, possesses sk_i , the private key of the adversary, assuming that the adversary is the i^{th} node. However, unless the number of nodes equals to or greater than thr colludes, decryption is unavailable [10]. Thus, given that the ElGamal encryption scheme is probabilistic [29], it is computationally impossible to obtain $\vec{S}^i$ from $\vec{E}^i$ [18], unless the number of nodes the adversary colludes with equals to or greater than thr .

도면17

As a result, from Lemma 1 and [18], it is computationally impossible to $\hat{b}^i$ from $\hat{S}^i$ for a node without the collusion of the number of nodes the adversary colludes with equals to or greater than thr . $\square$

Combining the lemmas above, we get the following theorem.

Theorem 1. *If the seed s_k^i is generated pseudorandomly, an adversary is computationally impossible to identify a user's plain balance values ($\hat{b}^i$) from the user's snapshot value ($\hat{S}^i$), unless the number of nodes that collude equals to or greater than thr .*

While Bi-FiSA includes privacy-enhancing measures, certain challenges remain. During the process, plain seed values are shared with the exchange, potentially enabling the exchange to access user balances. However, this does not escalate any privacy concerns. CEXs inherently have access to user balance information, a standard practice. Thus, Bi-FiSA's privacy level aligns with or is less intrusive than conventional CEX operations and it **does not amplify** privacy risks beyond typical exchange practices.

7.2 Potential Threat Analysis

7.2.1 Threat Model

While Bi-FiSA promotes user participation in audits, malicious CEXs pose a threat. During the seeding stage, CEXs can exclude users if their seed value comparisons are false, potentially causing trust issues. Such manipulations can impact total reserves audit results.

Threat Model 1. *A malicious CEX can manipulate the seed value comparison results as false during the seeding stage, resulting in manipulation of the total reserves audit result.*

There can also be an issue of trust during the snapshot stage. Assuming a malicious user, the user completes the seeding stage successfully and then can use a different seed to calculate the snapshot. If this is the case, the user can deliberately sabotage the individual and total reserves audits.

Threat Model 2. *A malicious user can deliberately use different seed values at the snapshot stage, resulting in manipulation of the individual and total reserves audit results.*

7.2.2 Threat Mitigation

Assuming malicious CEX or users exploit the system with threat models 1 and 2, trust issues will be raised by users or the CEX, respectively. If so, disputes will occur between users and the CEX.

To resolve such disputes, nodes can decrypt user-encrypted seeds via ADKG to identify audit discrepancies, calculate disputed balances, and verify rightful claims using user trading

records. This decryption capability remains essential, even with honest users and the exchange, as local device issues can manipulate timestamps during synchronized snapshot-taking, potentially causing disputes.

With nodes allowed to decrypt some values under ADKG protocol, a malicious entity that exploits the system through threat models 1 and 2 can be identified through backtracking of published data. Furthermore, in the case of a malicious user, nodes decrypt the user-posted encrypted seed values and trace the source of the difference with data published on the blockchain (including user-snapshot-calculated reserves and exchange-snapshot-calculated reserves). By identifying the malicious user and excluding the user's snapshot from the round, threat model 2 can be mitigated.

8 Conclusion

The crypto asset industry's rapid growth demands transparency in CEXs. The high-profile CEO arrest incident underscores the need for exchange audits. Our research delved into CEX challenges, lacking regulatory oversight, and audit concerns. We introduced Bi-FiSA, enhancing CEX security through ADKG, bidirectional snapshot verification between a CEX and users, and a minimized 'pause period' with the implementation methods of EDH and UDH. This proactive approach bolsters platform soundness, fosters trust by utilizing a bidirectional audit structure, and balances transparency with privacy, addressing user concerns. While Bi-FiSA is expected to be a promising protocol for auditing the financial soundness of CEXs, the protocol appears to have a few opportunities for improvement, so we propose a potential future research agenda:

- On Bi-FiSA, disputes between users and the CEX over trust issues can be resolved by tracking published data and nodes decrypting some of them. Given that the process may require some entities' consents for decryptions, research on the delicate process of dispute-resolving is required.
- Given that blockchain transactions require some fees, users and nodes should be adequately compensated with rewards surpassing participation costs. Thus, research on the adequate distribution of Bi-FiSA participation rewards is required.
- A research on a custom blockchain platform or implementation of scalability solutions such as rollups is required, in order to ensure that Bi-FiSA's round length meets a desired audit frequency.