ChatGPT

수탁형 지갑 설계를 위한 강의 준비 – Ronin 블록체인 해킹 사례와 지갑 설계의 통제 포인트

1. 배경과 개요

크로스체인 브리지와 수탁형 지갑

블록체인 생태계는 서로 독립된 체인들이 많아 자산을 이동하기 위해 **브리지(Bridge)** 라는 연결 계층을 사용한다. 브리지는 소스 체인에서 자산을 **락(Lock)** 하거나 **태워(Burn)** 버린 뒤, 대상 체인에서 같은 양을 **발행(Mint) 또는 언락(Unlock)** 하여 유동성을 제공한다 [1]. 이 과정에서 자산을 맡아주는 **수탁형 브리지**는 특정 주체(검증자 집합 또는 신뢰된 기관)가 사용자의 자산을 보관·서명하기 때문에 보안 문제와 규제 요구사항이 높다. 일반적인 취약점은 다음과 같다 [2]:

- **수탁자 위험**: 소수의 검증자 또는 관리자에게 키가 집중되어 있으면, 키가 유출되거나 관리자가 악의적 행동을 할 때 브리지 전체가 무력화된다. Ronin 해킹은 바로 이러한 구조적 취약점을 노렸다 [3].
- **오라클/릴레이어 조작**: 체인 간 메시지를 전달하는 off-chain 노드를 조작해 가짜 증명을 제출하거나 메시지를 변경하는 공격이 가능하다 [4].
- **스마트컨트랙트 버그**: 브리지 로직의 코드 오류, 초기화 함수 누락, 재진입 취약점 등으로 악용된다. 2024년 8월 Ronin 브리지에서 발견된 버그는 초기화 함수(v3)가 호출되지 않아 검증자 투표 가중치가 0으로 설정되면서 한 번의 결함으로 4,000 ETH와 2M USDC를 도난당한 사례다 [5].

Ronin 브리지 해킹 요약

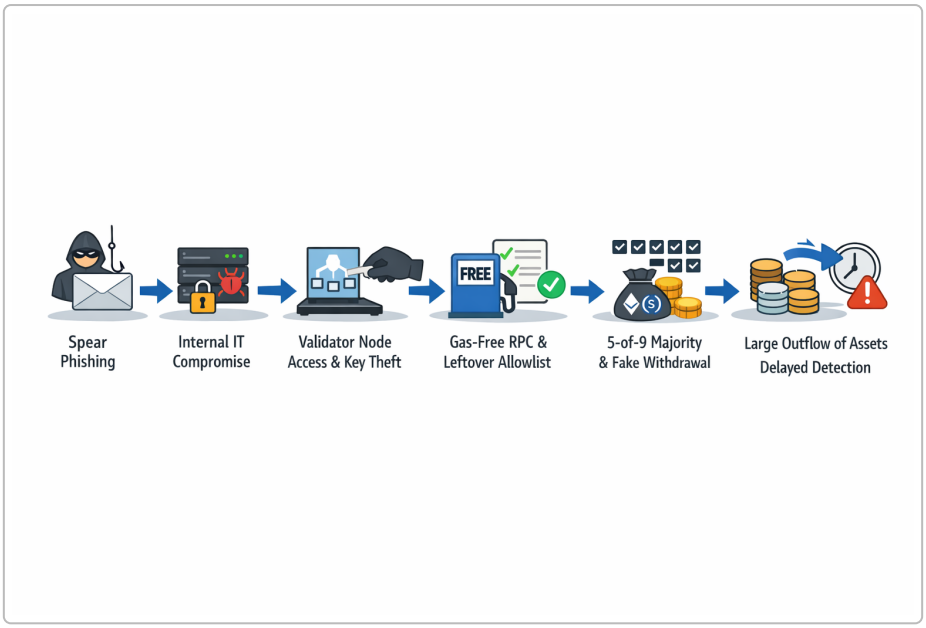
2022년 3월 23일, Axie Infinity 게임을 위해 만든 이더리움 사이드체인 **Ronin**의 브리지에서 173,600 ETH와 25.5 M USDC가 유출되었다 [6]. 이 사건은 6일이 지난 3월 29일에야 사용자 신고로 발견되었고, 미 재무부는 공격 배후를 북한의 **Lazarus Group**으로 지목하며 자금세탁에 사용된 지갑을 제재했다 [7]. 공격은 다음의 몇 단계로 진행되었다:

1. **스피어피싱으로 내부 직원 계정·단말 감염**: 보안업체의 보고에 따르면 공격자는 Sky Mavis 직원 한 명을 표적 피싱으로 감염시켜 내부 IT 인프라 접근 권한을 얻었다 [8].
2. **내부 IT 침해를 통한 검증자 노드 접근**: 감염된 계정을 이용해 Sky Mavis의 내부 네트워크에 침투한 뒤, Ronin 브리지의 검증자 노드가 실행되는 환경에 접근했다 [9].
3. **검증자 키 탈취 및 다중서명 과반 확보**: Ronin 브리지는 9개의 검증자 노드 중 5개 서명을 얻어야 출금이 승인된다. 공격자는 Sky Mavis가 소유한 4개 검증자 키를 탈취했고, 나머지 1개는 과거 **Axie DAO**가 Sky Mavis를 임시 허용(allowlist)하면서 남아 있던 **가스 프리(RPC) 백도어**를 이용해 획득했다 [10].
4. **가짜 출금 서명과 자금 이체**: 다섯 개의 키로 두 건의 대형 출금 트랜잭션을 서명해 173,600 ETH와 25.5 M USDC를 빼냈다 [11].
5. **대형 유출 탐지 지연**: Sky Mavis 내부에 대형 outflow 모니터링 시스템이 없어 6일간 해킹 사실을 알아차리지 못했다 [12].

이 사건은 단순한 코딩 실수나 해커의 영리함이 아닌 **구조적 취약점**이 결합된 결과였다. 다음 절에서는 공격 체인을 단계별로 분석하고 각 단계에서 방어선이 어떻게 무너졌는지 살펴본다.

2. 공격 체인 분석

아래 그림은 Ronin 해킹 과정의 주요 단계를 시각화한 것이다. 공격자는 사람을 속이는 피싱 공격으로 시작하여 내부 시스템에 발판을 마련했고, 최종적으로 서명 키와 잔존 권한을 이용해 브리지 자산을 탈취했다.



2.1 피싱과 내부 침해

- **스피어피싱**: 보고된 바에 따르면 공격자는 Sky Mavis 개발자를 대상으로 한 맞춤형 피싱 메일을 보내 악성 소프트웨어를 설치하게 했다 [8].
- **내부 IT 인프라 침해**: 감염된 단말을 통해 내부 VPN·CI/CD 서버에 접근하고 신뢰할 수 있는 네트워크로 이동했다. 이는 브리지 운영 환경과 업무용 환경이 분리되어 있지 않았음을 의미한다 [8].

2.2 검증자 노드와 키 탈취

- **검증자 노드 집중**: Ronin 체인은 9개의 검증자 노드 중 5개 서명으로 출금을 승인하는 구조인데, Sky Mavis가 4개의 노드를 직접 운영했다 [3]. 중앙화된 운영이 공격자가 적은 노력으로 과반수를 탈취할 수 있게 했다.
- **키 저장 환경의 문제**: 탈취된 키들은 업무용 인프라와 같은 네트워크 내에 저장되어 있었고 별도 물리적·논리적 분리가 없었다. SlowMist의 분석에서 "직원 계정 침해 → IT 인프라 침투 → validator 노드 접근 → 키/서명권 도달" 순으로 공격이 진행되었다고 한다 [13].

2.3 백도어: Gas-Free RPC와 잔존 권한

- **임시 allowlist의 영구화**: 2021년 11월 Sky Mavis는 사용자 트래픽 폭증을 처리하기 위해 Axie DAO에게 자신들을 대신해 트랜잭션을 서명하도록 요청했고, DAO는 Sky Mavis를 허용리스트에 등록했다 [14]. 12월에 해당 프로그램은 종료됐지만, 허용 권한은 철회되지 않았다 [15].
- **Gas-Free RPC 노드**: Sky Mavis가 운영하던 가스 프리 RPC 노드는 사용자들이 수수료 없이 트랜잭션을 보낼 수 있도록 돕는 서비스였다. 공격자는 이 노드의 백도어를 활용해 Axie DAO 검증자 키의 서명을 대신 생성했다 [16]. 이로써 총 5개의 서명 키를 확보해 가짜 출금 요청을 승인했다.

2.4 출금과 탐지 지연

- **가짜 출금 승인**: 5개 서명으로 두 건의 출금 트랜잭션을 만들어 Ronin 브리지 계약에서 173,600 ETH와 25.5 M USDC를 인출했다 [11].

- **탐지 실패**: Sky Mavis는 대형 출금 모니터링 시스템을 갖추지 않아 6일 후 사용자 제보로 사건을 인지했다 [12]. 사후보고에서는 향후 "대형 outflow는 사람介入 없이는 불가"하도록 절차를 변경하겠다고 밝혔다 [17].

3. 구조적 취약점과 통제 포인트

Ronin 해킹은 단순한 인간 실수나 특정 코드의 취약점보다 **지갑 및 브리지 설계의 구조적 문제**가 핵심 원인이었다. 여기서는 사건의 취약점을 **세 가지 범주**로 정리하고, 이를 해결하기 위한 **통제 포인트(Control Points)**를 제시한다.

A. 키 관리와 환경 분리의 실패

취약점: 다중 서명을 구성하는 9개의 검증자 중 4개를 Sky Mavis가 운영했고, 이 키들이 내부 업무 시스템과 같은 네트워크에서 관리되었다. 직원 계정이 침해되자 곧바로 검증자 노드와 키에 접근할 수 있었다 [13].

통제 포인트:

- **물리·논리적 분리**: 서명 키를 보관하는 환경은 업무용 IT 인프라와 완전히 분리해야 한다. 전용 네트워크, 별도 단말 또는 **하드웨어 보안 모듈(HSM)/하드웨어 지갑**을 이용하고, 서명 장치는 외부 인터넷에 연결하지 않는 것이 바람직하다. SEAL의 다중서명 가이드라인은 모든 서명자가 하드웨어 지갑을 사용하고, 서명용 키를 저장한 장치가 서로 다른 물리적 위치에 분산되어야 한다고 강조한다 [18].
- **분산된 서명자 구성**: 중앙 조직이 다수의 검증자 키를 보유하지 않도록 다양한 주체에 서명 권한을 분산하고, 지리적·운영적 다양성을 확보해야 한다 [19]. SEAL 가이드라인은 최소 3명 이상의 서명자, 50% 이상 서명 임계값, 100만 달러 이상 자산을 보관하는 경우 7명 이상의 서명자를 권장한다 [20].
- **안전한 서명 프로세스**: 서명 요청은 별도의 승인 경로를 거쳐야 하며, 서명자는 **EIP-712**와 같이 구조화된 메시지(도메인, 체인 ID, 검증 대상 고정)를 검증하고 서명해야 한다. 이는 피싱과 서명 위조 공격을 줄인다.

B. 잔존 권한과 변경 관리 부족

취약점: 임시 트래픽 대응을 위해 Axie DAO가 Sky Mavis를 허용리스트에 올렸고, 프로그램 종료 후에도 권한이 자동으로 회수되지 않아 백도어가 남았다 [15]. 이는 "일시적 예외"가 영구 취약점이 되는 전형적인 사례다.

통제 포인트:

- **만료 설정과 자동 회수**: 권한 부여 시 기본 만료 기간을 설정하고, 기간이 만료되면 자동으로 권한을 회수하도록 설계한다. 시스템적으로 **expiry** 값을 반드시 입력하도록 하고, 주기적으로 권한 목록을 검토·정비해야 한다.
- **변경 관리와 책임 분리**: 권한 부여와 철회를 동일한 사람이 할 수 없도록 분리하여 최소 2인 이상의 승인을 요구한다. change management 프로세스를 통해 수정 기록과 승인 절차를 로그로 남기고, 감사 가능한 형태로 관리해야 한다.
- **정책 엔진의 화이트리스트 관리**: 정책 엔진 모듈에서 출금 대상 화이트리스트를 유지할 때는 등록·갱신·삭제에 대한 엄격한 승인 절차와 만료 조건을 포함해야 한다. Halborn은 다중체인 스테이블코인 보안을 위한 베스트 프랙티스로 모든 배포 코드에 대한 **엄격한 감사와 다중서명 지갑** 사용을 권고하며 [21], 이는 브리지 운영에도 적용된다.

C. 감시와 탐지의 부재

취약점: Ronin은 대형 출금/비정상 패턴을 실시간으로 감지할 시스템이 없어 6일간 이상을 모니터링하지 못했다 [17].

통제 포인트:

- **실시간 모니터링 및 알림**: **Confirmation Tracker** 모듈은 단순히 컨펌 수를 세는 역할을 넘어, 출금 규모·빈도·대상 변화 등을 감지하고 비정상 행동을 알림 또는 자동 차단 트리거로 연결해야 한다.

- **한도 및 속도 제한**: 정책 엔진에서 출금 한도, 일일 속도 제한, 특정 기간 동안의 최대 금액 등을 설정해 이를 초과하면 자동으로 트랜잭션을 중단하거나 추가 승인을 요구한다 [21].
- **사람介入**: 대형 출금은 반드시 인간 검토를 거치도록 하고, 휴지기간(time delay)을 두어 커뮤니티나 감사팀이 검토할 시간을 확보한다 [22]. SEAL의 권고는 타임락과 **비토(quorum) 기능**을 두어 긴급 상황에서만 우회하도록 한다 [23].

4. 7모듈 설계와 통제 포인트 체크리스트

수탁형 지갑 설계에는 업무 흐름을 분리한 7개 모듈이 있다. 각 모듈은 고유한 책임을 가지며, Ronin 사건에서 확인된 취약점을 방지하려면 아래와 같은 통제 포인트를 갖춰야 한다.

모듈	목적 / 역할	사건에서 무너진 점과 개선방안
Policy Engine	출금 한도·속도 제한, 허용 주소 목록 관리, 이상징후 점수 산출	허용리스트에 만료·자동 회수가 없었다. 명시적 만료일을 기본값으로 설정하고 주기적 검토를 적용한다 [15]. 출금 한도와 속도 제한을 설정하여 대형 출금은 자동 중단 후 별도 승인 절차를 거치게 한다 [21]. 이상징후 점수를 통해 평소 대비 금액·빈도·대상 변화를 감지하고 정책 위반을 막는다.
Approval	트랜잭션 승인 관리(다중 승인, 긴급 중지 기능)	Ronin은 5/9 서명 임계값만 충족하면 자동 실행되도록 설계돼 대형 출금에 대한 추가 인간 검토가 없었다. 대형 출금은 다중 승인과 사람이介入하는 검토 과정을 필수로 하고, 긴급 정지·일시 중단(Freeze) 절차를 도입한다.
Signer	서명 키 보관 및 서명 실행. 서명 요청의 형식과 도메인 검증(EIP-712 등)	검증자 키가 업무 IT와 동일한 환경에 있었고, 스피어피싱으로 키가 탈취되었다 [13]. 서명 키는 HSM이나 하드웨어 지갑에 저장하고, 서명 장치는 업무용 네트워크와 분리한다 [18]. 서명자는 구조화된 메시지(EIP-712)에서 도메인·체인·검증 계약 등을 확인하고 서명해야 한다.
Broadcaster	서명된 트랜잭션을 블록체인 네트워크에 전파. 다중 RPC 사용, 전송 경로 다양화	Ronin은 gas-free RPC 노드 하나를 운영했고, 백도어가 발견됐다 [16]. 브로드캐스터는 여러 RPC 엔드포인트를 사용하고, 각 경로에서 메시지 위·변조 여부를 검증해야 한다. 특정 RPC 노드에 예외 권한을 부여할 경우 만료와 모니터링을 반드시 설정한다.
Confirmation Tracker	체인에서 트랜잭션이 확정되는 과정을 추적, 대형 outflow 감지 및 알림	Ronin은 대형 outflow를 즉시 탐지하지 못했다. Confirmation Tracker는 컨펌 수 외에도 금액·패턴을 분석하고, 설정된 한도를 초과하는 트랜잭션을 알림과 함께 자동 차단 트리거로 전환한다.
Ledger & Audit	내부 원장과 블록체인 이벤트를 대사하고 감사 로그를 제공	내부 IT 침해와 잔존 권한을 정기적으로 감사하지 않아 권한 오남용을 감지하지 못했다. 원장 이벤트와 체인 이벤트를 일치시키고, “누가·왜·어떤 근거로 승인했는지”를 기록해야 한다. 외부 감사와 정기 검토를 통해 미사용 권한 및 허용리스트를 점검한다.

모듈	목적 / 역할	사건에서 무너진 점과 개선방안
API/Orchestrator	모든 출금 요청에 고유 ID를 부여하여 정책→승인→서명→전파→확정까지 전체 흐름을 추적	Ronin에서는 공격자가 내부 시스템을 장악한 후 여러 단계를 우회해 서명까지 이루어졌다. API는 각 단계의 상태를 명시적으로 표시하는 **상태 머신**을 구현하여 누락된 단계가 있으면 다음 단계로 진행할 수 없게 해야 한다. 또한 요청의 추적 정보를 감사 로그에 저장하고, 외부 모듈과 통신 시 인증 및 권한 검사를 철저히 해야 한다.

5. 상태 머신과 전반적인 설계

브리지 출금 프로세스는 각 모듈을 거치며 **상태(state)** 가 변하는 상태 머신으로 볼 수 있다. 상태 머신을 명확히 정의하면 한 단계가 누락되거나 권한이 초과될 때 감지할 수 있다. 예를 들어 다음과 같은 상태를 정의할 수 있다:

1. **Requested**: 사용자가 출금을 요청하면 API/Orchestrator는 고유 ID를 할당하고 초기 상태로 저장한다.
2. **PolicyChecked**: Policy Engine이 요청을 검토하여 금액·대상·빈도 등의 정책을 통과했는지 판단한다. 실패하면 거절 상태로 전환한다.
3. **Approved**: Approval 모듈에서 다중 승인 및 필요 시 인간介入을 통해 트랜잭션을 승인한다. 대형 출금이면 추가 확인과 타임락을 거친다.
4. **Signed**: Signer 모듈이 구조화된 메시지를 검증하고, 정해진 형식(EIP-712)에 따라 서명한다. HSM/하드웨어 지갑에서 발생된 서명을 기록한다.
5. **Broadcasted**: Broadcaster 모듈이 서명된 트랜잭션을 여러 RPC 경로로 전파한다.
6. **Confirmed**: Confirmation Tracker가 체인 상의 컨펌 수와 금액, 패턴을 추적하고 정상적으로 확정됐음을 기록한다.
7. **Logged**: Ledger & Audit 모듈이 모든 단계의 로그와 체인 이벤트를 매핑해 최종 기록을 완료한다.

이러한 상태 머신 설계는 모듈 간 **책임 분리**와 **가시성**을 확보하여, Ronin 사례처럼 내부 IT 침해나 잔존 권한으로 단계가 건너뛰어도 시스템이 이를 허용하지 않도록 한다. 또한 각 상태 전환 시점에 **감사 로그**를 남겨 추적성과 규제 준수(예: SOC2, ISO 27001)를 보장한다.

6. 추가 사례와 지속적인 위협

2024년 8월 Ronin 브리지 재해킹 사례

2024년 8월 Ronin 브리지는 **스마트컨트랙트 업그레이드 중 초기화 함수(v3)가 호출되지 않는** 오류로 또다시 공격을 받았다. 업그레이드 과정에서 `_totalOperatorWeight` 값을 설정하는 v3 초기화 함수가 호출되지 않아 최소 투표 가중치가 0으로 초기화되고, 모든 트랜잭션이 즉시 승인되는 상태가 되었다. MEV 봇 운영자는 이 취약점을 이용해 **4K ETH와 2M USDC를 탈취**했으나, 화이트해커 역할을 해 자금을 반환했고 Ronin 팀은 50만 달러의 버그 바운티를 지급했다 [24]. 이 사건은 **업그레이드 코드에 대한 감사**와 초기화 함수 호출 확인의 중요성을 다시 일깨워 주었다 [25].

크로스체인 브리지 보안 베스트 프랙티스

- **철저한 코드 감사와 지속적 모니터링**: Halborn은 다중체인 스테이블코인 보안에서 모든 배포 코드에 대한 정기적인 감사와 실시간 모니터링이 필요하다고 강조한다 [21]. 스마트컨트랙트는 배포 전·후에 formal verification과 동적 분석을 수행해야 한다.
- **분산된 검증자 집합과 투표 가중치 관리**: 검증자 수를 충분히 늘리고(예: Ronin은 해킹 후 11개에서 21개, 장기적으로 100개 이상의 노드를 목표로 한다 [26]), 단일 주체가 과반을 차지하지 못하도록 한다. 또한 브리지의 계약 변수(예: 최소 투표 가중치)를 업그레이드 시에 정확히 설정해야 한다 [5].

- **다중 요소와 키 분할**: MPC(다자간 계산)나 Shamir Secret Sharing 등을 사용해 키를 분할하면 하나의 키를 탈취해도 전체 서명을 생성할 수 없다. Cobo 등 기관용 커스터디 업체는 키를 여러 노드에 분산 저장하고, HSM과 MPC를 결합한 아키텍처를 제공한다 [27].
- **오라클·릴레이어 보안 강화**: 브리지의 off-chain 컴포넌트(릴레이어, 센서 노드 등)는 별도의 무결성 검증, 서명 및 투표 메커니즘을 가져야 한다. Quantstamp의 SoK 논문은 Celer Bridge에서 하나의 검증자가 동일한 업데이트에 여러 번 투표해 공격을 수행할 수 있었으며, off-chain 코드에 대한 보안 감사와 버그 바운티 프로그램 확대가 필요함을 강조한다 [28].
- **제로 트러스트와 인적 보안**: Ronin 사후보고에서 Sky Mavis는 "제로 트러스트 조직"을 목표로 삼겠다고 밝혔다 [29]. 이는 모든 내부/외부 연결에서 지속적으로 신원과 권한을 검증하며, 직원 교육을 강화하고 업무용 디바이스를 별도로 관리하는 것을 의미한다.
- **규제 준수와 제재 대응**: Lazarus Group과 같은 제재 대상이 브리지 해킹에 연루되면서 미국 OFAC는 관련 지갑 주소를 SDN 리스트에 추가했다 [30]. 커스터디 사업자는 제재 명단을 실시간으로 모니터링하고, 거래 차단 및 보고 의무를 준수해야 한다.

7. 요약 및 강의 포인트

- **사건 요약**: 2022년 3월 23일 Ronin 브리지 해킹은 스피어피싱–내부 IT 침해–검증자 키 탈취–잔존 allowlist–가짜 출금 승인–탐지 지연이라는 다단계 공격이었다 [31]. 약 6일 동안 173,600 ETH와 25.5 M USDC가 유출되었고, 미 재무부는 Lazarus Group을 제재했다. 2024년 8월에는 초기화 함수 호출 누락으로 또다시 12 M 달러 규모의 해킹이 발생했다 [24].
- **구조적 취약점**: (A) 서명 키와 업무 IT를 분리하지 않아 스피어피싱으로 키가 탈취될 수 있었다 [13]. (B) 임시 허용 권한을 철회하지 않아 Gas-Free RPC를 통한 백도어가 남았다 [15]. (C) 대형 출금 모니터링이 부재해 탐지가 늦었다 [17].
- **설계 교훈**: **정책 관리**, **다중 승인**, **키 분산 및 HSM**, **백도어 만료와 변경 관리**, **실시간 모니터링**, **상태 머신 구현** 등을 통해 이러한 취약점을 해결할 수 있다. SEAL의 다중서명 가이드라인은 하드웨어 지갑 사용과 지리적/역할적 다양성을 강조하며 [18], Halborn은 엄격한 코드 감사와 사고 대응 계획의 중요성을 설명한다 [32].
- **모듈별 체크리스트**: Policy Engine에서 출금 한도와 화이트리스트 만료를 관리하고, Approval 모듈에서 대형 출금에 사람介入과 긴급 정지 기능을 추가한다. Signer 모듈은 키를 HSM/하드웨어 지갑에 보관하고 구조화된 서명 형식을 검증한다. Broadcaster는 여러 RPC 경로를 사용하며 백도어 권한을 최소화한다. Confirmation Tracker는 대형 outflow를 감지·차단하고, Ledger & Audit은 모든 이벤트를 대사해 감사 가능성을 높인다. API/Orchestrator는 요청을 상태 머신으로 관리해 누락된 단계가 없도록 한다.

이 보고서를 바탕으로 강의에서는 사건의 세부 흐름과 각 모듈의 역할을 명확히 설명하면서 "해커가 똑똑해서가 아니라 우리의 설계가 무엇을 허용했는가"를 강조해야 한다. 또한 2024년 이후의 추가 사례와 최신 베스트 프랙티스를 제시해 지속적인 개선과 규제 준수의 중요성을 강조할 수 있다.

---

[1] [2] [4] Simplex | Cross-chain bridges explained: multi-chain asset transfers | Simplex
https://www.simplex.com/cross-chain-bridges-explained

[3] [7] [10] [11] [16] [31] Hack Track: Analysis of Ronin Network Exploit | Merkle Science
https://www.merklescience.com/blog/hack-track-analysis-of-ronin-network-exploit

[5] [24] [25] Explained: The Ronin Network Hack (August 2024)
https://www.halborn.com/blog/post/explained-the-ronin-network-hack-august-2024

[6] [8] [9] [12] [14] [15] [17] [19] [26] [29] Ronin Blog | Back to Building: Ronin Security Breach Postmortem
https://roninchain.com/blog/posts/back-to-building-ronin-security-breach-6513cc78a5edc1001b03c364

[13] Report on the Ronin Network Exploit and AML Analysis of Stolen Funds | by SlowMist | Medium
https://slowmist.medium.com/report-on-the-ronin-network-exploit-and-aml-analysis-of-stolen-funds-692b2a589a96

18 20 22 23 Secure Multisig Best Practices | SEAL
https://frameworks.securityalliance.org/wallet-security/secure-multisig-best-practices/

21 32 Multi-Chain Stablecoins: Security, Risks and Best Practices
https://www.halborn.com/blog/post/multi-chain-stablecoins-security-risks-and-best-practices

27 Evaluating Crypto Custody Firms: Institutional Guide 2025
https://www.cobo.com/post/the-definitive-guide-to-evaluating-crypto-custody-firms-for-institutional-investors

28 SoK: A Review of Cross-Chain Bridge Hacks in 2023
https://arxiv.org/html/2501.03423v1

30 OFAC Sanctions Tracker: How Sanctions Impact Crypto Crime - Chainalysis
https://www.chainalysis.com/blog/ofac-sanctions/