# ZSL 개념증명 (POC)

* [개요](#bookmark=id.gjdgxs)
* [구현](#bookmark=id.30j0zll)
* [주식 거래 적용 예시](#bookmark=id.1fob9te)
* [프로토콜](#bookmark=id.3znysh7)

## 개요

Quorum은 퍼블릭 콘트랙트 (표준 이더리움 방식으로 실행되며 분산 원장의 모든 참여자가 볼 수 있음)와 프라이빗 콘트랙트 (Tessera를 사용하는 프라이빗 콘트랙트의 당사자 간에 공유되지만 다른 사람이 읽을 수는 없음)를 모두 지원합니다. 이 접근법은 프라이빗 콘트랙트 당사자의 프라이버시를 보존하고 프라이빗 콘트랙트의 비즈니스 논리를 기밀로 유지해 줍니다. 그러나 중요한 제한 사항은 프라이빗 콘트랙트 내에서 교환되는 디지털 자산에 대한 이중 지출 방지를 지원하지 않는다는 것입니다.

ZSL (0지식 보안 레이어, zero-knowledge security layer)은 zk-SNARKS를 활용하여 발신자, 수신자 또는 자산 수량에 대한 정보를 공개하지 않고 분산원장에서 디지털 자산을 전송할 수 있도록 Zcash 팀이 설계한 프로토콜입니다.

J.P. Morgan과 Zcash팀은 ZSL이 도입된 퍼블릭 스마트 콘트랙트(z-contracts)을 사용하여 디지털 자산을 발행할 수 있도록 Quorum에 ZSL의 개념증명(POC)을 구현하기 위해 파트너십을 체결했습니다. 이렇나 디지털 자산을 ‘z-토큰’이라고 합니다. Z-토큰은 공개되지 않고 프라이빗하게 거래될 수 있습니다. 차폐 거래가 실행된 증거는 프라이빗 콘트랙트에 나타나며 퍼블릭 z-콘트랙트를 사용하여 실행된 차폐 거래에 따라 프라이빗 콘트랙트는 자신의 상태를 업데이트합니다.

Tessera의 프라이빗 콘트랙트와 ZSL의 z-콘트랙트의 결합은 완전한 개인정보 및 기밀을 유지하면서도 프라이빗 콘트랙트로부터 발생하는 의무를 해결할 수 있게 합니다.

더욱 많은 정보는 [POC Technical Design Document](https://github.com/jpmorganchase/zsl-q/blob/master/docs/ZSL-Quorum-POC_TDD_v1.3pub.pdf)를 참조해 주십시오.

## 구현

ZSL의 개념증명은 다음과 같이 구현됩니다. \* ZSL 특정 코드는 [zsl-q](https://github.com/jpmorganchase/zsl-q) 저장소에 있습니다. \* Quorum의 통합은 Quorum 저장소(the [zsl\_geth1.5](https://github.com/jpmorganchase/quorum/tree/zsl_geth1.5) branch)의 별도 브랜치로 구현되었습니다. \* Quorum 예시 저장소를 위한 ZSL 특정 브랜치 또한 존재하며 (the [zsl\_geth1.5](https://github.com/jpmorganchase/quorum-examples/tree/zsl_geth1.5) branch) \* [zsl-q-params](https://github.com/jpmorganchase/zsl-q-params) 저장소는 zk-SNARK 증명을 생성하고 검증하는 데 필요한 공유 매개 변수를 포함합니다.

ZSL이 도입된 Quorum을 설치하는 방법에 대한 상세한 가이드는 [zsl-q README](https://github.com/jpmorganchase/zsl-q/blob/master/README.md)를 참조하십시오.

이 개념증명(POC)은 ZSL이 Quorum을 보완할 수 있는 방법을 보여주고 다양한 사례를 실험하고 탐색할 수 있는 플랫폼을 제공하기 위한 것입니다. 신속한 프로토타이핑이 가능하도록 Zerocash 프로토콜의 간소화된 버전을 구현합니다. 프로토콜에 대한 공식적인 보안 증거는 없으며 증명 검증을 위한 예외 처리가 구현되지 않았고 소프트웨어는 엄격한 테스트는 거치지 않았기 때문에 **생산 준비 완료 단계라고 간주되어서는 안됩니다.**

대체로, Quorum ZSL은 가상 자금이 암호화되어 난독화된 ‘노트’에 묶여들어 갈 수 있도록 하는 콘트랙트를 제공합니다. 각 노트는 가치 저장소를 나타내며 비밀 지출 키로만 잠금 해제하거나 ‘상환’할 수 있습니다. 예를 들어, 프라이빗 전송을 수행하기 위해 Alice는 값을 노트에 묶어서 비공개 오프 체인 채널을 통해 노트의 비밀 키를 Bob에게 전송할 수 있습니다. 그런 다음 Bob은 이 노트를 온체인에서 사용하면서 그 과정에서 Alice와 자신 사이의 퍼블릭 링크를 공개하지 않을 수 있습니다. 이전 버전에서는 이더리움과 노트 서명을 연결하지 못한 것이 일종의 ‘전면 실행’ 공격을 가능하게 했습니다. 이 점은 PR [#587](<https://github.com/jpmorganchase/quorum/pull/587>)에 의해 수정되었습니다.

## 주식 거래에서의 활용 사례

다음 예시는 Alice가 Bob으로부터 ACME 주식을 구매하는 단순한 주식거래를 통해 ZSL이 도입된 Quorum의 특정 사용 사례를 보여줍니다. 개념증명(POC)에는 이 사례를 구현하는 데모가 포함되어 있습니다. 이를 실행하기 위한 가이드는 [여기](https://github.com/jpmorganchase/zsl-q/blob/master/README.md#example-2---private-contract-trade)를 참조하십시오. .

[Quorum 주식 거래 활용 사례 다이어그램]Quorum 주식 거래 활용 사례 다이어그램

### 사례의 전제:

* Z-contracts는 US달러(USD z-콘트랙트)와 ACME주식 거래를 위해 만들어졌습니다.
* Z-토큰은 관련 발급기관에 의해 두 콘트랙트 모두에 발행이 된 다음 차폐되어 Alice와 Bob에게 양도되었습니다.
* Alice는 USD z-토큰을 소유하고 있고 Bob은 ACME z-토큰을 소유하고 있습니다. 둘 모두의 소유 상태는 차폐되어 있습니다. (제 3자는 누가 어떤 것을 소유하고 있는지 알지 못합니다.)

### 사용자 스토리:

1. **Tessera를 사용하여 Alice와 Bob 간에 프라이빗 콘트랙트가 설정됩니다.** 
   1. 프라이빗 콘트랙트는 주식 거래 상에서의 두 특정 당사자인 Alice(ACME 주식을 매수하는 사람)와 Bob(ACME 주식을 매도하는 사람), 거래 시 ACME 주식의 특정 수량과 그 USD환산 가격을 지정합니다.
   2. 프라이빗 콘트랙트는 USD 및 ACME z-콘트랙트, 관련 퍼블릭키 및 당사자의 페이먼트 주소를 포함합니다.
   3. 일방이 계약을 초기화합니다. (이는 매수/매도와 동일한 개념입니다.) 어떤 당사자가 해당 작업을 수행하는지는 중요하지 않습니다. 본 예시에서는 Alice로 하겠습니다.
   4. 초기화 된 후 계약 상태는 “매수” (Bob이 초기화하였다면 “매도”) 입니다.
2. **상대방은 프라이빗 콘트랙트에 조건의 수락을 나타내는 트랜잭션을 보냅니다.**
3. 본 예시에서는 Bob이 Alice의 매수를 수락하는 것으로 하겠습니다.
4. 이 시점에서 거래가 “완료”됩니다. (즉, 조건이 합의되고 양 당사자가 거래에 참여하게 됩니다.) 이제 남은 것은 조정 단계입니다. 달러가 우선 지불되어야 한다고 가정해보겠습니다.
5. 콘트랙트 상태: 완료
6. **프라이빗 콘트랙트가 지불을 지시합니다.**
7. 콘트랙트 상대가 “완료”로 업데이트 되면 콘트랙트는 구매자(여기서 Alice)의 클라이언트에게 상응하는 금액의 USD를 판매자(Bob)에게 지불하도록 지시합니다.
8. Alice의 클라이언트는 해당 지시를 받아 대기상태로 놓고 차폐 지불을 지시합니다.
9. **구매자는 판매자에게 USD를 지불합니다.**
10. Alice는 필요한 zk-SNARK 증명을 생성하고 이를 USD z-콘트랙트로 내보내 Bob의 USD 지불주소로 거래에 상응하는 금액의 USD z-토큰을 지불합니다.
11. 차폐 거래가 발생하여 Bob만이 사용할 수 있는 z-콘트랙트 내역이 생성됩니다. (즉, Bob의 z-토큰 잔액이 증가합니다.)
12. 이에 따라 Alice의 USD z-토큰 잔액은 감소합니다.
13. **구매자는 프라이빗 콘트랙트에 지불증거를 제공합니다. ,**
14. Alice는 USD 지불의 결과에 대한 노트를 포함한 트랜잭션을 프라이빗 콘트랙트로 보냅니다.
15. 이는 Bob에게도 노트를 전송하여 Bob 또한 이를 사용할 수 있습니다.
16. **해당 프라이빗 콘트랙트는 지불을 확인해줍니다.**
17. 프라이빗 콘트랙트는 해당 지불이 유효한지 확인하기 위해 Alice가 제공한 노트를 이용하여 USD z-콘트랙트에 지속적으로 함수를 호출합니다.
18. z-콘트랙트는 바이너리 방식으로 응답하여 노트 커미트먼트가 z-콘트랙트의 노트 누적기에 있는지 여부(이 경우 차폐된 지불이 유효함)를 나타냅니다.
19. 유효한 경우, 콘트랙트의 상태는 “지불 완료”로 업데이트되고...
20. **프라이빗 콘트랙트는 “딜리버리”를 지시합니다.**
21. 프라이빗 콘트랙트는 판매자(여기서 Bob)의 클라이언트에게 구매자에게 상응하는 양의 ACME 주식을 구매자에게 양도도록 지시를 내립니다.
22. Bob의 클라이언트는 해당 지시를 받아 대기상태로 놓고 지불을 요청합니다.
23. **판매자는 ACME 주식을 구매자에게 양도합니다.**
24. Bob은 필요한 zk-SNARK 증명을 생성하고 이를 ACME z-콘트랙트로 전송하여 상응하는 ACME z-토큰을 Alice의 지불 주소로 보냅니다.
25. 차폐 거래가 이루어지고 Alice만 사용 가능한 노트 아웃풋을 생성합니다. (즉, Alice의 ACME z-토큰 잔액이 증가합니다.)
26. 이에 따라 Bob의 ACME z-토큰 잔액은 감소합니다.
27. **판매자는 프라이빗 콘트랙트에 해당 전송에 대한 증거를 제출합니다.**
28. Bob은 ACME 전송에 대한 아웃풋 노트를 포함한 트랜잭션을 프라이빗 콘트랙트에 보냅니다.
29. 이는 Alice에게도 노트를 전송하여 이를 “사용”할 수 있도록 합니다. (즉, 해당 토큰들을 다른 사람들에게 전송할 수 있게 됩니다.)
30. **프라이빗 콘트랙트가 딜리버리를 확인합니다.**

프라이빗 콘트랙트는 Bob이 제공한 노트를 이용하여 ACME z-콘트랙트를 호출(상수 함수를 사용)하며 전송이 유효한지 확인합니다.

유효한 경우, 콘트랙트 상태가 “정산됨(Settled)”으로 업데이트됩니다.

Alice가 USD z-토큰을 5번째 단계에서와 같이 Bob에게 전송을 완료하면 Bob은 해당 토큰들을 제 3자(에로, Carol)에게 전송할 수 있게 됩니다. \*Carol은 토큰의 출처는 알 수 없습니다. (Bob이 Alice에게 해당 토큰을 받았음을 알지 못합니다.) \*Alice는 Bob이 제 3자에게 언제 해당 토큰을 전송하는지, 누구에게 전송하는지 알 수 없습니다. Alice는 (트랜잭션이 Alice가 접근권한을 가지고 있는 메인 Quorum 체인 상의 z-콘트랙트에 작성되었기 때문에) 거래가 일어났다는 사실은 볼 수 있지만 보낸 사람, 받는 사람 및 전송된 토큰의 수량은 알 수 없습니다. \*앞의 내용은 Alice가 Bob으로부터 취득한 ACME z-토큰의 사례에도 적용됩니다.

### 프로토콜

아래의 다이어그램은 위의 예시에서 암호화 프로토콜이 1단계에서 6단계까지를 지원하는 방법을 보여줍니다.

[ZSL/Quorum Proof of Concept Protocol (v0.4)]ZSL/Quorum Proof of Concept Protocol (v0.4)