37 신뢰 실행 환경을 활용한 스마트 컨트랙트 실행 보호 기술 개발

소속 정보컴퓨터공학부 분과 D 팀명 TrustForge 참여학생 이준태, 이준혁, 위재준 지도교수 권동현

과제 개요

과제 배경

✓ 블록체인의 스마트 컨트랙트는 무결성과 투명성은 보장하지만, 실행 환경 자체가 일반 운영체제에 의존하기 때문에 악성 코드나 공격자에 의해 위변조될 위험이 존재함

✓ TZ4Fabric은 체인코드를 ARM TrustZone에서 실행하여 보안성을 강화하였지만 원격 증명 기능이 부족하고 성능 저하 문제가 존재

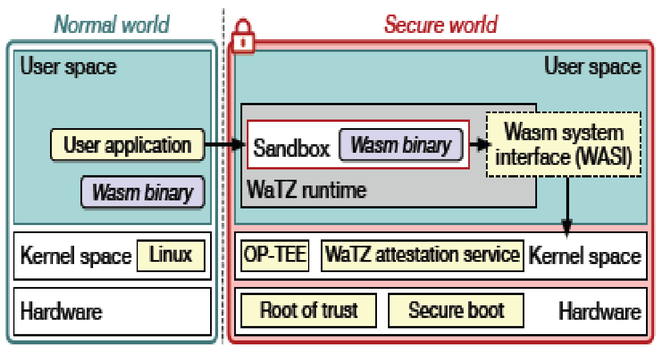
✓ WaTZ는 WebAssembly 기반 체인코드를 TrustZone에서 실행하고 Remote Attestation을 지원하여 보안성과 이식성을 높였지만, Hyperledger Fabric과의 연동성은 고려하지 않음

과제 목표

✓ TZ4Fabric과 WaTZ를 결합하여 WebAssembly 기반 체인코드를 TrustZone에서 안전하게 실행하고, Proxy 구조를 통해 Hyperledger Fabric과 연동하여 신뢰성 있는 스마트 컨트랙트 실행 보호 기술을 구현하는 것

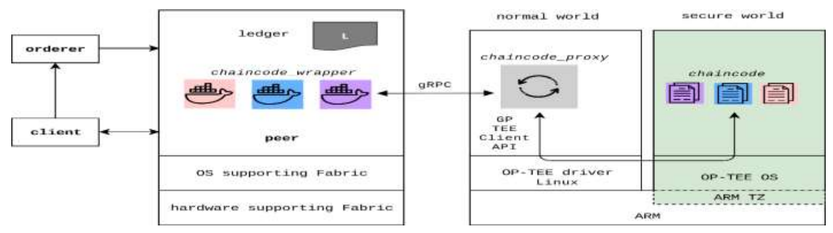
과제 내용

WaTZ 구조



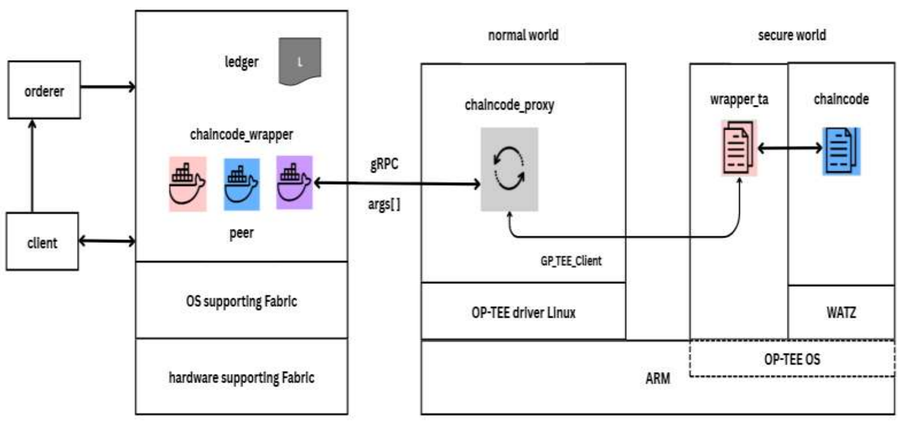
- TrustZone 내부에서 Wasm 모듈 실행
- WASI 인터페이스로 범용성, 이식성 제공
- Sandbox 실행으로 코드 격리 보장
- Remote Attestation 지원으로 실행 무결성 검증

TZ4Fabric 구조



- Hyperledger Fabric 체인코드를 TEE에서 보호 실행
- Wrapper-Proxy-Chaincode 구조 적용
- Peer는 Normal world, 체인코드는 Secure World에서 동작
- 체인코드 결과를 Ledger에 기록

WaTZ + TZ4Fabric 통합 실행 흐름



1. Invoke 요청: 피어노드가 Porxy에 체인코드 Invoke 요청을 전송
2. 세션 생성: Porxy는 요청을 수신하여 TEE 내부에서 세션을 열고, Wasm 체인코드 실행 환경을 초기화함
3. 체인코드 실행 시작: Wrapper_TA가 WaTZ 런타임을 이용해 Wasm 체인코드를 Secure World에서 실행함
4. 체인코드 로직 처리: 체인코드 로직이 실행되며, 실행 중 get_state/put_state 요청이 발생할 경우 Wrapper_TA 내의 Native Function을 호출함
5. 상태 접근 요청: Native Fuction이 호출되면 현재 실행 컨텍스트를 보존한 뒤, Porxy에 get_state 또는 put_state 요청을 전달
6. 원장 접근: Proxy는 gRPC를 통해 Hyperledger Fabric 원장(Ledger)에 접근하여 요청된 데이터를 읽거나 기록함
7. 결과 반환: Proxy는 원장에서 얻은 결과를 다시 Wrapper_TA에 전달하고, TA는 해당 값을 체인코드 실행에 반영함
8. 최종 결과 전송: 체인코드 실행이 완료되면 최종 결과를 Proxy가 수신하고, 이를 gRPC를 통해 Fabric 네트워크로 반환

결론 및 향후 연구 방향

결론

✓ i.MX8M-EVK 보드 실험에서 TEE 도입 시 성능 저하가 1.6~2.1배 수준임을 확인하였고, 이는 보안성을 위한 수용 가능한 비용임을 검증함

✓ WatZ의 Wasm과 TZ4Fabric의 Fabric연동 구조를 통합하여, TEE 기반 블록체인 아키텍처의 확장 가능성을 제시함

향후 연구 방향

✓ 현재는 Wasm 체인코드만 지원하지만, 장기적으로 Solidity(EVM) 등 다양한 언어 지원을 검토

✓ IoT와 같은 자원 제약 환경을 고려한 경량화 및 에너지 소비 분석도 필요