2020년도 기본연구 신규과제
연구계획서(연구내용)

과제명	국문	서비스 레벨 보안 강화를 위한 스마트 컨트랙트 변형 및 감시 기술
	영문	Transforming and monitoring techniques of smart contracts for service-level security

1. 연구의 목표 및 내용

블록체인 서비스의 개화 : 스마트 컨트랙트와 외부 모듈을 이용한 서비스로 발전

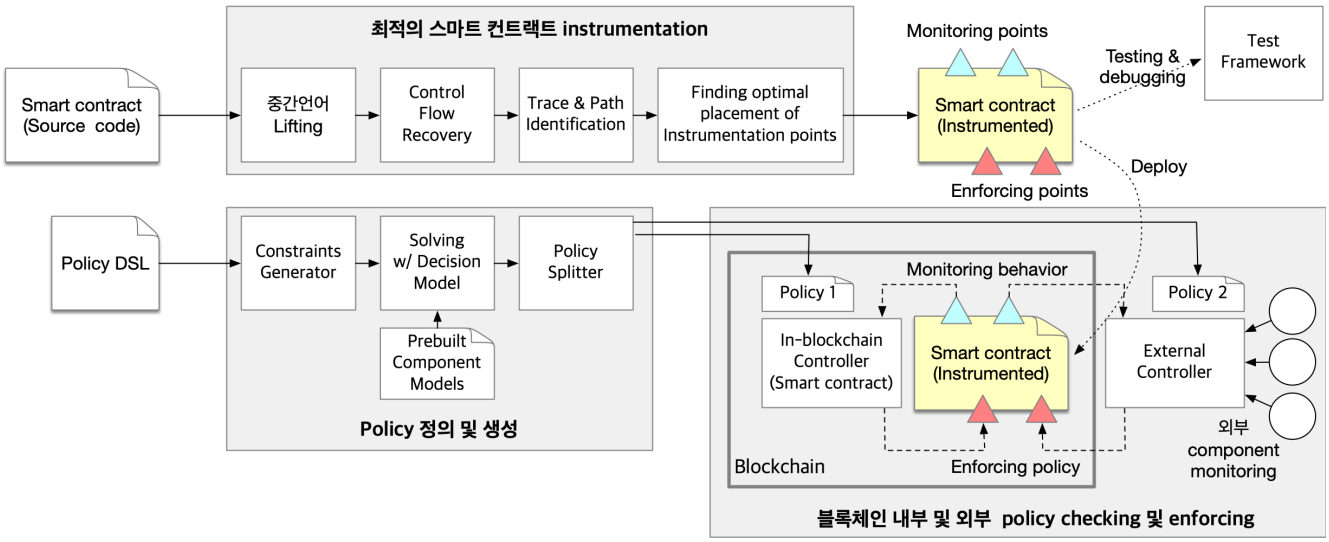
블록체인의 서비스는 Decentralized Application(DApp)라 불리는 블록체인 서비스들은 탈중앙화된 운영을 목표로 하고 있다. 일반적으로 하나의 블록체인 서비스는 실제 로직이 구현되어 있는 스마트 컨트랙트(smart contract)들을 핵심으로 외부의 사용자로의 인터페이스를 연결하여 완성된다. 한동안 대표적인 서비스가 없어 침체되었던 블록체인 서비스들은 최근 Compound[1], Uniswap[2], MakerDAO[3], Zerion[4]과 같은 분산형 금융서비스(Decentralized Finance, DeFi)를 중심으로 활력을 찾고 있다. 이에 따른 기술적 측면의 변화로, 각 블록체인 서비스는 단일 스마트 컨트랙트의 형태를 벗어나 타서비스의 스마트 컨트랙트 및 외부 정보와 연계하는 형태로 확장되고 있다.

블록체인 서비스 안전성 보장의 한계 : 서비스 레벨의 안전성 강화 방안 필요

한번 설치되고 나면 변경 불가능한 스마트 컨트랙트의 특성과 개입없이 운영되어야 하는 블록체인 서비스의 탈중앙성 요건 때문에 블록체인 서비스의 안정성에 대한 요구가 그 어느 때보다 높은 상황이다. 하지만 최근 bZx, IOTA, Falcrum의 해킹 사건[5,6,7]과 같이 DeFi 서비스의 공격은 기존의 스마트 컨트랙트 보안감사 체계로는 사고 예방이 어렵다는 한계점을 드러내고 있다. 이러한 불확실성에 대비하고자 대표 DeFi 서비스인 MakerDAO와 Compound (5000억원 이상 예치)[8]는 이미 감사를 마친 스마트 컨트랙트 내에도 단계마다 kill switch의 역할을 할 수 있는 코드를 포함시켜 안전을 도모하고 있다.

최종목표 : 서비스 레벨 안전성 강화를 위한 스마트 컨트랙트 변형 및 보안 모듈 기술 개발

서비스 단계에서 안전성을 보장 받기 위해서는 스마트 컨트랙트 코드의 보안감사만이 아닌 외부효과에 의해 운영 중에 발생할 수 있는 이상 상황에 대처할 수 있어야 한다. 특히 다른 서비스들과 많은 상호작용이 많은 DeFi와 같은 블록체인 서비스에서는 단순히 코드의 보안감사만이 아닌 서비스 운영 과정에서 이상동작을 탐지하고 제어할 수 있는 구조가 필요하다. 따라서 본 과제에서는 블록체인 서비스 레벨의 안전성 강화를 위하여 스마트 컨트랙트를 모니터링이 가능하도록 변형하고, 이를 기반으로 이상 상황에 대하여 서비스의 안전성을 보장할 수 있는 제어기술을 개발하고자 한다.



[그림 1] 서비스 레벨 안정화를 위한 스마트 컨트랙트 변형 및 보안 모듈

세부 요소 기술

(1) 효율적인 스마트 컨트랙트 모니터링 및 instrumentation 기술

서비스 개발의 고도화 과정에서 코드의 문제점을 파악하고 코드의 운영 정보(예를 들어 특정 기능의 이용 빈도)를 효과적으로 파악하기 위해서는 코드를 관찰하고 세부적으로 조절할 수 있는 instrumentation 기술이 필수적이다. 하지만 현재 스마트 컨트랙트를 고려한 instrumentation 기법의 개발과 연구는 거의 전무한 상황이다. 최근 SIF[9]란 시스템이 Solidity 언어로 작성된 스마트 컨트랙트에 대한 instrumentation 목적의 framework으로 처음 등장하였으나, 이용자에게도 전문적인 AST(Abstract Syntax Tree) 단계의 지식을 요구하고 있어서 사용성이 떨어진다. 또한 instrumentation의 아이디어를 기반으로 한 ContractLarva[10]도 이를 활용한 스마트 컨트랙트 보안 강화 방안을 모색하고 있지만, 오히려 instrumentation과정에서의 높은 overhead의 문제를 드러내고 있다.

본 과제에서는 효율적인 스마트 컨트랙트 instrumentation을 위하여 다음과 같은 접근 방법을 적용하고자 한다.

- **스마트 컨트랙트 특유의 overhead를 고려한 최적의 instrumentation** : 가장 많이 사용되는 EVM(Ethereum Virtual Machine)기반의 스마트 컨트랙트는 코드의 명령어 사용 방식에 따라서 다른 비용(e.g., gas)이 발생한다. 이러한 비용을 고려하여 최적의 위치에 최소의 코드를 주입하면서도 관찰성(observability)를 유지할 수 있도록 instrumentation을 위한 spot을 찾는다.

- **중간언어를 활용한 언어 선택적 instrumentation** : 스마트 컨트랙트 동작의 관찰은 함수의 호출 관계나 상태 변수 변화 등과 같이 프로그래밍 언어에 비종속적인 경우가 대부분이다. Slither[11]등의 정적분석 과정에서 많이 사용하는 중간언어 방식을 활용하여, 프로그래밍 언어 종속적인 패턴과 비종속적인 행동 패턴에 대한 선택적 instrumentation이 가능하게 한다.

(2) 블록체인 서비스 policy 정의 및 블록체인 내·외부 policy checking 및 enforcing 기술

DeFi 서비스에 대한 해킹 사건들에서 드러나듯이, 블록체인 서비스에 대한 안전성 보장은 단순히 코드의 취약점 점검을 벗어나 참여하는 component들의 전체적인 서비스 안전성을 점검해야 하는 단계로 접어들었다. 또한, 스마트 컨트랙트의 변경불가한 특성이 양날의 검처럼 작용하여, 작성과정에서 미지의 위험성을 미리 대비하여 대응하기 위한 조치 또한 필수적이다. 가장 현실적인 해결책으로써 현재 Compound, MakerDAO, Uniswap 등에서는 controller 또는 comptroller라 불리는 policy checking 및 enforcing 모듈을 블록체인 내에 함께 스마트 컨트랙트로 개발하여 두고, 스마트 컨트랙트의 동작 과정 중에 개입하여 policy 위배 여부를 판단하는 방법을 도입하고 있다. 하지만, 아직 이러한 안전장치는 도입 초기로써, 최근의 DeFi들에 대한 공격[5,6,7]에서도 서비스 참여자를 모두 포함하는 policy에 대한 고려가 부족하다는 점이 드러나 지속적인 연구와 개발이 요구되고 있다.

본 과제에서는 서비스의 policy를 정확하게 정의하고, 스마트 컨트랙트 instrumentation 기술을 이용하여 정의된 policy가 블록체인 내부 및 외부 모듈을 통해 자동으로 점검되고 강제되는 위한 보안 기술을 개발한다.

- **서비스 레벨의 policy 정의 및 점검 기술** : 블록체인 서비스의 policy는 스마트 컨트랙트 코드 레벨에서 지켜야 하는 property에서 시작하여, 서비스별 비즈니스 로직까지의 다양한 레벨에서 정의될 수 있다. 하지만, temporal logic[12]나 automata 방식[10]의 policy 정의는 복잡도가 높아 서비스 레벨의 policy에 바로 적용하기 쉽지 않다. 본 과제에서는 전통적으로 network 분야에서 DSL(Domain Specific Language)를 이용하여 충돌없는 policy 및 설정 정보를 생성하는 방법[13]을 착안하여, 블록체인 서비스 레벨의 policy 정의 방법과 블록체인 내·외부에 위치할 policy 점검 모듈을 위한 분배하는 자동화 기술을 개발하고자 한다.

- **블록체인 내부와 외부를 조합한 policy checking 및 enforcing 기술** : 블록체인 서비스를 위하여 정의된 policy를 바탕으로 서비스가 의도대로 동작하는지 점검하고 이상동작이 일어나지 않도록 제어한다. 예를 들어 Compound 서비스에서는 관리 스마트 컨트랙트인 comptroller에서 외부 가격 변화로 인해 대출조건을 만족시키지 못하는 사용자에 대한 동작 권한을 관리한다. 본 과제에서는 스마트 컨트랙트의 instrumentation 기법을 스마트 컨트랙트 행동 관찰을 위한 monitoring point와 동작 제어를 위한 enforcing point(또는 slot)를 생성하도록 확장하고, policy 점검 및 적용 모듈을 블록체인 내부(스마트 컨트랙트 레벨)와 외부(외부 정보 변화 적용)로 구성하여 각각에 맞는 policy 조건을 제공하여 실용적인 해결책을 찾는다.

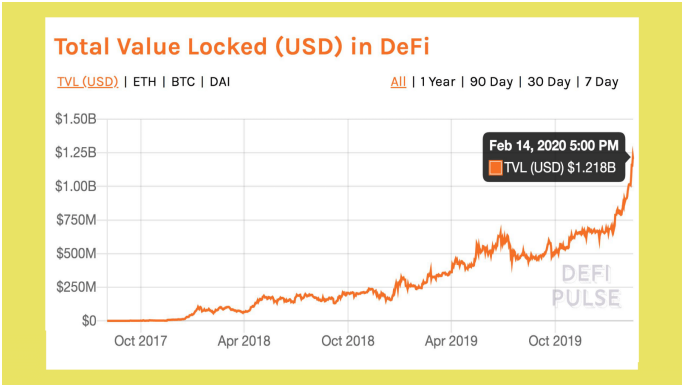
2. 연구의 필요성

블록체인 서비스 안전성 검증의 필요

블록체인의 도입은 기존의 체계에서와는 전혀 다른 접근방법을 가지는 서비스들을 가능하게 할 것으로 기대되고 있지만, 기술과 서비스에 대한 명확한 이해가 부족한 상황에서 안전하고 올바르게 동작하는 서비스를 판별하기 어려운 상황이다. 특히 사기성이 짙은 블록체인 프로젝트들이 많은 피해자들을 만들기도 하고, 나쁜 의도가 없더라도 생각지 못한 보안 취약점 때문에 경제적인 피해를 가져오기도 하고 있다. 특히 스마트 컨트랙트의 소스코드를 공개하고도 어떻게 사용자에게 불공정한 ICO 서비스를 만들 수 있는지를 겨뤘던 Underhanded Solidity Coding Contest[14]와 같은 대회에서도 알 수 있듯이, 코드의 외부환경까지 고려한 면밀한 검증 없이는 안정성을 확인하기 어렵다.

각자도생적인 안전성 확보를 요구하는 DeFi 서비스의 진입장벽

2020년 2월을 기준으로 총 1.4조원 가량의 자산을 보유할 정도로 발전하고 있는 DeFi 서비스들은 블록체인 서비스의 활성화라는 측면에서 중요한 의미를 가지고 있지만, 제도적 도움 없이 블록체인 시스템만으로 사용자가 납득할 수 있는 안전성을 보장해야 한다는 진입장벽을 가지고 있다. 즉, DeFi 서비스를 시도하는 업체마다 스마트 컨트랙트 코드에 대한 보안감사와 비즈니스 로직에 대한 검증, 코드와 비즈니스 로직과의 정합성, 이상 상태 발



[그림 2] DeFi 서비스의 자산 보유량

생에 따른 대응 체계 구축 등의 작업들을 직접 해결해야 하는 상황이다. 본 과제의 연구 결과를 통하여 반복적인 보안성 강화 방안을 체계화하고 동시에 서비스 개발의 진입장벽을 낮추고자 한다.

서비스 레벨의 보안 대응을 요구하는 블록체인 해킹의 진화

최근 집중되고 있는 블록체인 서비스이 공격은 대상이 되는 스마트 컨트랙트 코드에 직접 exploit payload를 보내는 방식에서 해당 스마트 컨트랙트가 참조하고 있는 다른 스마트 컨트랙트들의 이상 상태를 유도하고 이를 전파되게 함으로써 목적을 달성하는 형태로 진화하고 있다. 2020년 2월의 bZx에 대한 공격[5] 또한 서비스 자체의 방어체계를 갖추었음에도 자체 코드만을 고려하고 연결된 서비스의 변화(liquidity의 변화)를 고려하지 못하여 발생하게 되었다. 따라서 이러한 해킹의 진화에 대응하는 서비스 보안의 강화가 필요한 시점이다.

3. 연구자의 연구 수행역량

기존 Grant 과제 연구 성과

(1) **"보안 위험성 분석을 위한 자동화된 소프트웨어 역공학 시스템 개발"** (신진연구) : 소프트웨어의 정교화된 역공학 과정을 통하여, 바이너리 코드에 내재되어 있는 소프트웨어의 특성과 동작과정을 파악하는 역공학 시스템을 개발. 공동 연구를 통하여 USENIX Security 등의 우수 학회에 논문 발표.
(2) **"차세대 M2M 금융 시스템을 위한 보안 기술 개발"** (기본연구) : 소프트웨어 중심의 M2M 금융 모델엑서의 안전한 보안 시스템 구축 기술 개발. 기존의 금융 모델을 설명하기 위해 사용되었던 게임이론을 보안 시스템에 함께 적용하여 안전성 및 보안성을 높인 기법들을 개발하여 SCI 급 논문들을 발표.

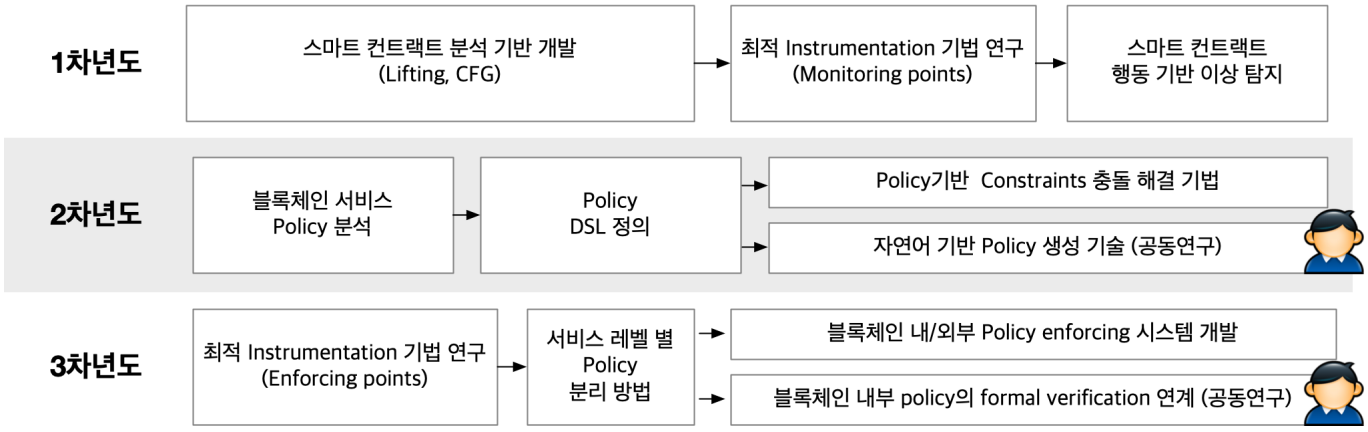
세부 연구 관련 수행역량

(1) 효율적인 스마트 컨트랙트 모니터링 및 instrumentation 기술
- 중간언어를 이용한 자동역공학 시스템 개발 과제 수행
- (안드로이드에서의) 프로그램 분석을 통한 프로토콜 복원 논문 및 특허
- 정밀도 높은 소프트웨어 분석 기법 관련 우수 논문 발표

(2) 블록체인 서비스 policy 정의 및 블록체인 내·외부 policy checking 및 enforcing 기술
- Deep learning을 이용한 스마트 컨트랙트 자동 분석 기술 논문 게재 및 특허 출원
- 스마트 컨트랙트 취약점 탐지 과제 수행 중
- Temporal logic의 model checking을 사용한 블록체인 탈중앙 서비스 모델 논문 게재

4. 연구의 추진전략 및 방법

실용성 높은 연구 성과를 얻기 위하여 각 요소기술을 MVP(Minimum Viable Product) 형태의 프로토타입을 구성하고 [그림 3]과 같이 연차별로 진행한다. 연구의 진행 과정 중에서 deep learning을 이용한 자연어 기반 policy 생성 기술과 formal verfication과의 연계 기술 들은 관련 분야 외부 전문가와의 공동 연구를 통해서 연구의 활용도를 높이고 이후 연구의 초석이 될 수 있도록 전략적으로 연구를 추진한다.



[그림 3] 단계별 연구 추진체계

5. 연구결과의 중요성

블록체인 서비스 레벨의 검증에 대한 요구 대응

역설적이게도 블록체인의 탈중앙성과 스마트 컨트랙트의 변경불가성은 블록체인 서비스가 런칭된 후에는 기민하게 대응하기 힘든 상황을 만들게 되었다. 특히 갈수록 고도화 되어 가고있는 블록체인 서비스를 효과적으로 분석하고 검증하기 위한 기술 개발은 아직 초기단계에 머물러 있다. 본 과제는 이러한 서비스 레벨의 안전성 확보를 위한 필수적인 기능을 제공할 수 있다.

스마트 컨트랙트의 자동화된 모니터링과 테스트 방법 제공

고도화된 서비스에 맞추어 기존의 컴퓨팅 환경에서 LLVM Clang이나 Clang LibTooling 등에서 제공하는 수준의 테스트 방법들이 스마트 컨트랙트에 필요하지만 아직 요원하다. 현재 가장 많이 사용되는 스마트 컨트랙트 개발 환경은 Truffle Suite[15]으로 개발자가 매번 meta정보들을 Javascript나 JSON으로 작성해야 하는 command line interface (CLI) 기반의 기초적인 환경을 벗어나지 못하고 있다. 하지만 Truffle만이 스마트 컨트랙트에 대한 테스트 환경(Solidity와 javascript의 테스트케이스 작성 지원)을 제공하고 있어, 불편함을 감수하고 많은 스마트 컨트랙트 개발자들이 사용하고 있는 상황이다. 본 과제의 연구 결과는 기존의 스마트 컨트랙트 테스트 환경을 더 체계화하고 자동화하여 더 나은 개발환경을 제공하는데 적용될 수 있다.

6. 연구기간 및 연구비 적정성

본 과제의 연구 결과는 policy의 점검 부분에서는 전통적인 정형 검증(formal verification) 기법, 그리고 policy 생성 부분에서는 deep learning의 자연어 처리 분야와의 연계를 통하여 결과의 활용도를 높일 수 있다. 이러한 목적에 맞는 시스템 구축과 전문가 활용 계획으로 예산을 구성한다.

7. 기타

해당 사항 없음

– 참고문헌(Reference)

[1] Compound, https://compound.finance/

[2] Uniswap Exchange Protocol, https://uniswap.io/

[3] Maker DAO, https://makerdao.com/en/

[4] Zerion, https:/zerion.io

[5] “DeFi lending protocol bZx exploited, ‘a portion of ETH lost’” https://www.theblockcrypto.com/linked/56134/defi-lending-protocol-bzx-exploited-a-portion-of-eth-lost

[6] “IOTA takes a nosedive following Trinity wallet hack”, https://cryptoslate.com/iota-takes-a-nosedive-following-trinity-wallet-hack/

[7] “Fulcrum had a $2.5m vulnerability over a month ago and still hasn’t told anyone,” https://medium.com/@1inch.exchange/yes-we-hacked-bzx-fulcrum-but-one-month-ago-3f7e5c437ee3

[8] DeFi Locked value historic overview, https://dapptotal.com/defi

[9] Peng, Chao, Sefa Akca, and Ajitha Rajan. “SIF: A Framework for Solidity Contract Instrumentation and Analysis.“ 2019 26th Asia-Pacific Software Engineering Conference (APSEC). IEEE, 2019.

[10] Azzopardi, Shaun, Joshua Ellul, and Gordon J. Pace. “Monitoring smart contracts: Contractlarva and open challenges beyond.“ International Conference on Runtime Verification. Springer, Cham, 2018.

[11] Slither, the Solidity source analyzer, https://github.com/crytic/slither

[12] Permenev, Anton, et al. “Verx: Safety verification of smart contracts.“ 2020 IEEE Symposium on Security and Privacy, SP. 2020

[13] Tian, Bingchuan, et al. “Safely and automatically updating in-network ACL configurations with intent language.“ Proceedings of the ACM Special Interest Group on Data Communication. 2019. 214-226.

[14] Underhanded Solidity Coding Contest, https://u.solidity.cc/

[15] Truffle Suite, https://www.trufflesuite.com/