블록체인 기반 안전 거래 플랫폼

이규민*, 김준현*, 성우상*, 이승훈*, 전준영*, 최병준*, 이병천**
*중부대학교(학생), **중부대학교(교수)

Blockchain based Safe Transaction Platform

Gyumin Lee*, Junhyeon Kim*, Woosang Seong*, Seunghoon Lee*,
Junyoung Jeon*, Byeongjun Choi*, Byoungcheon Lee**

*Joongbu University(Student),
**Joongbu University(Professor)

요 약

본 논문에서는 기존의 중고 거래 플랫폼의 사기 위험성과 같은 단점을 보완하기 위해 안전성을 향상시킨 블록체인 기반 안전 거래 플랫폼을 구현하였다. 구매자와 판매자 사이에 에스크로 기능을 두어 판매 대금을 스마트 컨트랙트에 예치하고, 구매자가 판매자의 상품을 실제로 받아본 뒤에 구매 확정을 하면 예치되어 있던 판매 대금이 판매자에게 전송되는 프로세스를 갖추고 있다. 이런 에스크로 기능을 통해 구매자가 상품을 확인하고 구매 확정을 하는 기능을 제공하여 거래 사기 등의 악용을 미리 방지할 수 있는 장점을 기대할 수 있다.

I. 서론

우리나라에서는 팬데믹 이후 중고 거래 플랫폼의 비대면 거래 규모가 매년 증가해왔다. 하지만 그만큼 비대면 중고 거래 사기와 사기 피해액 또한 기하급수적으로 증가하고 있다[1]. 기존 중고 거래 플랫폼에서 악성 유저들은 유명 신발이나 노트북 같은 값비싼 제품들을 판매하는 글을 게시하고, 먼저 선금을 받고 물품을 보내주지 않는 수법의 사기를 행해왔다. 즉, 판매 대금을 받았음에도 허위 매물을 보내주거나 아예 제품을 보내지 않는 등 약속과 다르게 진행함으로써 판매자가 부당한 이득을 추구할 가능성이 있다. 이러한 방식을 사용한 악성 유저들로 인해 기존의 중고 거래 플랫폼은 사기의 위험과 허위 매물 등과 같은 악용 사례들이 꾸준히 증가하는 경향을 보여왔다.

이런 사례들을 방지하고자 블록체인 기반 안전 거래 플랫폼을 개발했다. 중고 거래에 블록체인을 사용하면 거래 과정에 있어서 더욱 신뢰성을 높이고 무결성이 보장된 블록체인을 통해 데이터를 저장, 제공하고 스마트 컨트랙트를 통해 거래 기록을 저장한다[2]. 이 과정에서 앞서 언급한 악용 사례를 방지하기 위해 사전에 거래 대금을 결제해야만 상품을 보내주는 기존의 방식에서 벗어나, 판매자와 구매자 사이에서 컨트랙트 내에 토큰을 예치하는 에스크로 기능을 스마트 컨트랙트에 포함했다.

본 논문에서는 여러 블록체인 플랫폼 중 이더리움 네트워크 기반 스마트 컨트랙트에 에스크로 기능을 포함하여 믿을 수 있는 거래 환경을 제공하는 안전 거래 플랫폼을 제안한다. 본 논문의 구성은 다음과 같다. 2장에서는 서비스에 사용되는 관련 기술에 관해서 기술하고, 3장에서는 본 논문의 서비스에 대한 대략적인 흐름과 구상도를 제시한다. 4장에서는 논문의 플랫폼을 이용함으로써 얻을 수 있는 이점과 결론을 제시한다.

II. 관련 연구

2.1 블록체인

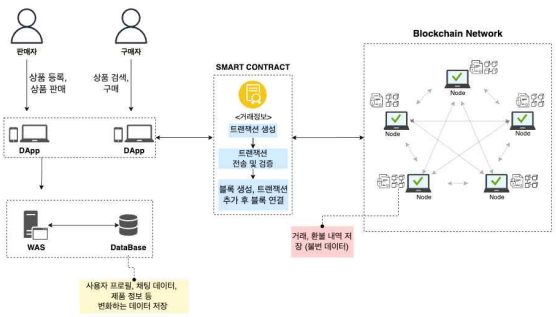
블록체인은 특정 데이터들이 여러 개의 트랜잭션의 형태로 블록이라고 하는 소규모 단위로 모여 P2P방식 기반으로 생성된 후 체인 형태로 연결되는 분산 데이터 저장 환경으로 해당 환경에 저장된 데이터는 참여하는 모든 노드에 분산 저장되며 누구라도 임의의 수정이 불가능에 가까우며 누구나 변경의 결과를 열람할 수 있다. 새로운 트랜잭션을 저장하고 블록을 연결하는데 블록체인은 각각의 합의 방식을 사용한다. 대표적으로 1세대 블록체인인 비트코인은 작업증명(PoW)을 합의 방식으로 사용하고 2세대 블록체인인 이더리움은 초기 이더리움 1.0일 때에는 작업증명(PoW)이었으나, 추후 이더리움 2.0으로 바뀌는 과정에서 합의 메커니즘이 작업증명에서 지분증명(PoS)으로 바뀌었다[3]. 그 외에도 블록체인의 종류와 공개 유형에 따라 다양한 합의 알고리즘이 존재한다.

2.2 이더리움, 스마트 컨트랙트

본 논문에서는 앞서 소개한 블록체인 중 이더리움 블록체인을 사용하여 안전 거래 플랫폼을 구현했다. 비트코인과 같은 1세대 블록체인에서는 거래 내역만 저장 가능하다는 한계점이 있었는데 이더리움 블록체인은 스마트 컨트랙트이라는 실행 가능한 프로그램 코드를 기록할 수 있다. 이런 스마트 컨트랙트를 통해서 커스텀 토큰과 분산 앱을 구현할 수 있다. 작성된 컨트랙트는 이더리움 블록에 포함되고 계약 실행을 위해서 이더리움 가상 머신(EVM)이 구동된다[4]. 이더리움 플랫폼상에서 커스텀 토큰이 원활하게 상호작용하고 컨트랙트 간 토큰이 전송될 때 호환성을 제공하기 위해 구현된 ERC 표준 프로토콜들이 있다. 본 논문에서 제시된 안전 거래 플랫폼에서는 ERC-20의 표준 토큰 규격을 통해 커스텀 토큰을 구현하여 사용하였다. 스마트 컨트랙트를 작성하는 언어로는 C++, Java, Python, Go 등으로 여러 프로그래밍 언어를 지원하는데 본 논문의 플랫폼에서는 솔리디티를 사용해서 스마트 계약을 구현했다.

III. 시스템 프로세스

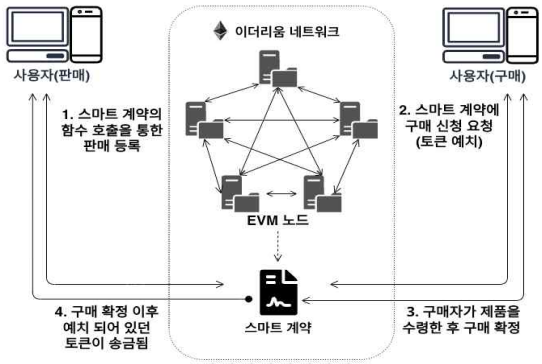
3.1 서비스 구상도



[그림 1. 서비스 구상도]

그림 1은 웹 애플리케이션의 서비스 구상도이다. 전통적인 DB와 블록체인에는 각각 다른 데이터가 저장된다. 거래 내역과 같은 필수적으로 유지해야 하는 데이터들은 블록체인에 기록되고 채팅 데이터, 제품 데이터, 사용자 프로필 등 가변적인 데이터는 MySQL DB에 저장된다. 그러므로 기능마다 요청 형식이 다르다. 구매(구매 및 구매 확정) 기능은 클라이언트에서 Web3.js와 Thirdweb[5] 프레임워크를 이용하여 스마트 컨트랙트의 함수들을 호출하여 실행하고 이외의 기능은 백엔드 서버에 API 요청을 보낸다.

3.2 에스크로 기능 관련 구상도



[그림 2. 에스크로 구상도]

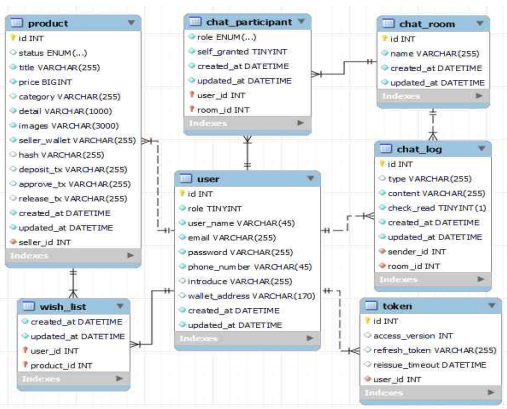
그림 2는 플랫폼의 에스크로 기능에 대한 구상도이다. 판매자가 제품을 판매 등록하면 제품과 제품 데이터에 대한 해시값이 생성되어 MySQL DB에 저장된다. 이때 생성되는 해시는

이더리움 플랫폼에서 많이 사용되는 Keccak-256 알고리즘을 사용했다. 구매자는 판매 등록된 상품에 대하여 구매 요청을 하고, 상품의 정보와 해시값을 인자로 하여 스마트 컨트랙트의 함수를 호출하게 되면 계약에서는 해시를 이용해 무결성 검사를 진행한 후 그림 3과 같이 계약에 정의해놓은 Escrow 구조체에 대해서 제품 ID 값으로 매핑해 데이터를 생성하고 상품 가격만큼의 토큰을 계약에 예치하게 된다. 이후 판매자는 구매자에게 상품을 전달하게 되고, 구매자는 상품을 받아보고 기존에 전달받은 상품의 상태와 같은지 판단 후, 구매 확정 단계를 거치게 된다. 구매 확정을 한다면 생성했던 Escrow 데이터의 isApprove 값을 true로 갱신하여 예치되어 있던 토큰은 판매자가 수령할 수 있는 상태로 바뀌고, 판매자가 토큰 수령 버튼을 누르면 판매자의 지갑에 상품 가격만큼의 토큰이 입금된다.

```
struct EscrowData {
    uint64 prodId;
    uint256 amount;
    uint32 buyerId;
    address buyer;
    uint32 sellerId;
    address seller;
    bool isApprove;
}
mapping(uint64 => EscrowData) escrows;
```

[그림 3. 스마트 컨트랙트에 정의된 구조체]

3.3 데이터베이스

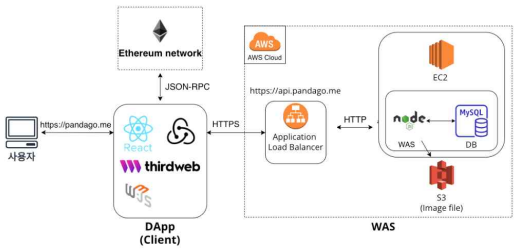


[그림 4. 데이터베이스 ERD]

그림 3은 플랫폼에서 비교적 사이즈가 크거나 가변적인 데이터를 저장하는 MySQL DB에 대한 ERD이다.

3.4 아키텍처 구성

그림 4는 전체적인 서비스 구상도에서 DApp과 WAS에 대한 자세한 아키텍처 구상도이다. DApp에서 온 요청 중 블록체인에 데이터를 기록하거나 데이터를 가져오는 요청은 그림 1에서의 과정처럼 스마트 계약에 Thirdweb과 Web3.js 를 통해 요청을 보내고 가변 데이터에 대한 API 요청은 WAS로 보내게 된다. 클라이언트에서 온 API 요청은 express 서버를 배포한 AWS의 ALB(Application Load Balancer)로 보내지게 된다. 이때 요청은 HTTPS 통신으로 보내지고 ALB는 들어온 트래픽을 EC2 인스턴스로 보낸다. express 서버는 MySQL 서버와 데이터를 주고받고 AWS S3에 이미지를 저장한다. 부가적으로 채팅 관련 통신은 소켓을 통해 통신한다. 블록체인에 저장된 데이터와 MySQL DB에 저장된 데이터 간 동기화를 위해 express 서버에 이벤트 리스너를 두고 블록체인에 특정 트랜잭션이나 이벤트가 발생할 시 DB에 저장할 수 있게 하고 시간마다 동기화 로직을 실행시킬 수 있게 스케줄러를 설정해놓는 방법으로 블록체인과 DB의 데이터를 동기화 시킨다.



[그림 5. 아키텍처 구상도]

3.4 특징 및 기존 플랫폼과의 비교 분석

쿠팡 등 기존의 상거래 플랫폼에서는 거래의 안전을 위해 서버가 거래 중개자로서 에스크로 서비스를 제공하는 사례가 있다. 판매자와 구매자는 서버의 에스크로 기능을 신뢰하여 거래를 진행하고, 서버는 분쟁이 발생하는 경우 대금 지급을 중지할 수 있도록 하여 거래의 안정성을 보장하는 것이다. 이러한 방식은 중앙화된

시스템에 의존하는 전통적인 방식이다.

본 논문에서는 스마트 컨트랙트 자체에 에스크로 기능을 추가하여 탈중앙화된 에스크로 서비스를 구현하였다. 기존에도 이더리움을 에스크로 서비스에 활용하는 연구가 진행되었다. 여러 연구 중 논문에서 구현한 플랫폼에서는 EIP(이더리움 개선 제안)-5528에서 제안한 인터페이스를 참고하였다[6]. EIP-5528에서는 보편적으로 구현될 수 있는 예제를 제시했지만, 플랫폼에서는 중고 거래 플랫폼의 특성에 맞게 에스크로 관련 데이터에 제품의 ID 등을 추가하여 구현했다.

이를 통해 판매자와 구매자는 중앙화된 서버에 의존하지 않고 거래의 안정성을 확보할 수 있게 되었다. 에스크로 트랜잭션을 발생시킬 때 추가 가스비(Gas price)를 지출해야 한다는 점이 있지만, 거래에서 사기를 당했을 때의 불필요한 지출을 생각했을 때 위험성을 줄여주는 것만으로도 큰 가치가 있다고 판단된다.

IV. 결론

본 논문에서는 중고 거래 시장의 문제를 해결하기 위해 이더리움 기반 스마트 계약을 도입한 플랫폼을 제안하였다. 중고 거래는 판매자와 구매자의 신뢰성이 형성되기 어려운 구조이며, 직거래를 통한 거래가 이루어지거나 혹은 구매자가 사기 위험을 감수하고 택배 거래를 통해 거래가 이루어지고 있다. 실제로 중고 거래 사기 피해 금액이 900억 원에 육박했다. 2014년 202억에서 4배 이상 피해 금액 규모가 늘었다. 이렇게 사기 등의 문제를 해결하기 위해 이더리움을 활용해 안전 거래 플랫폼을 구현했다.

본 연구에서 제안한 블록체인 기술에 기반을 둔 중고 거래 플랫폼은 구매자가 제품을 구매하면 곧바로 토큰이 판매자에게 송금되지 않고 스마트 계약 자체에 예치되었다가 구매자가 제품을 확인하여 구매 확정을 하면 판매자에게로 토큰이 송금되는 에스크로 기능을 필수적으로 넣어 사기 가능성을 낮췄다. 구매자가 물품을 확인하고 구매 확정을 하지 않아 판매자에게로 토큰이 송금되지 않을 수 있는 문제점을 해결하기 위해 주어진 시간이 지나면 자동으로 구매 확정되는 아이디어를 활용했다. 판매 이력 및 구매 이력에 대한 정보가 수정 불가능하다는 점은 판매자와 구매자의 높은 신뢰성을 보장할 수 있다.

[참고문헌]

[1] 장승주, "[중고거래사기] 커지는 중고거래플랫폼, 늘어나는 사기범들", 아주경제, 2022, 10. 02.

[2] 최환석, 박민영, 송유빈, 이우섭, "블록체인 기반의 신뢰적 거래 플랫폼 개발", 디지털콘텐츠학회논문지, vol. 22, no. 8, pp. 1153-1163, 2021

[3] E. Kapengut and B. Mizrach, An Event Study of the Ethereum Transition to Proof-of-Stake, Commodities, vol. 2, no. 2, pp. 96-110, 2023.

[4] 안병태, "블록체인 기술을 이용한 안전 거래 시스템 구축(사례:중고자동차)", 디지털융복합연구, vol. 18, no. 4, pp. 237-242, 2020.

[5] Thirdweb, https://thirdweb.com/

[6] ERC-5528: Refundable Fungible Token, https://eips.ethereum.org/EIPS/eip-5528