스마트 계약을 이용한 블록체인 기반 공항 체크인 모델

김혜빈*, 박지선*, 김종규*, 신상욱**
*부경대학교 대학원 정보보호학협동과정
**부경대학교 IT융합응용공학과
e-mail : khbin1346@pukyong.ac.kr

Blockchain-Based Airport Check-In Model Using Smart Contract

Hye-Bin Kim*, Ji-Sun Park*, Jong-Kyu Kim*, Sang Uk Shin**
*Interdisciplinary Program of Information Security, Graduated School, Pukyong National University
**Dept. of IT Convergence and Application Eng., Pukyong National University

요 약

현재까지 공항에서의 체크인 시스템은 기본적으로 서버-클라이언트(Server-Client)방식으로 구동되어 져왔다. 서버-클라이언트 방식에서 서버는 단일실패지점이기 때문에 천재지변이나 공격자에 의해 서버 마비와 같은 문제가 생기면 체크인 시스템 전체가 마비될 수 있다. 뿐만 아니라 실제 해커에 의해 항공사 중앙 서버가 공격을 받게 되면, 서버가 관리하는 항공사 데이터들의 무결성이 훼손될 수도 있다. 따라서 단일 실패지점에 대한 위협을 제거함과 동시에 저장된 데이터들의 무결성을 유지하는 방법을 논의할 필요가 있다. 그 중 하나로 블록체인 기술을 이용한 탈중앙화 시스템(Decentralized System) 모델을 적용하는 방법이 있다. 본 논문에서는 스마트 계약을 이용한 블록체인 기반 공항 체크인 시스템에 대한 모델을 제안한다.

1. 서론

최근 들어 천재지변이나 전력 공급부족과 같은 문제로 공항의 전산망이 마비되거나 DDoS(Distributed Denial of Service) 공격과 같은 해킹 등에 의해 항공사 서버가 마비되어 탑승 수속 하려는 승객들이 불편을 겪는다는 뉴스가 종종 보도되곤 한다. 사람들은 비행기 탑승 전 직접 공항의 체크인 카운터에 가거나 셀프 체크인 키오스크를 이용하여 본인이 갖고 있는 항공권을 이용하여 탑승권(Boarding Pass)을 발급받는데[1], 이 때 위에서 서술한 천재지변과 같은 이유로 시스템이 제대로 작동하지 못한다면, 수기로 체크인을 하거나, 그것도 불가능하다면 시스템이 복구될 때까지 비행기에 탑승하지 못하고, 공항에 발이 묶여있을 수밖에 없다.

이렇게 시스템 전체가 한꺼번에 마비가 되는 이유는 공항의 체크인 시스템이 기본적으로 서버-클라이언트(Server-Client)방식으로 구동되어져왔기 때문이다[2].

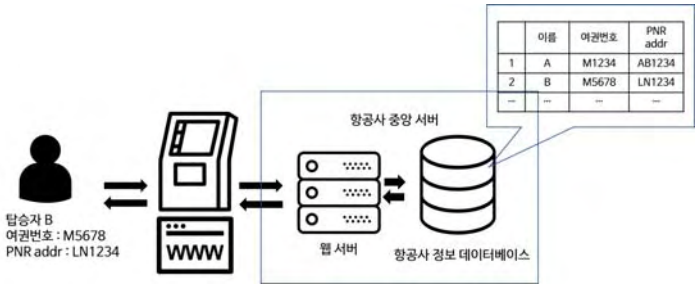
기본적으로 서버-클라이언트 방식의 서버는 단일 실패지점(Single Point of Failure)의 문제점을 갖는다. 단일 실패지점이란 시스템 구성 요소 중에서 동작하지 않으면 전체 시스템이 중단되는 요소를 의미한다[3]. 다시 말해, 서버가 다운되면 시스템 자체에 불능이 올 수 있다.

뿐만 아니라 실제 해커에 의해 서버가 공격을 받게 되면, 서버가 관리하던 데이터베이스에 저장되어 있는 데이터들의 무결성이 훼손될 수도 있다. 이런 문제가 발생한다면, 탑승 예약자들의 스케줄이 지켜지지 않을뿐더러 항공사의 신뢰도 하락까지 이어질 수 있다. 따라서 단일 실패지점을 없애며, 데이터의 무결성을 훼손시키지 않을 방법을 고려해볼 필요가 있는데 그 방법 중 하나로 탈중앙화 시스템을 고려해볼 수 있다. 본 논문에서는 대표적인 탈중앙화 시스템인 블록체인을 기반으로 하는 스마트 계약을 이용한 체크인 시스템에 대한 모델을 제안한다.

2. 관련연구

2.1. 기존 공항 체크인 시스템

이 절에서는 기존 시스템에서 어떻게 탑승 예약자가 체크인을 하는지에 대하여 간단히 살펴본다. <그림 1>은 기존 체크인 시스템에 대한 간략한 구조이다.



< 그림 1 > 기존 체크인 시스템

원하는 항공편을 예약 하게 되면 예약자에게는 전자 항공권(E-Ticket)이 주어짐과 동시에 항공사 정보 데이터베이스에 예약자의 이름, 여정, 연락처 정보로 구성된 PNR(Passenger Name Record)이 저장이 된다[4]. 항공권은 그에 대한 예약번호(PNR Address)로 식별 할 수 있다.

체크인을 위해 탑승 예약자는 체크인 카운터나 셀프 체크인 키오스크를 이용하여 공항의 체크인에 필요한 정보를 입력한다. 이 정보에는 예약번호 또는 예약자의 여권번호와 이름이 해당된다. 정보를 입력받은 체크인 웹 페이지가 서버에 체크인 요청을 하면, 중앙 서버는 웹 페이지로부터 실제로 입력받은 전자 항공권에 있는 정보와 항공사 정보 데이터베이스에 저장된 PNR 정보가 일치하는지 여부를 체크한 다음, 옳다고 판단이 되면 좌석지정을 할 수 있는 단계로 넘어간다. 좌석 예약도 확정 되면 항공사 정보 데이터베이스에 확정 정보가 기록되고, 체크인 결과는 웹 페이지를 통해 보여준다. 이로써 승객은 탑승권을 수령할 수 있다.

2.2. 이더리움 스마트 계약(Ethereum Smart Contract)

블록체인(Blockchain)은 2008년 Satoshi Nakamoto에 의해 고안된 암호화폐 비트코인의 근간이 된 기술이다. 이는 수백 개의 트랜잭션들을 블록으로 모아 저장한 탈중앙화 데이터베이스이다[5]. 다시 말해, 중앙 관리 서버 없이 P2P 네트워크의 노드들에 의해 합의 알고리즘(Consensus Algorithm)이란 규칙을 기반으로 하여 전체 시스템이 유지·관리되는 기술이다[6]. 각 블록은 이전 블록의 해시값과 타임스탬프를 포함하여 블록 내용이 변경되면 그 후에 연결된 블록들의 내용들도 변경된다. 따라서 블록 내용의 위·변조가 어렵다. 그리고 트랜잭션을 포함한 블록이 채굴을 통해 한번 체인에 추가가 되면, 블록의 내용을 바꿀 수 없다. 따라서 일반적인 분산 데이터베이스에 비해 상대적으로 높은 안전성과 불변성, 투명성을 갖는다. 또한 모든 노드가 동일한 데이터베이스를 관리하므로 단일 실패지점에 대한 위험성이 낮다.

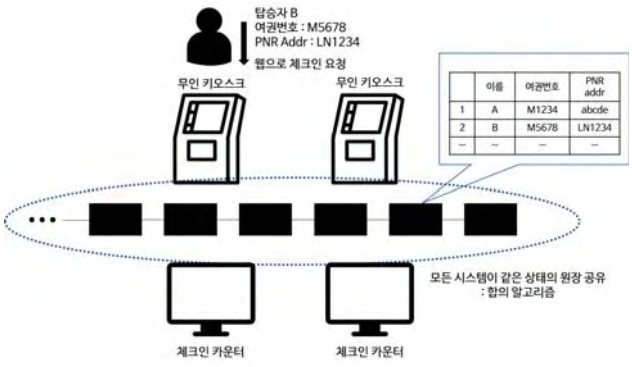
이더리움 스마트 계약(Ethereum Smart Contract)이란 블록체인을 기반으로 하는 금융거래, 부동산 계약, 공증 등 다양한 형태의 계약을 체결하고 이행하는 것을 의미한다. 이는 2013년 Vitalik Buterin이 블록체인 기술을 이용하여 다양한 스마트 계약을 처리할 수 있도록 기능을 확장하면서 널리 알려지게 되었다[7]. 개발자가 직접 계약 조건과 내용을 코딩할 수 있기 때문에 확장성을 갖는다. 이더리움에서는 튜링 완전한 언어인 솔리디티(Solidity)를 이용하여 이를 구축할 수 있다. 블록체인 기반 분산 어플리케이션인 DApp(Decentralized Application)과 연동하여 더 다양한 시스템의 스마트 계약이 구현 가능하다[8].

3. 스마트 계약을 이용한 체크인 시스템

이 절에서는 스마트 계약을 이용한 체크인 시스템의 동작 프로세스에 대하여 논하고자 한다. 아래의 <그림 2>는 그에 대한 간략한 구조를 나타내고 있다.

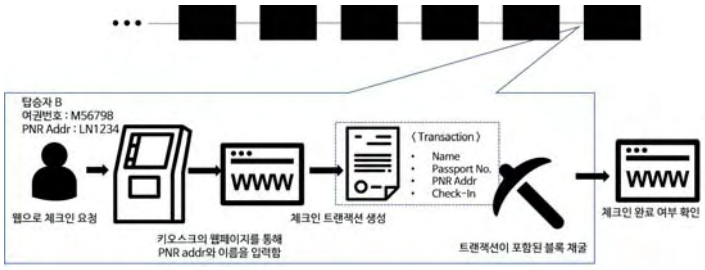
제안 시스템의 기반이 될 블록체인 네트워크에 참여하는 노드들에는 항공사 별로 설치된 체크인 카운터의 PC들과 셀프 체크인 키오스크가 해당된다.

키오스크와 카운터 PC에 구동되는 웹 페이지는 본 시스템의 DApp이다. 노드들은 각자 블록체인 데이터베이스를 보유하고 있으며 새로운 블록 추가 시 합의 알고리즘에 의해 채택이 되어야한다. 이로 인해 노드들은 동일한 내용의 데이터베이스를 갖고 있을 수 있다.



< 그림 2 > 블록체인 기반 체크인 모델

기존에 웹 서버가 모든 요청에 응답을 하던 방식이 아닌, 이 요청과 응답을 웹 페이지와 항공사의 데이터가 담긴 블록체인의 사이에서 스마트 계약을 이용하여 처리한다. 동작 방식을 알아보기 위해 본 논문에서는 솔리디티를 이용한 유연성과 확장성을 가진 이더리움 스마트 계약으로 구현을 한다. 대략적인 동작방식은 아래의 <그림 3>과 같다.



< 그림 3 > 체크인 요청 과정

스마트 계약의 내용은 데이터 구조체와 항공권 체크인에 관한 다양한 함수들로 이루어진다. 우선 탑승 예약자가 갖고 있는 전자 항공권의 정보가 담긴 PNR은 그것의 고유한 예약 번호, 예약자의 영문이름, 여권번호, 여정의 내용, 그리고 체크인의 완료 여부를 필드로 갖는 구조체로 구현된다. 또한 계약 내에서 사용될 수 있는 함수로는 체크인 기능을 하는 함수, 체크인이 완료가 되었는지 여부를 웹 페이지에 출력하는 기능을 갖는 함수, 체크인 후 좌석 지정을 하는 함수가 있는데 본 논문에서는 체크인 기능을 하는 함수에 대해서 우선적으로 논의한다.

탑승 예약자가 본인의 전자항공권 예약번호나, 영문이름과 여권번호를 웹 페이지에 입력하고 '확인' 버튼을 누른다. 확인 이벤트를 받은 계약 내의 함수는 사용자가 웹에 입력한 정보와 블록체인에 저장된 PNR 데이터와의 비교를 통해 일치여부를 확인해야 한다.

위의 함수를 통해 데이터간의 일치가 확인되면, 읽어온 데이터를 입력 값으로 갖는 트랜잭션을 생성한다. 그 후에 채굴자들이 트랜잭션이 포함된 블록을 채굴하면, 트랜잭션의 결과가 시스템에 반영이 된다. 체크인 여부는 기타 추가적인 기능을 수행한 후 웹 페이지에서 확인할 수 있다.

4. 분석

기존의 체크인 시스템은 항공사 중앙 서버와 해당 항공사에 부여된 공항 체크인 카운터가 서버-클라이언트 방식으로 동작했기 때문에, 서버가 단일실패지점을 되는 단점을 갖고 있었다. 따라서 해킹과 같은 공격이나 천재지변으로 항공사 중앙 서버가 다운되면, 접속 자체가 불가능하여 체크인이 되지 않는 문제가 있었다.

본 논문에서 제시한 체크인 시스템은 탈중앙화 시스템인 블록체인을 적용하여 항공사 정보 데이터베이스를 중앙 서버만이 관리하는 방식이 아니라, 네트워크에 참여하는 노드들이 동일한 데이터들을 관리할 수 있다. 이 블록체인 네트워크에 참여하는 일부 노드가 여러 이유로 시스템이 마비되더라도, 재부팅 후 정상 작동하는 노드들로부터 데이터들을 다운로드받거나, 최악의 경우에도 일부 노드만으로 시스템을 유지할 수 있다. 즉 단일 실패지점이 없어 공항의 모든 PC가 동시에 공격받지 않는 이상 전체 시스템 마비는 불가능하다.

그리고 모든 체크인 트랜잭션은 암호학적 해시함수를 통해 블록체인에 저장되어 있기 때문에, 블록체인의 특성에 따라 트랜잭션 위·변조의 위험성이 낮아 기록된 모든 승객의 데이터에 대한 무결성을 유지할 수 있다[9]. 따라서 보안성 측면에서 여러 공격에 대하여 기존의 시스템보다 더 안전하다고 판단할 수 있다.

본 논문에서 제안된 모델의 구현 가능성을 보여주기 위한 방법으로 이더리움 스마트 계약이 사용되었다. 이더리움의 평균 블록 채굴시간은 평균 12초인데, 실제 체크인 시스템은 실시간성을 가져야 함으로 12초라는 시간도 승객이 카운터 앞에서 기다리기에는 굉장히 긴 시간일 수 있다. 그리고 블록 채굴 후 분기(Fork)가 발생할 수 있음에 따라[10], 체크인 과정에서 서로 다른 데이터간의 충돌이 생길 수 있다.

또한 체크인 데이터는 공개 범위가 제한되어야 할 개인의 민감한 정보가 포함되어 있다. 이는 노드들로 하여금 참여 제한을 두지 않는 일반적인 공개 블록체인(Public Blockchain)의 특징인 투명성과는 부합하지 않는다. 따라서 참여 노드들 간에 프라이빗 네트워크(Private Network)를 구축하여 체크인 데이터에 대한 기밀성을 가질 필요가 있다. 이는 미리 정해진 노드들이 네트워크의 접근과 합의 과정에 참여할 수 있는 사설 블록체인(Private Blockchain)으로 구현될 수 있다. 이를 구현할 수 있는 플랫폼은 하이퍼레저 패브릭(Hyperledger Fabric)이다[11].

따라서 현실적인 구현 방식을 논의한다면 우선 사용자들의 프라이버시 보호를 위해 네트워크에 참여하려는 노드들에게 자격 제한을 둬야 할 것이다. 또한 합의 알고리즘으로는 빠르게 합의가 이루어지고 분기가 발생하지 않는 BFT(Byzantine Fault Tolerance) 방식이 적합할 수 있다.

5. 결론

본 논문에서는 스마트 계약을 이용한 블록체인 기반 공항 체크인 시스템의 동작 가능성에 대해 논하였다. 이 시스템은 단일 실패지점을 가지고 있던 기존의 공항 체크인 시스템과는 다르게 네트워크에 참여하는 노드들이 같은 내용의 블록체인 데이터들을 갖고 있기 때문에, 시스템의 노드 일부가 마비가 되더라도 다른 노드들로부터 빠르게 업데이트함으로써 결과적으로 전체 시스템이 마비되지 않게 한다는 점과 데이터들이 블록체인에 있기 때문에 데이터의 위·변조에도 취약하지 않다는 장점이 있었다. 따라서 시스템 마비 시에 기존 시스템과는 달리 탑승 예약자들은 물론 항공사의 피해를 최소화시킬 수 있다.

앞에서 언급된 바와 같이, 향후 본 시스템은 실제 체크인 시스템에서 고려해야하는 실시간성을 지키는 방식으로 확장될 수 있다. 이는 합의 속도를 고려하여 BFT 기반의 대표적인 합의 알고리즘인 PBFT(Practical Byzantine Fault Tolerance)를 사용하며[12], 네트워크에 참여하는 노드들이 미리 정해진 사설 블록체인을 지원하는 하이퍼레저 패브릭을 채택하여 구현될 수 있을 것이다.

Acknowledgement

본 연구는 한국전자통신연구원 연구운영비지원사업의 일환으로 수행되었음. [18ZH1200, 데이터 안심사회를 위한 트러스트 데이터 커넥톰 핵심 원천 기술 개발]

참고문헌

[1] Airport Check-in, Retrieved from URL: https://kr.koreanair.com/korea/ko/traveling/airport-check-in.html

[2] Mehovic, Farid. "System for propagating, retrieving and using transaction processing facility airline computerized reservation system data on a relational database processing platform." U.S. Patent No. 6,122,642. 19 Sep. 2000.

[3] Wikipedia, Single Point of Failure, Retrieved from URL : https://en.wikipedia.org/wiki/Single_point_of_failure

[4] Wikipedia, Passenger Name Record, Retrieved from

URL:
https://en.wikipedia.org/wiki/Passenger_name_record
[5] Romano, Diego, and Giovanni Schmid. "Beyond Bitcoin: A Critical Look at Blockchain-Based Systems." Cryptography 1.2 (2017): 15.
[6] Sankar, Lakshmi Siva, M. Sindhu, and M. Sethumadhavan. "Survey of consensus protocols on blockchain applications." Advanced Computing and Communication Systems (ICACCS), 2017 4th International Conference on. IEEE, 2017.
[7] Wikipedia, Smart Contract, Retrieved from URL : https://en.wikipedia.org/wiki/Smart_contract
[8] Buterin, Vitalik. "A Next-Generation Smart Contract and Decentralized Application Platform, Ethereum White Paper." 2014.
[9] Nakamoto, Satoshi. "Bitcoin: A peer-to-peer electronic cash system." 2008.
[10] 임종철, 유현경, 곽지영, 김선미, "블록체인과 합의 알고리즘.", 전자통신동향분석, Vol. 33, No.1, pp. 45-56, 2018.
[11] Androulaki, Elli, et al. "Hyperledger fabric: a distributed operating system for permissioned blockchains." Proceedings of the Thirteenth EuroSys Conference. ACM, 2018.
[12] Mingxiao, Du, et al. "A review on consensus algorithm of blockchain." Systems, Man, and Cybernetics (SMC), 2017 IEEE International Conference on. IEEE, 2017.