블록체인을 활용한 전자 문서 관리 시스템에 관한 연구

이아름*, 신용태**
*숭실대학교 컴퓨터학과
**숭실대학교 컴퓨터학부
e-mail:coo0121001@gmail.com

A Study On Electronic Document Management System Using Block Chain

Ah-Reum Lee , Yong-Tae Shin
Dept of Computer Science and Engineering, Soong-Sil University

요 약

21세기 산업 전반에 걸쳐 IT기술의 접목은 해당 산업의 부가가치와 편의성을 증가시키고 있다. 몇 가지 예로 전자 문서 관리 시스템, 디지털 자산 관리 시스템은 사용자에게 편리함과 편의성을 더해줬지만 이와 함께 정보의 위·변조, 유출과 같은 위험요소 역시 야기되었다. 그 해결 방안으로 블록체인 네트워크를 이용해 문서를 관리하는 시스템을 소개한다. 본 논문의 제안 시스템은 문서에 대한 모든 접근을 기록하여 추적을 가능하게 함으로써 문서 이동 경로의 투명성을 보장하는 기능을 제시한다. 또한 기존 시스템에서 소요되던 유지 비용을 줄이고 데이터의 무결성을 유지하여 중요 데이터의 유출 위험을 감소시키는 데 기여한다.

1. 서론

기존 전자 문서의 관리 현황을 보면 별도의 보안체계를 구축하여 보안 업계가 중앙이 된 시스템에서 관리하는 형식으로 열람, 반출 시 별도의 인증 수단을 거쳐 위험 요소들을 배재하는 식으로 운영 되고 있다. 이러한 관리 시스템의 예로는 문서 암호화, 데이터베이스 이중화 등이 있으며 이를 통해 데이터 위·변조, 유출을 방지한다. 하지만 보안 업계를 중심으로 한 중앙 집중형 관리 시스템은 복잡한 백업과 복구, 보안 활동의 유지 비용을 지속적으로 유발함으로써 편의적, 비용적 측면에서 계속적인 사용에 부담감이 있다. 그렇기 때문에 본 논문은 외부의 간섭 없이 보안을 향상시키며 문서를 관리할 수 있는 기법을 제안한다. 블록체인을 활용한 전자문서 관리 시스템은 외부 업체가 중앙이 되지 않더라도 이해 당사자 간의 참여로 문서의 무결성을 유지할 수 있다. 또한 새로운 네트워크를 구축하여 지속적인 유지 비용을 필요로 했던 기존의 보안 업체의 역할을 없애 비용을 절감할 수 있다. 뿐만 아니라 문서의 무결성을 유지하고 문서 이동 경로를 추적하여 투명성을 보장함으로써 관리 시스템을 운영하는 것에 있어 큰 장점을 가진다.

본 논문의 구성은 다음과 같다. 2장에서는 대표적인 블록체인 플랫폼의 종류와 기술적 특징들에 대해 소개하며 블록체인 네트워크를 응용할 수 있는 서비스를 설명한다. 3장에서는 본 논문의 제안하는 시스템에서 사용되는 프라이빗 블록체인, 데이터베이스의 설계 방법에 대해 설명하고 4장에서 결론을 맺는다.

2. 관련 연구

2.1 블록체인 플랫폼

본고의 2장에서는 대표적인 블록체인 플랫폼인 Ethereum과 Hyperledger 프로젝트의 Fabric에 대해 살펴본다.

Ethereum은 2015년 비탈릭 부테린에 의해 분산 어플리케이션 제작을 위한 대체 프로토콜을 만드는 것을 목적으로 개발된 것이다. 대규모 분산 어플리케이션에 유용할 것이라 생각되는 여러 종류의 제작기법을 제공하며, 빠른 개발 시간, 규모가 작은 어플리케이션을 위한 보안, 다른 어플리케이션과의 효율적인 상호작용이 중요한 상황에 특히 주안점을 두고 있다. 이더리움은 튜링 완전 언어(Solidity)를 내장하고 있으며 이 언어를 사용해 누구든지 스마트 컨트랙트, 분산 어플리케이션을 작성하고 소유권에 대한 임의의 규칙, 트랜잭션 형식, 상태변환 함수 등을 생성할 수 있다[1].

Hyperledger는 리눅스 재단에서 기업용 블록체인 개발을 위해 만든 오픈 소스 형태의 프로젝트로서 프라이빗 블록체인 형태로 개발되었다. 이더리움, 비트코인 등 누구나 참여할 수 있는 퍼블릭 블록체인과 달리 MSP라는 인증 관리 시스템에 등록된 사용자만이 하이퍼레저 패브릭 블록체인에 참여할 수 있다. 하이퍼레저의 패브릭 참여자들은 비즈니스 목적에 알맞은 형태로 블록체인 플랫폼을 구축하는 것을 목표로 개발하고 있으며 프라이버시와 기밀성, 작업 구간별 병렬 처리, 체인코드, 모듈화된 디자인을 특징으로 한다[2].

2.2 합의 알고리즘

<표 1> 블록체인 플랫폼 비교[3]

Characteristic	Ethereum	Hyperledger Fabric
Description of platform	-Generic Blockchain platform	-Modular Blockchain platform
Governance	-Ethereum developers	-Linux Foundation
Mode of operation	-Permissionless, public or private	-Permissioned, private
Consensus	-Mining based on proof-of-work(POW) -proof-of-stake(POS) (expected) -Ledger level	-Broad understanding of consensus that allows multiple approach -Transaction level
Smart Contracts	-Smart contract code (e.g, Solidity)	-Chain code (e.g, Go, Java...)
Currency	-Ether -Tokens via smart contract	-None -Currency and tokens via chaincode

2.2.1 이더리움 합의 알고리즘

이더리움 합의 알고리즘은 탈중앙화된 시스템에서 네트워크 참여자들의 신뢰성을 보장해 줄 수 있는 역할을 한다. 대표적으로는 현재 사용하고 있는 POW(작업증명 방식)와 앞으로 사용 예정인 POS(지분증명 방식)가 있다. POW는 컴퓨터 연산을 통해 블록 생성 및 검증에 기여하는 방법으로 누구나 합의 프로세스에 참여할 수 있도록 허용해줌으로써 합의결정권에 대한 정치적 문제를 해결할 수 있을 뿐만 아니라 Sybil Attack과 같은 해킹 공격에도 방어해줄 수 있는 메커니즘을 제공한다[1]. POS는 참여자의 코인 지분이 블록생성 및 검증에 참여하여 보상을 받는 것으로 블록 완성 기능성은 그 참여자의 관련 암호 화폐 보유량과 보유기간에 비례하여 증가한다[4].

2.2.2 하이퍼레저 패브릭 합의 알고리즘

하이퍼레저 패브릭의 합의는 이더리움과 달리 트랜잭션 생성부터 최신 블록이 peer에 저장되기까지의 모든 과정이다[2]. 하이퍼레저 패브릭 합의 알고리즘에 적용되는 Fault Tolerance는 어떤 장애가 발생해도 서비스가 중단되지 않도록 장애를 허용하는 기법을 말한다. 대표적으로 현재 사용되고 있는 Kafka 오더링 서비스의 기반인 CFT(Crash Fault Tolerance)와 개발 중인 PBFT 알고리즘의 기반인 BFT(Byzantine Fault Tolerance)로 나눌 수 있다. 먼저 CFT 기반 합의 알고리즘은 내부의 공격은 거의 없을 것으로 예상하고 특정 노드의 오류가 생기면 나머지 노드로 시스템을 동작하게 하는 충돌 장애 허용 기법이다. BFT는 모든 노드의 합의를 거쳐 데이터를 전달하는 동기적인 특성 때문에 특정 노드에서 잘못된 데이터를 전달하는 경우에도 최종적으로는 올바른 데이터가 전달될 수밖에 없는 기법이다[4].

3.블록체인을 활용한 전자 문서 관리 시스템

본 장에서는 블록체인을 활용하여 전자 문서를 관리하는 시스템에 대하여 제안한다. 제안하는 전자 문서 관리 시스템은 직접적인 관계가 있지 않은 참여자에게는 거래 내용을 공개되는 것을 방지하기 위해 권한을 부여 받은 사용자만이 참여할 수 있는 프라이빗 블록체인으로 구성한다.

3.1 노드 구성

본 논문의 블록체인 네트워크를 구성하는 노드는 크게 두 가지로 참여자 노드, 관리자 노드로 이루어진다.

①참여자 노드

참여자 노드는 기업으로부터 부여받은 권한으로 문서 저장 및 열람 활동을 하는 주체이다.

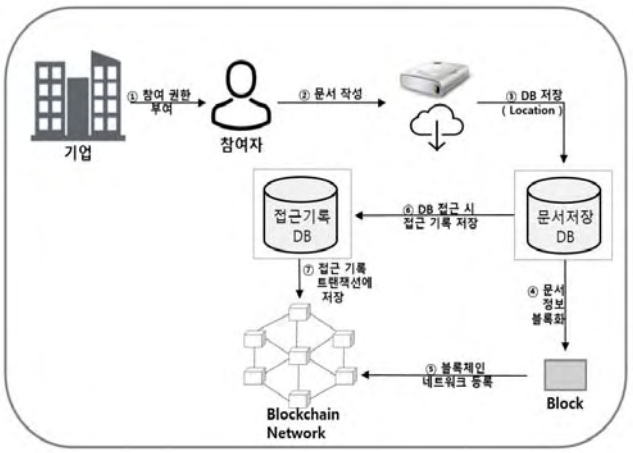
②관리자 노드

관리자 노드는 기본적으로 참여자 노드와 동일한 역할을 하며 추가적으로 문서의 이동 경로를 확인하는 역할을 한다.

3.2 전자 문서 관리 시스템 구조

본 논문에서 제안하는 시스템의 프라이빗 블록체인에서는 전자 문서 관리를 원하는 기업의 직접 참여가 이루어지며 권한을 부여하는 역할을 함으로써 신뢰 가능한 노드로 구성된 네트워크를 형성한다. 권한이 있는 참여자는 문서 저장 및 접근 활동을 자유롭게 할 수 있으며 참여자의 이러한 활동에는 체인 코드 및 합의 알고리즘의 실행이 동반된다. 이는 문서의 모든 이동경로 파악을 가능하게 하기 위한 것으로 추적을 통한 보안성 확보를 목적으로 한다.

위의 (그림 1)은 전자 문서 관리 시스템은 본 논문에서 제안하는 시스템의 구성을 보여주는 것으로 블록체인 네트워크에 참여하는 과정까지를 간략히 설명한다.

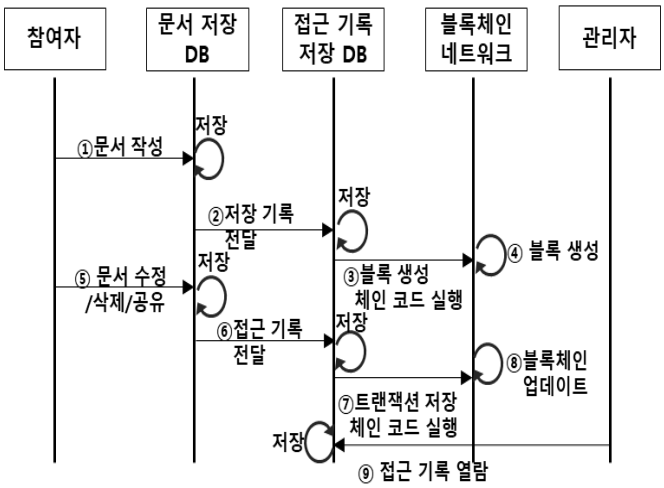


(그림 1) 전자 문서 관리 시스템 구성도

①기업은 사원번호, 이름, 부서, 직급, 주민등록번호가 인증된 사원에게 참여자의 권한을 부여한다.
②참여자로 지정된 사원은 문서를 작성하고 저장을 요청한다.
③문서저장 데이터베이스에 저장이 실행되면 문서번호가 key값, 문서의 local위치가 value값으로 들어간다.
④문서 저장 데이터베이스에 저장이 완료되면 일부 문서의 정보가 블록의 헤더 파일로 저장되며 블록을 생성한다.
⑤생성된 블록은 블록체인 네트워크에 다른 블록과 체인으로 연결되어 블록체인 네트워크를 형성한다.
⑥참여자가 문서 저장 데이터베이스에 있는 문서에 접근 시, 모든 접근자의 기록은 접근 기록 데이터 베이스에 저장된다.
⑦접근 기록 데이터베이스가 업데이트 되는 것이 확인되면 접근되어진 문서에 해당하는 블록의 트랜잭션으로 접근 기록이 저장된다.

3. 프라이빗 블록체인

3.3.1 프라이빗 블록체인 네트워크



(그림 2) 블록체인 네트워크 업데이트 및 등록

(그림 2)는 참여자의 문서 작성으로 블록 생성, 체인코드 실행, 블록체인 네트워크가 업데이트 되는 시기까지의 흐름을 나타낸다.

①참여자의 문서 작성

②해당 데이터베이스에 저장된 문서의 작성자 및 문서 정보를 접근 기록 데이터베이스에 저장

③참여자의 문서 저장 행위를 감지하여 체인 코드 실행

④실행된 체인코드의 내용에 따라 문서의 정보를 헤더파일로 한 블록 생성

⑤권한을 가진 참여자는 데이터베이스의 문서를 수정·삭제·공유 가능

⑥접근한 참여자의 정보 및 기록을 데이터베이스에 저장

⑦접근 기록 데이터베이스의 변경 내용을 감지하여 체인코드 실행

⑧해당 문서 블록의 트랜잭션으로 접근 기록 저장

⑨관리자 권한이 있을 시 접근 기록 데이터베이스 열람 가능

3.3.2 체인코드

본 논문의 제안하는 시스템에서 체인코드 발생의 경우는 두 가지이다. 참여자의 문서 작성에 의해 블록이 생성되는 과정, 참여자의 문서 접근으로 인해 트랜잭션이 저장되는 과정이다. 이 과정에서 체인코드가 발생하여 합의 알고리즘이 실행되는 경우, 미리 설정된 보증 정책에 지정된 peer의 허가를 받아야 블록 생성 및 트랜잭션 저장이 성립될 수 있다. 만약 보증 정책이 충족되지 못한 채로 peer에게 전달된다면 블록이 생성되지 않거나 트랜잭션이 저장되지 않는다. 본 논문에서 제안하는 시스템의 보증 정책 설계는 각 부서를 peer라고 가정했을 때, 기준점=100이라고 가정한다. 프로젝트에 속한 부서의 참여율로 각 peer에 가중치를 부여하고 각 peer의 허가 의사인 인증서를 받으면 가중치만큼의 값이 더해진다고 한다. 이 때, 이 값의 결과가 기준점을 넘으면 peer의 허가를 받았다고 인정된다.

①블록 생성

블록 생성 체인코드는 참여자가 작성한 문서를 최초로 문서 저장 데이터베이스에 저장할 경우 블록 생성을 위해 발생한다. 문서 저장 과정에서 데이터베이스에 저장된 [문서 번호, 문서 저장 시간]은 체인코드의 실행을 통해 블록체인 네트워크로 전달된다. 이후 합의 알고리즘의 과정을 거쳐 위에서 설명한 보증정책에 부합하여 peer의 인증을 받게 되면 새로운 블록의 헤더 정보로 들어가 새로운 블록으로 생성된다.

②트랜잭션 저장

트랜잭션 저장 체인코드는 참여자가 문서를 저장할 때, 데이터베이스에 접근하여 [수정, 삭제, 공유] 행위가 감지될 때 트랜잭션 저장을 위해 발생한다. 접근 행위 감지 시, 해당 데이터베이스와 연동된 접근 기록 데이터베이스에 문서 번호, 접근자의 이름, 사원 번호, 해당 행위가 기록된다. 이 값이 체인코드의 실행을 통해 블록체인 네트워크에 전달되면 합의 알고리즘을 거쳐 위에서 설명한 보증정책에 부합하여 peer의 인증을 받게 되면 전달된 정보에 해당하는 문서 블록의

트랜잭션으로 저장된다.

3.4 데이터베이스

본 논문에서 제안하는 시스템에서는 문서 저장, 접근 기록 저장을 목적으로 하는 2개의 데이터베이스를 구성한다. 문 서는 원본이 저장되는 별도의 저장 장치 혹은 시스템이 있으며 문서 저장 데이터베이스는 문서 번호, 별도의 장치 혹은 시스템에서의 문서 위치를 저장한다.

참여자는 문서 저장 데이터베이스에 접근하여 원하는 문서의 위치를 찾아 열람할 수 있는데 참여자의 접근 기록 데이터베이스가 업데이트된다. 접근 기록 데이터베이스는 서버를 기반으로 동작하며 문서 저장 데이터베이스로의 접근에 대한 정보, 접근 기록 데이터베이스로의 접근에 대한 정보를 저장한다. 이렇게 저장된 문서 저장 데이터베이스의 정보는 체인코드를 통해 블록의 생성으로 이루어지고 접근 기록 데이터베이스의 정보는 트랜잭션 저장으로 이루어진다.

3.4.1 문서 저장 데이터베이스 테이블

<표 2> 문서 저장 DB Table 예시

Document Number	Location
Hash(No.1)	Hash("D:₩A_Team_Document ₩A.hwp₩")
Hash(No.2)	Hash("E:₩B_Team_Document ₩B.txt₩")
Hash(No.3)	Hash("F:₩C_Team_Document ₩C.pdf₩")

참여자 권한을 가진 이의 전자 문서는 다음과 같은 형식의 테이블로 문서 저장 데이터베이스에 저장된다. 각각의 Document Number가 key값이 되며 모든 문서의 Location은 <표 2>와 같은 형태의 Value값으로 저장된다.

3.4.2 접근 기록 데이터베이스 테이블

참여자가 전자 문서를 최초로 저장할 때, 문서 저장 데이터베이스에 접근하여 전자 문서를 공유·수정·삭제하는 행위를 할 때마다 접근 기록 데이터베이스에 <표 3>과 같은 형태로 행위자의 이름, 접근하는 문서 번호, 접근하는 날짜 및 시간, [생성, 공유, 수정, 삭제] 행위의 여부가 기록된다.

<표 3> 접근 기록 DB Table 예시

Name	Document Number	Date	Action	Coworker
A	Hash(No.3)	2019-01-10-09:00	Share	B
C	Hash(No.9)	2019-01-18-16:30	Modify	NULL
E	Hash(No.10)	2019-01-19-11:10	Create	NULL
E	Hash(No.10)	2019-01-19-11:11	Share	C
D	Hash(No.11)	2019-01-20-13:00	Delete	NULL

4. 결론

본 논문은 문서 사용 이력을 관리하기 위해 블록체인을 활용된 전자문서 관리 시스템을 제안하였다. 본 논문의 제안된 시스템에서는 블록체인을 활용하여 문서 정보의 무결성 및 신뢰성을 보장하고 문서 접근 이력의 모든 절차를 기록함으로써 관리를 효율적으로 할 수 있을 것으로 판단된다. 향후, Hyperledger Fabric 환경에서 구현 및 테스트를 진행하여 실제 사용 가능 여부에 대해 연구할 예정이다.

Acknowledgement

본 연구는 과학기술정보통신부 및 정보통신기술진흥센터의 SW중심대학사업의 연구결과로 수행되었음(2018-0-00209-001)

참고문헌

[1] Vitalik Buterin, "A Next-Generation Smart Contract and Decentralized Application Platform"

[2] 윤대근, "하이퍼레저 패브릭으로 배우는 블록체인", ETRI 블록체인기술연구센터,11.2018.

[3] Martin Valenta, Philipp Sandne, "Comparison of Ethereum, Hyperledger Fabric and Corda", Frankfurt School, 06.2017.

[4] "Community Magazine of College of Engineering Seoul National University", Summer, 2018, No.109

[5] Andreas Wittig, "Amazon Web Services in Action", 1st Edition, 10.2015