Journal of KIIT. Vol. 22, No. 6, pp. 121-131, Jun. 30, 2024. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2024.22.6.121

모바일 운전면허증 발급을 위한 스마트 컨트랙트 설계 및 평가

이상윤*, 김범수**, 박재영***

Design and Evaluation of Smart Contracts for the Issuance of Mobile Driver's License

Sangyun Lee*, Beomsoo Kim**, and Jaeyoung Park***

요 약

신원증명 기술의 발전에 따라 분산신원증명 기반의 모바일 운전면허증이 최근에 출시되었다. 본 연구는 이더리움 기반의 스마트 컨트랙트를 활용한 모바일 운전면허증 발급 시스템을 제안한다. 모바일 운전면허증 발급 과정에서 스마트 컨트랙트의 적절한 구현 및 운영 방식을 살펴보기 위해, 우선 Solidity 프로그래밍 언어를 사용하여 이름, 생년월일 등의 신원 속성을 블록체인에 저장하는 다양한 모델의 스마트 컨트랙트를 설계한다. 다음으로, 이더리움 테스트넷 및 배포 환경에서의 가스 사용량과 실행 시간을 측정하여 제안된 모델들의 성능을 비교 및 분석한다. 이 연구는 모바일 운전면허증 발급 시스템의 효율성 향상과 개인정보 보호 강화 방안을 제안함으로써, 디지털 신원증명 분야에 기여한다.

Abstract

With advancements in identity technology, mobile driver's licenses based on decentralized identity(DID) have recently been introduced. This study proposes a system for issuing mobile driver's licenses utilizing Ethereum-based smart contracts. To examine appropriate implementation and operational methods for smart contracts during the mobile driver's license issuance process, the study first designs various models of smart contracts using the Solidity programming language to store identity attributes such as names and birthdates on the blockchain. Subsequently, it measures gas consumption and execution times within the Ethereum testnet and deployment environments to compare and analyze the performance of the proposed models. By suggesting improvements for the efficiency of the mobile driver's license issuance system and enhancing privacy, this research contributes to the field of digital identity.

Keywords

decentralized identifier, DID, Ethereum, self-sovereign decentralized Identity, SSI, smart contract, mobile driver's license

* 한국거래소 수석
- ORCID: http://orcid.org/0009-0009-1704-613X
** 연세대학교 정보대학원 교수
- ORCID: http://orcid.org/0000-0003-4821-8271
*** 농협금융지주 금융연구소 책임연구원(교신저자)
- ORCID: http://orcid.org/0000-0002-1608-6641

· Received: Apr. 03, 2024, Revised: Jun. 13, 2024, Accepted: Jun. 16, 2024

· Corresponding Author: Jaeyoung Park
Finance Research Institute, Nonghyup Financial Group, Korea
Tel.: +82-2-3782-7384, Email: inyourface33@gmail.com

Ⅰ. 서 론

전통적인 신원 관리의 모델에서는 정부, 금융기관과 같은 중앙 집중식 기관이 개인정보의 저장, 관리, 검증을 담당한다. 이 모델의 주된 문제는 데이터 보안과 개인정보의 오용이다. 즉 이러한 모델에서 사용자는 자신의 데이터에 대한 직접적인 통제를 거의 하지 못하며, 정보가 어떻게 처리되는지와 누가 접근할 수 있는지에 대한 정보를 충분히 가지지 못한다.

현재의 온라인 서비스는 일반적으로 개별 신원증명 방식으로 사용자를 식별한다. 즉 사용자는 사용하는 서비스마다 별도의 아이디와 비밀번호를 생성해야 하며, 서비스 이용 시 해당 서비스에 맞는 아이디와 비밀번호를 입력함으로써, 본인 확인 절차를 완료한다. 이러한 방식은 사용자가 사용하는 서비스의 수가 증가함에 따라 아이디와 비밀번호 관리에 대한 부담을 가중시키는 구조적 문제를 가지고 있다. 또한 각각의 서비스 제공기관이 사용자의 개인정보를 개별적으로 관리하기 때문에 어느 한 기관에서 해킹 사고가 발생하면 사용자의 개인정보가 대규모로 유출될 수 있다.

이러한 문제를 해결할 수 있는 방안으로 자기주권 신원증명(SSI, Self-Sovereign decentralized Identity)이 제안되고 있다[1][2]. 자기주권 신원증명은 스스로 권한을 가진 신원을 말한다. 즉 중앙 기관이나 중개자에 의존하지 않고 사용자가 자신의 방식으로 신원 데이터를 관리하고 공유할 수 있는 방법을 제공한다. 자기주권 신원증명은 탈중앙화 구조를 기반으로 하며, 사용자에게 완전한 신원 주권, 분산, 그리고 사용자 제어를 부여하여 데이터 관리를 가능하게 한다. 사용자는 블록체인의 공개 키에 연결된 자체 디지털 신원(Digital identity)을 생성하고 관리할 수 있으며, 이를 통해 중앙 집중식 중개자 없이도 안전하고 개인적인 인증 및 트랜잭션 권한을 부여할 수 있다.

분산신원증명(DID, Decentralized Identity)은 블록체인을 기반으로 한 자기주권 신원증명 체계로, 블록체인에서 사용하고 있는 암호기술을 기반으로 믿을 수 있는 탈중앙형 ID 저장소를 구현한다[3]. 분산신원증명은 전통적인 서버-클라이언트 형태의 중앙집중형 신원 관리와는 다르게 사용자가 자신의 신원정보를 직접 관리하고 어떤 정보를 어디에 제출할지를 결정한다. 이러한 방식은 지속성, 휴대성, 개인정보 보호, 피어(Peer) 기반과 같은 특징을 가진다[4].

자기주권 신원증명의 주요 이점 중 하나는 개인에게 보다 많은 권한을 제공한다는 것이다. 즉 사용자는 자신의 신원 정보를 어떤 기관이나 개인과 공유할지, 그리고 얼마나 오랜 기간 동안 공유할지 스스로 선택할 수 있다. 이와 같은 사용자 중심의 보다 안전하고 투명한 신원 관리 방식은 기존 중앙집중식 신원 관리 체계가 지닌 신원 도용 및 개인정보 침해와 같은 위험을 상당 부분 완화할 수 있다.

최근 자기주권 신원증명 기술을 적용한 사례가 등장하고 있다. 특히 모바일 신분증이 주목받고 있다. 이러한 신원증명 기술 발전에 발맞춰, [5]는 모바일 운전면허증 도입에 있어서 분산신원 기반의 자기주권 신원증명 모델 적용 방안을 연구하였다. 하지만, 그들의 연구는 개념적 분석에 그쳤다. 또한 최근 자기주권 신원증명 기술을 활용한 스마트 컨트랙트 연구가 진행되고 있다. 스마트 컨트랙트는 신뢰할 수 있는 중개자 없이 당사자 간의 계약을 촉진, 실행 및 시행하기 위해 블록체인 플랫폼에서 실행되는 실행 코드이다. [6]은 데이터의 무결성을 보장하고 안전하게 공개키를 교환할 수 있는 스마트 컨트랙트 구현을 위한 자격증명 시스템을 제안하였다.

기존 연구는 다양한 분야에서의 스마트 컨트랙트 활용 가능성을 말해주고 있지만, 스마트 컨트랙트가 구현되는 최적의 방법은 사용 사례와 요구 사항 및 구현 방법에 따라 달라질 수 있다. 이에 본 연구에서는 기존 연구에서 다루지 않았던 모바일 운전면허증 발급 사례를 중심으로, 분산신원증명에 중점을 둔 이더리움 기반 스마트 컨트랙트 구현 방안을 제시한다. 구체적으로 본 연구는 다양한 모델의 스마트 컨트랙트 설계와 성능 평가를 통해 최적의 성능을 유지하면서 개인정보 보호를 강화할 수 있는 방안을 모색한다.

II. 관련 연구

2.1 자기주권 신원증명

자기주권 신원증명은 사용자가 자신의 신원 정보를 직접 관리하고 통제할 수 있는 디지털 신원 모델의 혁신적 발전이다. 크리스토퍼 알렌이 2016년에 제안한 이 개념은 기존의 중앙화된 디지털 신원 모델의 한계를 극복하고자 한다. 기존 모델은 서비스 제공자나 중앙 기관 같은 제3자에게 과도하게 의존하며, 이로 인해 공격자의 타깃이 되거나 대규모 개인정보 유출의 위험에 노출될 수 있다. 또한 사용자의 신원 주체성과 권한 보호가 충분히 이루어지지 않는 문제점을 가지고 있다. 크리스토퍼 알렌은 디지털 신원 모델의 진화를 중앙집중형 신원(Centralized Identity), 연합형 신원(Federated Identity), 사용자 중심 신원(User-Centric Identity), 그리고 자기주권 신원(Self-Sovereign Identity) 등 네 가지 단계로 구분한다.

마지막 단계인 자기주권 신원은 사용자가 자신의 디지털 신원에 대한 주권과 통제 권한을 완전히 가짐으로써, 중앙기관이나 제3자의 의존성 없이 신원을 관리하고 공유할 수 있어야 한다는 철학을 갖고 있다. 즉 사용자의 주권을 강조하여 신원 구조를 혁신하려는 목표를 가지고 있다. 이 개념은 디지털 신원 관리와 보안 분야에서 중요한 역할을 하며, 블록체인과 분산원장 기술과의 결합을 통해 구현되고 있다.

블록체인 기반 자기주권 신원증명의 구성요소는 DID(Decentralized Identifier), DID Document, VC(Verifiable Credential), VP(Verifiable Presentation)이다. VC는 검증가능한 자격증명을 말하며, VP는 검능가능한 제공 ID 데이터 집합이다. 자기주권 신원증명 모델의 주요 참여자는 발행인(Issuer), 사용자(Holder), 검증인(Verifier), DID 관련 정보를 저장하는 신뢰 분산저장소인 블록체인이다.

DID 및 DID Document는 식별자 및 인증 수단으로 사용되며, 발행인이 발행하는 VC는 보관용 ID로 사용된다. VC는 사용자가 발행기관으로부터 발급받은 신분증, 졸업증명서, 재직·재학 증명서, 입원·퇴원 확인서 등과 같은 신원증명을 의미한다. 사용자는 VC 내에서 필요한 신원정보만 추출한 후에 VP로 가공해서 서비스 제공자에게 제출한다. VP는 제출용 ID로 사용되는데, 사용자가 VC를 발급받은 후 VP로 가공하여 검증기관에 제출한다. 검증기관은 사용자로부터 VP를 수신하여 VP의 진위를 검증한다. VP를 수신한 검증기관은 VP의 제출자 DID를 통해 올바른 사용자로부터 VP를 수신했는지 알 수 있고, VP에 포함된 VC 속성을 가진 사람이 해당 사용자가 맞는지는 VC에 명시된 Subject DID를 통해 알 수 있다. 또한 VP에 포함된 VC 속성값이 어떤 발행기관에서 발행되었는지는 발행기관의 DID를 통해 알 수 있다. 검증기관은 VP에 존재하는 VC의 Subject DID가 진짜 사용자의 것인지는 DID Auth를 통해 알 수 있고, VP 및 VC에 포함된 전자서명 값도 DID Document를 통해 검증할 수 있다.

자기주권 신원증명 모델을 활용한 모바일 운전면허증 발급 서비스 플랫폼에서 핵심이 되는 부분은 블록체인으로 구현되는 DID Resolver와 DID Registry이다. DID Resolver는 자기주권 신원증명 모델에서 사용자의 DID를 해석하고, 해당 DID에 연결된 정보를 검색하는 서비스로 분산된 신원 관리 시스템에서 사용자의 신원을 확인하는 데 중요한 역할을 수행한다. 즉 DID Resolver는 DID 문서의 위치를 찾고 해당 문서를 검색하여 신원 정보를 확인할 수 있게 한다.

2.2 SSIBAC

[7]은 자기주권신원 기반의 접근제어인 SSIBAC (Self-Sovereign Identity Based Access Control)을 제안하였다. 신원주체가 갖고 있는 속성에 기반하여 접근제어를 수행하는 속성 기반 접근제어(Attribute based access control)이나 신원주체의 역할에 기반하여 접근제어를 수행하는 역할 기반 접근제어(Role based access control)을 사용하는 것이 기존 기업에서 접근제어를 수행하는 일반적인 방법이다. 하지만, 기존 구조에서는 접근제어를 수행하는 기관이 신원주체의 정보를 모두 자신이 보관한 상태에서 접근제어를 수행하는 중앙집중형신원의 구조에 가까웠고, [7]은 이것을 해결하기 위해서 기존의 접근제어 구조에 자기주권신원을 적용하여 신원주체의 자기주권성을 보장하고자 하였다.

SSIBAC은 기본적으로 검증 가능한 자격증명의 구조를 그대로 사용하되, 검증자가 별도의 접근 통제 엔진(Access control engine)을 두어 자격증명에서 필요한 속성들을 추출한 뒤 접근제어를 수행하여 사용자에게 접근권한을 부여하고 있다는 점이 다르다. 또한 SSIBAC도 기본적인 접근제어는 속성 기반 접근제어나 역할 기반 접근제어를 사용하고 있으나, 접근제어를 위한 신원정보를 중앙기관에 의존하는 것이 아니라 분산원장인 검증가능한 크리덴셜 레지스트리(Verifiable credential registry)에 의존한다는 점이 다르다.

SSIBAC는 접근제어를 수행할 때 중앙기관에 신원정보가 저장되는 것을 방지하기 위해 자기주권신원을 활용한 접근제어를 제안함으로써, 신원정보의 중앙화를 해결하였다. 신원주체가 직접 자신의 정보를 관리하게 됨으로써, 기관 의존성을 제거하고 탈중앙화 식별자나 검증 가능한 자격증명의 활용을 통해 자기주권성을 강화하였다.

하지만, SSIBAC은 한계점을 가지고 있다. 자기주권신원 10가지 원칙을 보면, 신원정보는 휴대성(Portable)을 가져야하고, W3C의 검증 가능한 자격증명 표준에 따라 자격증명은 타인에게 전송하는 것이 가능해야 한다. 또한 타인에게 전송된 자격증명이라 할지라도 신원주체의 통제가 가능해야 하고, 자격증명의 활용에는 반드시 신원주체의 동의가 있어야만 한다. SSIBAC은 이러한 조건을 만족시키지 못한다.

자기주권성을 실현하기 위해서는 실제로 사용자가 자신의 신원 정보와 데이터를 통제하고 관리할 수 있어야 한다. 또한, 데이터의 이동과 활용에 대한 통제 및 동의도 중요한 요소이다. 이러한 분산 원장과 상호 작용하여 자격증명과 접근 권한을 관리하는 데 사용되는 것이 바로 스마트 컨트랙트이다.

2.3 스마트 컨트랙트

스마트 컨트랙트는 신뢰할 수 있는 제3자의 개입 없이 신뢰할 수 없는 당사자 간의 계약을 촉진, 실행 및 시행하기 위해 블록체인 위에서 실행되는 실행 코드이다. 전통적인 계약과 비교하여 스마트 컨트랙트는 중앙기관의 감독 없이 자동화된 거래를 제공함으로써, 사용자가 계약 및 신뢰 관계를 성문화할 수 있도록 한다.

스마트 컨트랙트는 사전 정의된 기능 덕분에 정보를 저장하고, 입력을 처리하고, 출력을 작성한다. 예를 들어, 스마트 컨트랙트는 스마트 컨트랙트 생성을 가능하게 하는 생성자 기능을 정의할 수 있다. 블록체인에서 새로운 스마트 컨트랙트를 호스팅하는 것은 전송자가 생성자 기능을 호출하여 스마트 컨트랙트 소유자가 되는 트랜잭션을 통해 활성화된다.

스마트 컨트랙트는 이더리움 등 다양한 블록체인 플랫폼에서 개발 및 배포될 수 있다. 여러 플랫폼은 계약 프로그래밍 언어, 계약 코드 실행 및 보안 수준을 포함하여 스마트 컨트랙트 개발을 위한 고유한 기능을 제공한다. 이더리움은 스마트 컨트랙트 개발을 위한 최초의 블록체인 플랫폼이다. 프로그래밍 언어인 Solidity는 스마트 컨트랙트를 작성하는 데 사용되며, 계약 코드는 이더리움 가상 머신(EVM, Ethereum Virtual Machine) 바이트 코드로 컴파일되어 실행을 위해 블록체인에 배포된다. 이더리움은 현재 스마트 컨트랙트를 위한 가장 인기 있는 개발 플랫폼 중 하나이며, 여러 도메인에서 다양한 종류의 분산 애플리케이션(DApp)을 설계하는 데 사용될 수 있다[8].

스마트 컨트랙트는 다양한 분야에 응용될 수 있다[9]-[11]. 최근 연구에서 공유 전동킥보드 이용자 관리를 위한 스마트 컨트랙트를 설계하고 가스 수수료 측정 등을 통해 성능을 분석하였다[12]. 또한 스마트 컨트랙트를 이용한 임금지불 자동체결 시스템이 제안되었다[13]. 이처럼 다양한 맥락에서 스마트 컨트랙트가 활용되고 있지만, 최근 떠오르고 있는 모바일 신분증에서는 관련 연구가 거의 진행되지 않았다. 따라서 본 연구는 모바일 운전면허증 발급을 위한 다양한 유형의 스마트 컨트랙트를 설계하고 성능을 평가한다.

III. 모바일 운전면허증 발급

한국에서 모바일 운전면허증이 2022년 7월에 도입되었다. 2023년 7월 기준으로 도로교통공단이 발급한 모바일 운전면허증은 약 142만 7천여건이다.

시범운영 기간에 발급한 운전면허증까지 포함하면 약 151만 4천여건에 달한다. 전체 면허증 발급 건수의 16.6% 수준이다.

그림 1은 자기주권 신원증명 모델을 활용한 모바일 운전면허증 발급 과정이다. 사용자(Holder)는 디지털 지갑 애플리케이션을 사용하여 자신의 신원 정보를 안전하게 저장한다. 여기에서 사용자는 기존의 실물 운전면허증 보유자이거나 운전 면허시험을 통과한 자로 제한한다.

1-1. 디지털 지갑은 사용자의 개인 식별자(DID)와 관련된 정보를 포함하여 신원 정보(예를 들어, 사용자의 이름, 생년월일, 주소 등)를 DID Resolver에 전송한다.

1-2. DID Resolver는 사용자의 DID 정보를 전송하고, DID Registry는 해당 DID 정보를 저장한다.

2-1. 사용자가 발급자(Issuer), 즉 도로교통공단에게 모바일 운전면허증 발급을 요청하면 발급자는 사용자의 신원을 확인하기 위해 인증 및 신원 확인 프로세스(예를 들어, 본인인증)를 진행한다.

2-2. 발급자는 사용자의 신원 정보를 확인하기 위해 DID Resolver를 통해 사용자의 DID 정보를 검색하며, 이를 통해 발급자는 사용자의 공개키, 신원 정보, 모바일 운전면허증 발급 요청 등을 확인할 수 있다.

2-3. 발급자는 사용자의 신원 확인 후, 모바일 운전면허증을 생성하여 사용자의 디지털 지갑에 전달한다. 모바일 운전 면허증은 VC 형태(subject, issuer, claims, signature의 구성 요소)로 발급되며, 사용자는 디지털 지갑을 통해 이를 안전하게 보관한다.

2-4. DID Resolver는 사용자의 DID 정보를 전송하고 DID Registry는 해당 DID 정보를 저장하거나 업데이트한다.

3-1. 사용자는 발급받은 모바일 운전면허증(VP)을 검증자(Verifier, 예를 들어, 교통 경찰)에게 제시한다. 이를 위해 사용자는 디지털 지갑을 사용하여 모바일 운전면허증(VP)을 검증자에게 제시할 수 있다. 예를 들어, QR코드를 보여줄 수 있다.

3-2. 검증자는 DID Resolver를 통해 사용자의 DID 정보(예를 들어, 사용자의 공개키, 인증서 등)를 확인한다. 이를 통해 검증자는 사용자의 모바일 운전면허증의 유효성을 판단할 수 있다.

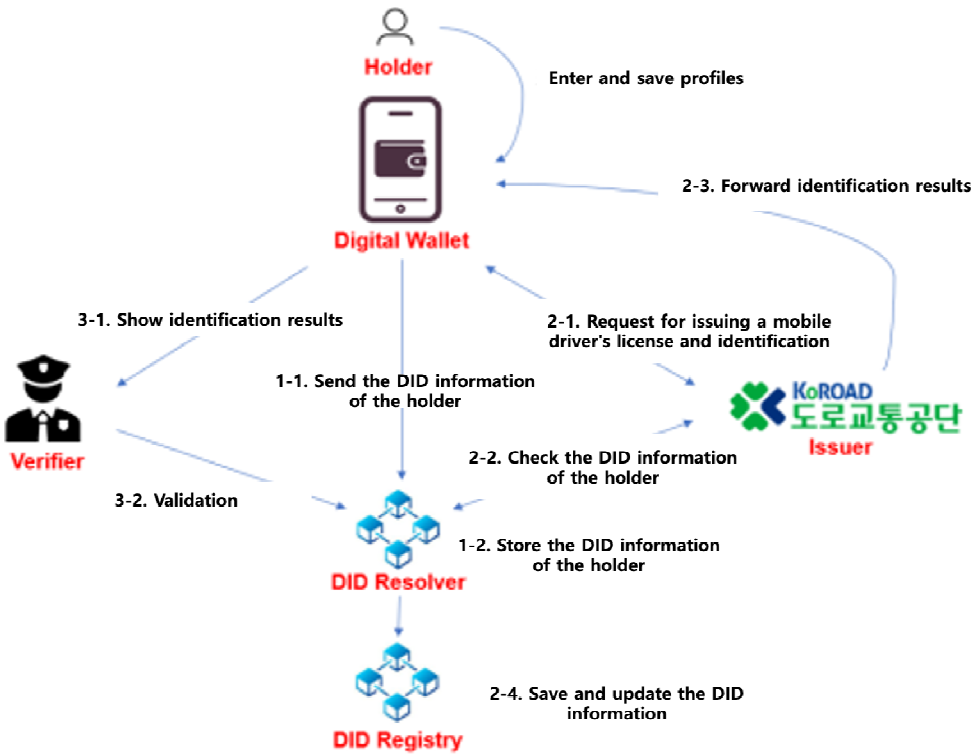


그림 1. 모바일 운전면허증 발급 과정 예시
Fig. 1. Example of the process of issuing a mobile driver's license

상기 프로세스에서 스마트 컨트랙트는 모바일 운전 면허증을 생성하고 사용자의 디지털 지갑에 안전하게 전달하는 역할을 할 수 있다. 즉 사용자가 모바일 운전면허증을 검증자에게 제시할 때, 스마트 컨트랙트는 검증을 위하여 사용자의 DID 정보를 확인하고 모바일 운전면허증의 유효성을 판단하는 데 중요한 역할을 한다. 스마트 컨트랙트가 이러한 검증 프로세스를 자동화하거나 간소화할 수 있는 것이다. 따라서 본 연구는 모바일 운전면허증 발급 프로세스에 스마트 컨트랙트를 적용한다. 구체적으로 모바일 운전면허증 발급에 필요한 신원 속성을 정의한 스마트 컨트랙트를 개발하고, 스마트 컨트랙트를 모델별로 다르게 구성하여 성능을 평가한다.

IV. 구현 및 평가

4.1 구현 방법

본 연구는 우선, Solidity 프로그래밍 언어를 사용하여 신원 속성(이름, 생년월일 등)을 정의한 스마트 컨트랙트를 생성한다. Solidity는 이더리움 등 블록체인 플랫폼에서 스마트 컨트랙트 작성과 구현에 사용되는 계약 지향 프로그래밍 언어이다. 다음으로, Truffle과 Hardhat을 통해 로컬 블록체인 환경인 Ganache에서 테스트를 수행한 후에 이더리움 블록체인 테스트넷인 Sepolia에 배포한다. Truffle과 Hardhat은 스마트 컨트랙트를 배포하고 실행할 수 있게 해주는 스마트 컨트랙트 개발 프레임워크이다. 테스트넷은 실제 블록체인 네트워크에 적용하기 전에 컨트랙트 코드를 미리 테스트해보는 환경을 말한다. 구현 환경은 표 1과 같다.

표 1. 구현 환경
Table 1. Implementation environment

Feature	Specification
CPU	Intel Core i7 2.80Ghz
RAM	10GB
Ethereum blockchain	Ganache
Testnet	Sepolia
Ethereum development environment	Truffle, Hardhat
Programming language	Solidity, JavaScript

본 연구는 스마트 컨트랙트를 모델 별로 다르게 구성함으로써, 모바일 운전면허증 발급에 있어서 스마트 컨트랙트가 어떻게 구현되고 운영되어야 최적의 성능을 유지하고 개인정보 보호를 강화할 수 있는지를 살펴본다. 본 연구에서 제안하는 모바일 운전면허증 발급을 위한 스마트 컨트랙트 5가지 모델은 표 2와 같다.

표 2. 모바일 운전면허증 발급을 위한 스마트 컨트랙트 모델
Table 2. Smart contract models for issuing mobile driver's license

Model	Content
Model 1 (base)	Store driver's license information and licensing events
Model 2	Model 1 + DID
Model 3	Model 1 + Limit data storage to one year
Model 4	Model 1 + Store data in array of bytes instead of string
Model 5	Model 1 + DID + Limit data storage to one year + Store data in array of bytes instead of string

기본 모델인 모델 1은 표 3과 같다. 스마트 컨트랙트는 모바일 운전면허증이 유효하고 해당 사용자에게 권한이 있는지 확인한 후에 기본적인 기능을 수행한다.

모델 2는 개인정보 보호 강화 목적으로 DID 기능을 추가한다. 모바일 운전면허증을 인증하는 경우 스마트 컨트랙트가 DID를 인증하거나 DID와 연결된 신원 정보를 검색하여 비즈니스 로직을 실행할 수 있다. 자기주권적인 신원 관리를 위해 DID는 사용자의 신원을 고유하게 식별하고, 이러한 신원 정보를 중앙화된 신원 공급자 또는 중개자 없이 관리할 수 있게 해준다. 이를 통해 사용자의 신원 정보는 분산형 레지스트리 또는 블록체인과 같은 안전한 분산 데이터 저장소에 저장되며, 암호화 및 디지털 서명을 통해 안전하게 관리된다. 또한 DID 사용을 통해 사용자는 필요한 경우에만 선택적으로 신원 정보를 공유할 수 있으며, 그 외의 정보는 숨길 수 있다. 즉 DID 기능은 최소한의 정보만을 공유하게 함으로써, 개인정보 노출의 위험을 감소시킬 수 있다[14].

표 3. 스마트 컨트랙트
Table 3. Smart contracts

```
pragma solidity ^0. 5. 16;
pragma experimental ABIEncoderV2;
contract DrivingLicense {
    struct License {
        string name;
        string birthdate;
        string location;
        string licenseType;
        string licenseNumber;
        string issueDate;
        string validUntil;
        }
    mapping(address => License) public licenses;
    function issueLicense(
        string memory name,
        string memory birthdate,
        string memory locationInfo,
        string memory licenseType,
        string memory licenseNumber,
        string memory issueDate,
        string memory validUntil
        )
    public {
        require(
            bytes(licenses[msg.sender].licenseNumber)
            .length == 0, "License already issued");
        licenses[msg.sender] = License({
            name : name,
            birthdate: birthdate,
            location: locationInfo,
            licenseType: licenseType,
            licenseNumber: licenseNumber,
            issueDate: issueDate,
            validUntil: validUntil
            });
        }
    function getLicense() public view returns
    (License memory) {
         return licenses[msg.sender];
         }
```

모델 3은 데이터 저장 기한 제한 기능을 추가한다. 스마트 컨트랙트에서 기간 제한은 잠재적으로 해로운 동작 또는 악의적인 활동을 방지하고 스마트 컨트랙트를 안전하게 유지하는 데 도움이 된다. 따라서 이러한 기능은 개인의 잊혀질 권리를 보호하는데 도움을 줄 수 있다. [1]은 자기주권신원기술의 국내 서비스 사례를 분석하면서 발급하는 인증서에 적정 기간을 두어 보안을 강화할 필요가 있다고 하였다. 데이터 저장 기한을 제한한다고 해서 가스 비용에 직접적인 영향을 미치지는 않을 수 있지만, 스마트 컨트랙트 코드 및 관리 작업 등 추가적인 리소스가 필요할 수 있으며, 이로 인해 가스 비용이 증가할 수 있다[15]. 모델 3은 유효기간 관리를 위해 'expireDate' 배열을 사용하고, 'issueLicense' 함수를 통해 이 배열에 각 디지털 운전면허의 만료일을 저장한다. 'getLicense' 함수 내에서는 현재 시간과 'expireDate' 배열에 저장된 만료일을 비교하여 면허의 만료 여부를 확인한다. 이를 통해 모바일 운전면허증이 1년 후에 만료되도록 설정하고, 만료된 면허는 더 이상 조회되지 않도록 보장된다.

모델 4는 문자열(String) 대신 바이트 배열로 데이터를 저장한다. Solidity는 다양한 정적 데이터 유형뿐만 아니라 일부 동적 유형, 즉 동적 배열 및 매핑도 지원한다[16]. 바이트 배열은 데이터를 바이트로 저장하는 데 사용되고 크기가 동적으로 조절 가능하지만, 문자열은 인코딩된 문자 데이터를 저장하는 데 사용된다. 따라서 데이터 저장방식에 따라 가스 비용과 스토리지 요구 사항에서 차이가 발생할 수 있다. 스마트 컨트랙트에서 모바일 운전면허 정보를 바이트 배열로 저장하면 블록체인에 저장된 데이터를 변경 또는 수정하기가 어렵기 때문에 개인정보의 무결성을 유지하기에 적합하다. 다만 바이트 배열로 저장된 정보는 인간이 읽기 어렵기 때문에 'bytesToString' 함수를 사용하여 바이트 배열로 저장된 정보를 문자열로 변환해 주어야 한다.

표 4. 개선된 스마트 컨트랙트를 위한 Solidity 코드
Table 4. Solidity code for enhanced smart contracts

```
function storeDIDData(
    string memory resolverData, string memory
    registryData) public {didData[msg.sender] =
                        DIDData({resolverData:
                                resolverData,
                                registryData:
                                registryData});
                    }

function isBefore(uint _date1, uint _date2) public pure
returns (bool) {if (_date1 < _date2) return false;
                return true;
                }

function bytesToString(bytes memory data) public pure
    returns (string memory) {
        return string(data);
        }
```

마지막으로, 모델 5는 앞에서 언급한 DID 기능(모델 2), 데이터 저장 기한 제한(모델 3), 바이트 배열 저장(모델 4)이 모두 포함된다. 추가된 코드는 표 4와 같다.

4.2 성능 평가

본 연구에서 제안한 모바일 운전면허증 발급을 위한 스마트 컨트랙트의 성능을 평가하기 위해서, 다음과 같은 3가지 시나리오를 고려한다. 즉 발급 조건을 다르게 한다. 첫째, 모바일 운전면허증을 '단건'으로 10건 발급한다. 둘째, 모바일 운전면허증을 '연속적'으로 10건 발급한다. 마지막으로, 모바일 운전면허증을 '동시에' 10건 발급한다.

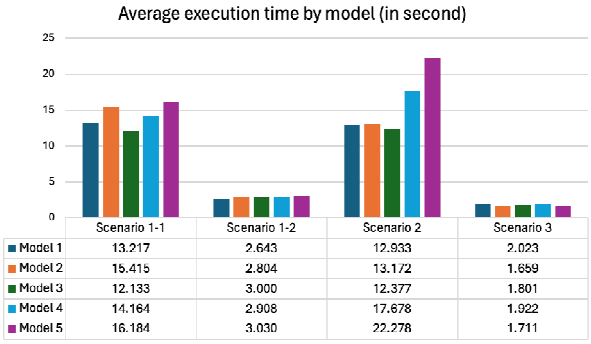
성능 평가에 대한 지표로 소요시간과 가스 수수료를 측정한다. 이더리움 플랫폼에서 스마트 컨트랙트를 배포하고 실행하기 위해서는 사용 메모리와 연산량에 따라 가스 수수료를 지불해야 한다. 측정된 값의 신뢰성을 확보하기 위해 테스트를 10회 실시한다.

성능 평가 시 단건 발급 시나리오는 Truffle과 Hardhat에서 실행하며, 연속 및 동시 발급 시나리오는 Hardhat에서만 실행한다. 표 5와 같이 신원 속성(이름, 생년월일 등)을 입력할 수 있다.

표 5. 운전면허증 정보 입력
Table 5. Driver's license information entry

```
const name = "John Doe";
const birthdate = "2000년 01월 01일";
const location = "Seoul Seodaemungu Yonseiro 1";
const licenseType = "운전 면허";
const licenseNumber = "1234567890"
const issueDate = "2023년 01월 01일";
const validUntil = "2032년 12월 31일";
```

실험 결과, 그림 2에서 보는 바와 같이, 데이터 저장 기한을 1년으로 제한한 모델 3이 대체적으로 소요시간이 짧았다. 반면 DID 기능을 추가한 모델 2와 바이트로 변환한 모델 4는 기본 모델인 모델 1보다 소요시간이 길었다. 단건 발급의 경우(시나리오 1-1과 1-2), 모든 모델에 대해서 Hardhat의 소요시간이 Truffle보다 상당히 짧았다. Truffle과 Hardhat 간에 실행 시간 차이가 크게 나타나는 이유는 Hardhat이 Truffle보다 내부적으로 최적화되어 있거나, 더 효율적인 가스 관리를 수행하고 있기 때문일 수 있다.



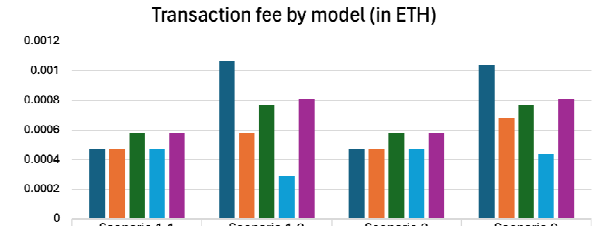
	Scenario 1-1	Scenario 1-2	Scenario 2	Scenario 3
Model 1	13.217	2.643	12.933	2.023
Model 2	15.415	2.804	13.172	1.659
Model 3	12.133	3.000	12.377	1.801
Model 4	14.164	2.908	17.678	1.922
Model 5	16.184	3.030	22.278	1.711

그림 2. 모델별 평균 소요시간(단위: 초)
Fig. 2. Average execution time by model (in second)

* Scenario 1-1: one time(Truffle)
Scenario 1-2: one time(Hardhat)
Scenario 2: consecutively 10 times(Hardhat)
Scenario 3: simultaneously 10 times(Hardhat)

이더리움 네트워크는 유한한 리소스를 가지고 있으며, 가스를 사용하여 트랜잭션을 처리하고 스마트 계약을 실행하기 때문에 효율적인 코드를 작성하여 가스 소비량을 최소화하는 것이 중요하다[17]. 이더리움 스마트 컨트랙트 가스 수수료는 새로운 스마트 컨트랙트를 이더리움 블록체인에 배포하거나 생성할 때 발생하는 계약 생성 수수료와 스마트 컨트랙트를 실행하거나 블록체인에서 데이터를 변경하는 모든 거래에 대해 발생하는 거래 수수료로 구분된다. 본 연구에서는 실제 운영 시에 더 중요한 역할을 하는 거래 수수료를 중심으로 성능을 평가한다.

그림 3에서 보는 바와 같이, 개인정보가 블록체인에서 계속해서 노출되는 것을 방지하고, 일정 기간 이후에는 조회가 제한될 수 있도록 스마트 컨트랙트에서 자동으로 저장 기한을 제한하는 모델 3이 모든 시나리오에서 모델 2와 모델 4보다 가스비용이 다소 높게 나왔다. 시나리오 1-2와 시나리오 3에서는 모델 1이 가스 비용이 가장 높게 나왔다. 또한, 개발 환경에 따라 성능이 다르게 나타났는데, 단건 발급의 경우(시나리오 1-1과 1-2), 대체적으로 Hardhat 환경(시나리오 1-2)에서 가스 비용이 높게 나왔다.



	Scenario 1-1	Scenario 1-2	Scenario 2	Scenario 3
Model 1	0.00047	0.00107	0.00047	0.00104
Model 2	0.00047	0.00058	0.00047	0.00068
Model 3	0.00058	0.00077	0.00058	0.00077
Model 4	0.00047	0.00029	0.00047	0.00044
Model 5	0.00058	0.00081	0.00058	0.00081

그림 3. 모델별 거래 수수료(단위: ETH)
Fig. 3. Transaction fee by model (in ETH)

* Scenario 1-1: one time(Truffle)
Scenario 1-2: one time(Hardhat)
Scenario 2: consecutively 10 times(Hardhat)
Scenario 3: simultaneously 10 times(Hardhat)

연속적으로 발급하는 경우(시나리오 2)와 동시에 발급하는 경우(시나리오 3)를 비교해보면, 동시에 발급하는 경우가 대체적으로 가스 비용이 높게 나왔다. 본 연구의 결과는 모바일 운전면허증 발급을 위한 스마트 컨트랙트의 성능이 구현 방식(모델)과 개발 환경 그리고 발급 조건(시나리오)에 따라 다르게 나타날 수 있음을 말해준다.

V. 결론 및 향후 과제

스마트 컨트랙트는 블록체인 기반의 투표 시스템, 의료 기록 관리, 자동차 보험, 분산금융서비스(DeFi) 등 다양한 분야에서 활용될 수 있다. 스마트 컨트랙트가 어떻게 설계되고 운영되어야 최적의 성능을 발휘하면서도, 사용자의 개인정보를 효과적으로 보호할 수 있는지에 대한 연구가 필요하다. 이에 본 연구는 효율성과 보안성에 초점을 맞춰 모바일 운전면허증 발급을 위한 다양한 스마트 컨트랙트 모델을 설계하고 이들의 성능을 평가했다.

테스트 결과, 이더리움 테스트넷에서의 처리 시간과 거래 수수료는 개발환경과 테스트 시나리오에 따라 달라질 수 있음을 확인했다. 즉 스마트 컨트랙트의 성능은 여러 요인에 영향을 받는다. 기본 모델에 DID 기능을 추가하여 개인정보 보호가 강화된 모델 2가 대체적으로 소요시간이 짧고 거래 수수료가 낮게 나왔다. 이것은 모바일 운전면허증 발급에 있어서 제안된 방식 중 모델 2의 방식이 가장 우수하다는 것을 말해준다. 한편 자원 효율성 측면을 고려하면, 소요시간은 상대적으로 길지만 거래 수수료가 가장 낮은 모델 4의 방식(바이트 변환)이 우수하다고 할 수 있다.

스마트 컨트랙트의 활용 범위가 다양함에도 불구하고, 스마트 컨트랙트를 최근 각광받고 있는 모바일 운전면허증에 적용한 연구를 거의 찾아볼 수 없다. 소수의 연구가 자기주권 신원증명 모델을 모바일 운전면허증에 적용하였지만, 실험이 아닌 개념적 분석에 그쳤다[5]. 본 연구는 스마트 컨트랙트를 활용하여 모바일 운전면허증 발급 시스템의 효율성과 사용자의 개인정보 보호를 강화하는 방법을 제안함으로써, 디지털 신원증명 분야에 학술적으로 기여한다. 또한 본 연구는 다양한 발급 조건과 구현 방식을 고려하였다. 따라서 본 연구의 결과는 블록체인 및 스마트 컨트랙트 개발자에게 스마트 컨트랙트를 설계하고 평가하는데 있어서 유용한 시사점을 제공한다.

이 연구에서는 Sepolia 환경에서 스마트 컨트랙트를 실행하여 실험을 수행했다. 테스트넷 환경에서의 실험 결과는 실제 블록체인 네트워크에서의 동작과 다소 다를 수 있다. 또한 본 연구는 DID 데이터, 저장기한 제한 및 바이트 변환 등을 스마트 컨트랙트에 추가 저장하는 방법에 중점을 두었지만, 블록체인에서의 개인정보 보호와 관련된 데이터 암호화나 신원 검증 등 다른 측면도 고려되어야 할 필요가 있다. 마지막으로, 본 연구에서 다룬 발급 뿐만 아니라 수정, 조회 등 다양한 경우에 대한 연구도 필요하다.

References

[1] J. H. Lee, J. W. Kim, C. S. Kim, and J. H. Yang, "A Study on Strengthening Personal Information Sovereignty through Analysis of Domestic Service Cases and Research Projects of Self-Sovereign Identity Technology", Journal of Korea Institute of Information, Electronics, and Communication Technology, Vol. 13, No. 6, pp. 575-589, Dec. 2020. https://dx.doi.org/10.17661/jkiiect.2020.13.6.575.

[2] M. S. Ferdous, F. Chowdhury, and M. O. Alassafi, "In search of self-sovereign identity leveraging blockchain technology", IEEE access, Vol. 7, pp. 103059-103079, Jul. 2019. https://doi.org/10.1109/ACCESS.2019.2931173.

[3] O. Avellaneda, A. Bachmann, A. Barbir, J. Brenan, P. Dingle, K. H. Duffy, E. Maler, D. Reed, and M. Sporny, "Decentralized identity: Where did it come from and where is it going?", IEEE Communications Standards Magazine, Vol. 3, No. 4, pp. 10-13, Dec. 2019. https://doi.org/10.1109/MCOMSTD.2019.9031542.

[4] Financial Security Institute, "Distributed ID and Technology Development Trends", e-Finance and Financial Security, Vol. 16, No. 4, pp. 15-39, Apr. 2019.

[5] E. Kim, H. Kim, H Jung, and N. Y. Lee, "A Study on Mobile Driver's License for Self-sovereign Identity Model Based on Decentralized Identity", Journal of The Korea Society of Information Technology Policy & Management, Vol. 12, No. 4, pp. 1869-1874, Aug. 2020.

[6] S. Yoo, S. Yoo, J. Jo, and A. Son, "Proposal of Verifiable Credential System for Smart Contract Implementation based on Self-Sovereign Identity", The Journal of Korean Institute of Information Technology, Vol. 18, No. 12, pp. 141-149, Dec. 2020. https://doi.org/10.14801/jkiit.2020.18.12.141.

[7] The Path to Self-Sovereign Identity, http://www.lifewithalacrity.com/2016/04/the-path-to-self-soverereign-identity.html [accessed: Mar. 26, 2024]

[8] R. Belchior, B. Putz, G. Pernul, M. Correia, A. Vasconcelos, and S. Guerreiro, "SSIBAC: self-sovereign identity based access control", In 2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom), Guangzhou, China, pp. 1935-1943, Dec. 2020. https://dx.doi.org/10.1109/TrustCom50675.2020.00264.

[9] M. M. Merlec and H. P. In, "SC-CAAC: A Smart Contract-Based Context-Aware Access Control Scheme for Blockchain-Enabled IoT Systems", IEEE Internet of Things Journal, Vol. 11, No. 11, pp. 19866-19881, Jun. 2024. https://doi.org/10.1109/JIOT.2024.3371504.

[10] Z. A. Khan and A. S. Namin, "A Survey of Vulnerability Detection Techniques by Smart Contract Tools", IEEE Access, Vol. 12, pp. 70870-70910, May 2024. https://doi.org/10.1109/ACCESS.2024.3401623.

[11] W. C. Tsai and C. W. Shen, "Using a smart contract for the floral supply chain", Asia Pacific Management Review, Jan. 2024. https://doi.org/10.1016/j.apmrv.2023.12.004.

[12] S. Lee, M. Park, N. Kim, and S. Seo, "Blockchain-Based Shared Electric Kickboard User Management Model", KIPS Transactions on Computer and Communication Systems, Vol. 12, No. 7, pp. 217-226, Jul. 2023. https://doi.org/10.3745/KTCCS.2023.12.7.217.

[13] S. Yoo, K. Kim, and J. Ahn, "Designing a Blockchain-based Smart Contract for Seafarer Wage Payment", Journal of the Korean Society of Marine Environment and Safety, Vol. 27, No. 7, pp. 1038-1043, Dec. 2021. https://doi.org/10.7837/kosomes.2021.27.7.1038.

[14] Decentralized Identifiers (DIDs) v1.0, https://www.w3.org/TR/did-core [accessed: Mar. 26, 2024]

[15] J. Golosova and A. Romanovs, "The Advantages and Disadvantages of the Blockchain Technology", In 2018 IEEE 6th workshop on advances in information, electronic and electrical engineering (AIEEE), Vilnius, Lithuania, pp. 1-6, Nov. 2018. https://doi.org/10.1109/AIEEE.2018.8592253.

[16] P. Kostamis, A. Sendros, and P. Efraimidis, "Exploring ethereum's data stores: A cost and performance comparison", In 2021 3rd Conference on Blockchain Research & Applications for Innovative Networks and Services (BRAINS),

Paris, France, pp. 53-60, Sep. 2021. https://doi.org/10.1109/BRAINS52497.2021.9569804.

[17] A. A. Zarir, G. A. Oliva, Z. M. Jiang, and A. E. Hassan, "Developing cost-effective blockchain -powered applications: A case study of the gas usage of smart contract transactions in the ethereum blockchain platform", ACM Transactions on Software Engineering and Methodology (TOSEM), Vol. 30, No. 3, pp. 1-38. Mar. 2021. https://doi.org/10.1145/3431726.

저자소개

이 상 윤 (Sangyun Lee)

1993년 2월 : 연세대학교 통계학과(이학사)
2024년 2월 : 연세대학교 정보대학원(정보시스템학석사)
2000년 2월 ~ 현재 : 한국거래소 수석
관심분야 : 개인정보 보호, 디지털 금융, 가상자산 이용자 보호

김 범 수 (Beomsoo Kim)

1990년 2월 : 서울대학교 경영학과(경영학학사)
1992년 2월 : 서울대학교 경영학과(경영학석사)
1999년 2월 : University of Texas at Austin(경영학박사)
1999년 ~ 2002년 : University of Illinois at Chicago 조교수
2002년 ~ 현재 : 연세대학교 정보대학원 교수
관심분야 : ICT의 효과적 활용, 데이터 거버넌스, 프라이버시, 개인정보 보호

박 재 영 (Jaeyoung Park)

2012년 8월 : 숭실대학교 정보통신전자공학부(공학사)
2017년 2월 : 연세대학교 정보대학원(정보시스템학석사)
2021년 8월 : 연세대학교 정보대학원(정보시스템학박사)
2023년 6월 ~ 현재 : 농협금융지주 금융연구소 책임연구원
관심분야 : 프라이버시, 개인정보 보호, 디지털 금융