Journal of KIIT. Vol. 18, No. 12, pp. 141-149, Dec. 31, 2020. pISSN 1598-8619, eISSN 2093-7571
http://dx.doi.org/10.14801/jkiit.2020.18.12.141

자기주권 신원증명 기반 스마트 계약 구현을 위한 자격증명 시스템 제안

유수민*[1], 유수빈*[2], 조정화*[3], 손애선*[4]

Proposal of Verifiable Credential System for Smart Contract Implementation based on Self-Sovereign Identity

Su-Min Yoo*[1], Soo-Bin Yoo*[2], Jung-Hwa Jo*[3], and Ae-Seon Son*[4]

본 연구는 문화체육관광부 및 한국저작권위원회의 2019년도 저작권연구개발사업의 연구결과로 수행되었음.
(2019-SC-9500)

요 약

현대 사회에서는 인터넷 상의 데이터 유출을 보장하면서 데이터 송 수신자들의 증명을 확인시켜 주는 공개키 기반구조(Public Key Infrastructure)를 활용하여, 온라인상에서 안전한 데이터 교환과 신원인증을 위해서 대칭키, 비대칭키 암호화 기반의 디지털 인증서 기술을 제공하고 있다. 하지만 CA(Certificate Authority, 인증기관)의 보안이 취약하다면 CA의 디지털 인증서를 사용하는 모든 사용자들 또한 데이터 유출에 취약해지며, 또한 유출된 정보로 인해 발생할 수 있는 2차적인 피해가 있다. 본 논문은 디지털 인증서 와 키 유출로 인한 피해를 방지하기 위해 자기주권 신원증명 기술을 활용하여 분산 원장 환경인 블록체인을 사용하여 데이터의 무결성을 보장하고 블록체인에 등록된 DID(Decentralized Identifier) Document의 공개키를 사용하여 탈중앙화 구조에서 안전하게 공개키를 교환할 수 있는 보안성을 확보하는 체계를 제안한다.

Abstract

Modern society has introduced a public key infrastructure that verifies the proof of data senders while ensuring data leakage on the Internet, providing digital certificate technology based on symmetric and asymmetric key encryption for safe data exchange and identity authentication online. But CAof security is fragile, CA of all users using digital certificates are also vulnerable, and also in a data breach. There is a secondary damage that can be caused by information leaked. In order to prevent damage caused by digital certificates and key leaks, this paper proposes a system to ensure the integrity of data by using the distributed ledger environment, the Block chain, and secure the security to exchange the open key safely in a decentralized structure using the open key of the DID (Decentralized Identifier) Document registered in the block chain.

Keywords
blockchain, decentralized identity, information security smart contract, uthentication technology

* 경주스마트미디어센터 연구원 (*[2]교신저자)
- ORCID[1]: https://orcid.org/0000-0001-5643-1422
- ORCID[2]: https://orcid.org/0000-0001-8095-7862
- ORCID[3]: https://orcid.org/0000-0002-1144-7153
- ORCID[4]: https://orcid.org/0000-0001-5153-2929

· Received: Nov. 24, 2020, Revised: Dec. 22, 2020, Accepted: Dec. 25, 2020
· Corresponding Author: Soo-Bin Yoo
Strategic Planning Dept. Gyeongju Smart Media Center, 587-18
Gyeonggam-ro, Gyeongju, 38118, Korea
Tel.: +82-54-781-2943, Email: yoosoobin@silgam.or.kr

Ⅰ. 서 론

현대 사회에서 사람들이 삶의 질을 높여주는 서비스의 시작은 인증이라 해도 과언이 아니다. 일상생활에서 인증 체계를 통해 은행에서 업무를 보거나 어떠한 물건을 구매하기 위해 카드로 결제한 후 사인을 하는 과정도 본인임을 증명하기 위한 신원인증 과정이다. 즉, 오늘날 편리한 삶을 누릴 수 있게 한 요소 중 하나는 바로 신원인증 기술이다. 보통 쉽게 접할 수 있는 신원인증 기술은 바로 신분증이다. 자신을 증명해줄 수 있는 수단인 신분증에는 다양한 문제점이 발생한다. 만약 신분증을 분실한다면 신분증 내 개인정보를 보호해줄 장치가 없기 때문에 개인정보 유출을 막을 수 없고, 신분증 정보를 볼 수 없도록 하는 암호화 장치도 존재 하지 않는다. 신분증의 유연성이나 프라이버시 측면에서도 취약점이 발생하게 된다. 신분증의 정보 요소는 매우 한정적인 정보만을 다루고 그 요소가 정해져있으므로 새로운 요소를 추가하거나 삭제가 불가능하며 필요한 속성만 선택해서 인증하는 것 또한 불가능하다. 다른 인증 기술로는 통합 로그인 기술이 있고 이 기술 역시 취약점이 발생한다. 각각의 웹 사이트마다 번거로운 회원가입 밟아야만 하고 다수의의 로그인 정보를 일일이 관리해야 하는 불편한 경우가 있는데 이를 보완하기 위해 사용되는 신원인증 기술이 SSO(Single Sign On, 통합 로그인)이다. SSO란 IdP(Identity Provider) 웹 사이트의 회원정보를 이용해서 RP(Relying Party) 웹 사이트로 회원 가입 혹은 로그인을 할 수 있는 기능을 말한다[1][2]. 예를 들어, 쇼핑몰 사이트에 회원가입을 위해 사용자 정보를 모두 작성하는 대신, 기존의 구글 혹은 페이스북 계정 정보로 간편하게 가입이 가능하다. 이때의 경우는 IdP는 구글 이고, RP는 쇼핑몰 사이트가 된다. 제어성과 보안성 측면의 SSO 시스템에서 실제 사용자 정보는 사용자가 아닌 IdP의 보안 시스템이 직접 전달해야하기 때문에 악의적인 IdP가 사용자 개인정보를 무단으로 사용될 수 있다[1][2]. 실제 구글은 2018년 5250만 명의 개인정보가 노출된 것으로 밝혀졌다.

본 논문은 DID(Decentralized Identifier, 탈중앙화 식별자)와 DID document를 발급하여 분산 원장 환경인 블록체인 환경에서 안전하게 공개키를 교환할 수 있는 체계를 제안한다. 논문의 2장에서 기존 신원인증 수단의 특징과 취약점을 분석한다. 3장에서는 자기주권 신원증명 기반 기술을 활용하여 데이터의 무결성을 보장하고 안전하게 공개키를 교환할 수 있는 방안을 제시하고 4장에서는 본 논문에 대한 결론과 향후 방향을 제시한다.

Ⅱ. 관련 연구

본 장에서는 현대사회에서 사용되고 있는 신원인증 수단과 취약점에 대해서 분석한다.

2.1 디지털 인증서

디지털 인증서는 사용자의 신원을 전자 수단으로 인증하는 방식으로, 흔히 온라인에서는 대칭키, 비대칭키 암호화 기반의 디지털 인증서가 사용된다. 대칭키 암호화는 동일한 두 개의 키 쌍을 사용해서 암호화/복호화를 수행하고, 비 대칭키 암호화는 서로 다른 두 개의 키 쌍인 공개키와 비밀키를 사용해서 암호화/복호화를 수행한다[3].

2.2 대칭키 암호화 방식

대칭키 암호화 방식은 암호화와 복호화에 같은 대칭 키를 사용하는 암호화 알고리즘이다. 그림 1과 같이 사용자와 수신자가 대칭키를 이용하여 암호화 통신을 하는 과정을 나타낸다[3][4].

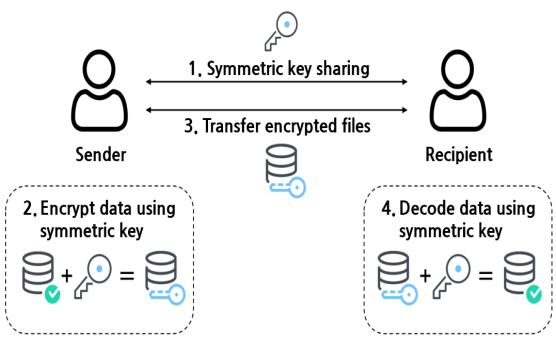

그림 1. 대칭키를 이용한 암호화 방식
Fig. 1. Encryption using symmetric keys

사용자가 우선적으로 대칭키를 생성하고 안전한 방법으로 상대방에게 공유를 해야 한다. 이후 공유한 대칭키를 이용하여 데이터를 암호화 한 후 사용자에게 전송을 하고 암호화된 데이터를 수신한 수신자는 송시자로부터 공유된 대칭키를 이용하여 암호화된 데이터를 복호화 할 수 있다. 송신자와 수신자 간 암호화 된 데이터는 서로 간의 공유된 대칭키가 있어야만 복호화를 할 수 있기 때문에 해커가 중간에서 데이터를 가로채더라도 공유 된 대칭키가 없는 이상 데이터의 내용을 알 수 없다. 하지만 대칭키를 공유하는 과정에서 해커로 인해 대칭키를 탈취당할 수 있다면 해커는 탈취한 대칭키를 이용하여 암호화된 데이터의 내용을 모두 알 수 있다는 취약점이 발생한다[4].

2.3 비대칭키 암호화 방식

비대칭키 암호화 방식은 비밀키(Private key), 공개키(Public key)라 불리는 서로 다른 키 쌍을 사용해서 암호화와 복호화를 수행한다. 기본적으로 공개키는 외부에 공개할 수 있고, 비밀 키는 주인이 안전하게 보관해야 하며 절대 유출해선 안 되며, 비밀키로 암호화를 수행하면 데이터 송신자 에 대한 사용자 인증이 가능하다. 그리고 공개키로 암호화를 수행하면 데이터 암호화가 가능하다. 그림 2는 공개키와 비밀키를 이용해서 사용자 인증과 데이터 암호화 구현 방법을 나타낸다[4].

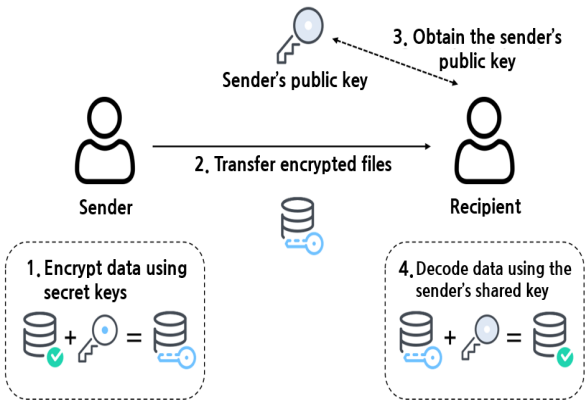

그림 2. 비대칭키를 이용한 암호화 방식
Fig. 2. Encrtption using asymmetric keys

우선, 송신자는 자신의 비밀키를 이용해서 전송할 데이터를 암호화한 후 파일을 수신자에게 전송한다. 송신자의 비밀키로 암호화된 파일은 오직 송신자의 공개키로 복호화가 가능하기 때문에 만약 암호화된 파일이 복호화가 되지 않는다면 해당 데이터는 송신자가 보낸 것이 아니게 된다. 하지만 비대칭키에도 다음과 같은 취약점이 존재한다. 송신자에서 본인의 비밀키로 공개키를 만들고 수신자에게 보내게 될 때, 수신자에게 전달하기 전에 해커가 패킷을 가로채 자신의 공개키를 송신측에 전달한다. 그리고 수신자는 해커의 공개키로 데이터를 암호화해서 보내면, 이 때 해커가 또 패킷을 가로채서 본인의 개인키로 데이터를 복호화하면 원본 데이터를 알아낼 수 있다.

2.4 SSL/TLS 프로토콜

앞부분과 같이 비 대칭키 암호화 방식만 이용해서 암호화를 수행하면 보안에 취약하다. 이를 방지하기 위해 CA(Certificate Authority, 인증기관)라고 부르는 인증노드를 활용한 SSL/TLS 프로토콜을 사용한다. SSL/TLS를 사용하면 클라이언트/서버 간 통신을 암호화하기 때문에 안전하게 통신할 수 있다. 또한 CA에서 발행한 서버 인증서를 사용해 서버의 신뢰성을 증명할 수 있다. 서버 인증서는 CA가 서버의 신뢰성을 증명하기 위한 디지털 데이터로, 서버의 공개 키와 사용자의 정보가 합쳐져 있다. 서버 인증서에는 CA가 전자 서명을 하고 있으며, 웹 브라우저(클라이언트)에 Root CA 인증서를 사용해 확인할 수 있다. SSL/TSL 데이터 흐름은 그림 3과 같다[5].

2.5 블록체인

블록체인 기술은 누구나 열람할 수 있는 분산원장 환경의 장부에 거래 내역을 투명하게 기록하고, 모든 참여자가 동일하게 관리 할 수 있는 분산형 데이터 저장 기술이다. 기존의 중앙 집중 시스템에서 데이터를 위변조를 하기 위해서는 중앙 서버를 공격하면 가능했지만 블록체인은 여러 참여자가 동일한 데이터를 나눠 가지고 있기 때문에 무결성을 지킨다[6][7].

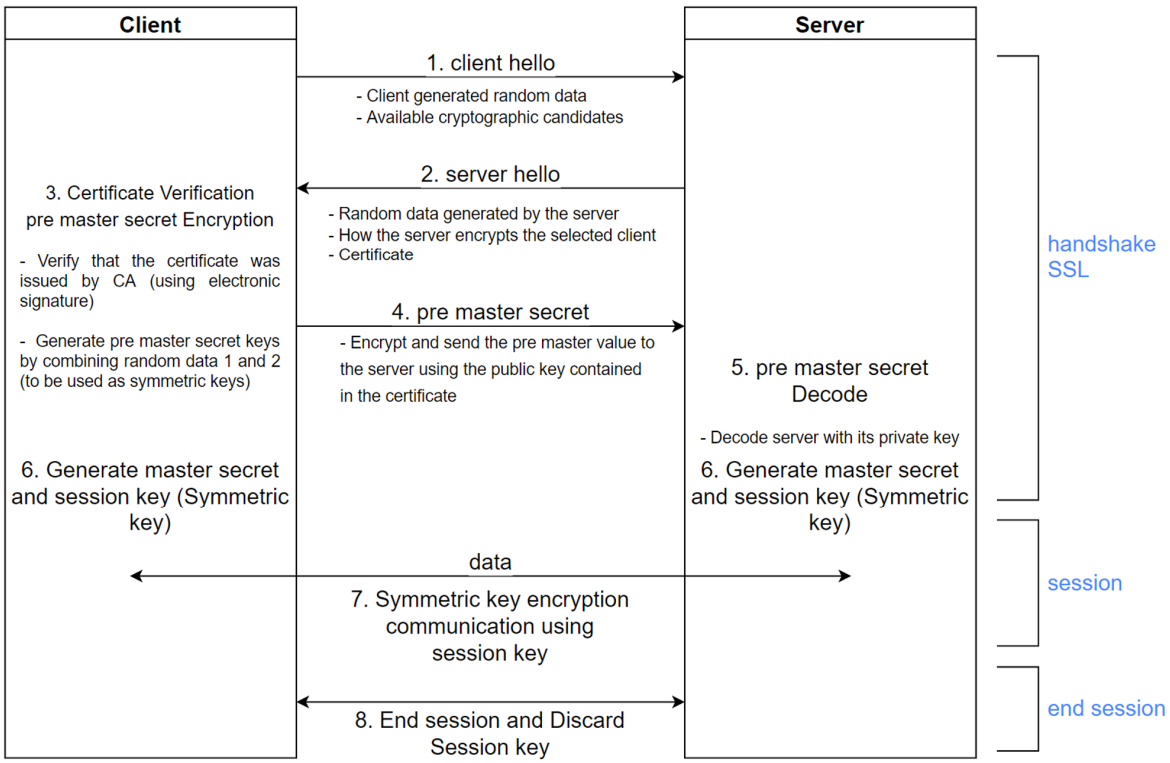

그림 3. SSL/TLS 프로토콜 데이터 흐름
Fig. 3. SSL/TLS protocol data flow

2.6 스마트 컨트랙트

스마트 컨트랙트는 1996년 Nick Szabo에 의해 고안된 개념으로 당사자들이 다른 약속에 따라 수행하는 프로토콜을 포함하여 디지털 형식으로 지정된 일련의 약속이라고 정의했다. 블록체인에서 스마트 컨트랙트는 일반적으로 솔리디티 같은 고급 언어로 작성되며 트랜잭션을 저장하고 사용할 수 있다[7].

III. 제안하는 시스템

기존의 많은 신원증명 기술들이 사용되고 있지만, 각각의 문제점들이 발생한다. 본 논문은 기존 신원 증명 기술의 취약점을 방지하고 DID 정보가 유출되더라도 타인이 정보를 확인 할 수 없는 자기주권 신원증명 기반의 계약 체결 DID 및 VC 체계를 제안한다.

3.1 DID 구조

기존 사람의 주민등록번호나 상품 일련번호 등의 중앙화된 식별자는 중앙 기관을 통해 발급받고 통제되는 구조이다. 반면 DID는 사용하는 사람 스스로 생성하고 제어할 수 있는 분산형 식별자이다[8].

그림 4는 플랫폼의 DID 주소 예시를 나타낸다. DID는 DID scheme, DID method, Method-specific identifier 3가지로 구성되어 있다. DID scheme은 URI(Uniform Resource Identifier)가 어떤 프로토콜을 사용해서 자원에 접근하는지 명시한다. URI scheme는 사람들이 흔히 사용하는 http 및 https를 포함한 다양한 프로토콜이 정의되어 있는데, DID scheme에 did가 들어가므로 정의한 자원 접근 방식에 따라 자원을 찾아 간다[8][9].

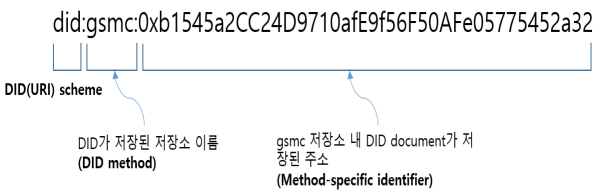

그림 4. 플랫폼 DID 구조 예시
Fig. 4. Platform DID structure example

DID method는 DID document가 어떤 저장소에 저장되어 있는지 보여준다. 다음 그림과 같이 DID method에 gsmc가 명시되어 있는데 gsmc는 이더리움 기반 프라이빗 네트워크이다. 따라서 gsmc 이름을 가진 이더리움 기반 프라이빗 네트워크에 접근하여 DID document를 검색한다. 마지막으로 DID method가 가리키는 저장소 내 DID document가 저장된 정확한 위치를 검색하기 위해서는 Method-specific identifier가 필요하다. DID method를 참조해서 프라이빗 이더리움 블록체인에 접근한 후 Method-specific identifier를 이용해 검색하면 DID document를 불러올 수 있다. 여기서 Method-specific identifier 부분은 DID method가 gsmc이므로, 어카운트 주소가 Method-specific이다[8][9].

3.2 플랫폼 DID document 구조

DID document에는 DID의 소유권을 증명할 수 있는 인증 방법이 포함되어 있다. 아래 그림 5는 DID document의 구조를 보여준다. 전체 구조에서 DID document는 @context, id, verification Method 그리고 authentication으로 구성된다. 먼저 id 항목은 id를 통해 식별되는 객체의 DID가 들어가는 항목으로, 일반적으로 DID와 DID document를 생성 혹은 생성 요청을 하고 등록한 사람의 DID가 id 항목에 들어간다. verification Method 항목에는 id, type, controller, signature라는 4가지 세부 항목들로 구성된다. id는 verification Method내 사용할 수 있는 인증 키의 위치를 나타낸다. type 항목은 id의 인증 방식이 들어가는데, 여기서는 RSA 비대칭키 인증이 사용된다. signature는 소유권 인증에 사용될 데이터가 저장돼 있으며, 해당 DID document 구조에서 저장소가 이더리움 기반이기 때문에 서명 값이 비대칭키 인증 방식에 사용되는 공개키로 저장된다. 마지막으로 controller 항목에는 해당 공개키와 쌍으로 이루는 비밀키를 가진 사람의 DID가 저장된다[10][11].

3.3 VC 데이터 모델 구조

사용자는 발행인으로부터 자신의 ID 속성을 증명할 수 있는 자격증명을 발급받을 수 있다. 이러한 자격증명을 VC(Verifiable Credential)라고 부르는데, 주민등록증과 같은 신분증부터 졸업증명서, 재직증명서 등 자신을 표현할 수 있는 모든 ID 속성이 VC에 포함될 수 있다. VC는 Credential metadata, Claim(s), Proof(s)로 구성된다. 우선 Credential metadata는 VC를 누가 발행했는지, VC가 명시하고 있는 객체(Credentail subject)를 나타낸다[12].

```
{
"@context": "https://www.w3.org/ns/did/v1",
"id": "did:gsmc:123456789abcdefghi",
"verificationMethod": [{
"id": "did:gsmc:123456789abcdefghi#key-1",
"type": " RsaVerificationKey2018 ",
"controller": "did:gsmc:123456789abcdefghi",
"signature": "0x3f2ead8dddb8b401d64a6f4a4a386d9f85d
a297f52fa81aeab23f515ed533aa8162c32f4e6f1347a9f805bb29d210cd99097c8d9232
}]
"authentication": [
// a relative DID URL used to reference a verification method above
“key-1”
}
```

그림 5. 플랫폼 DID document 구조
Fig. 5. Platform DID document structure

Claim(s)에는 객체의 ID 속성 정보가 Subject-Property-Value 방식으로 저장된다. 따라서 어떤 Subject에 대한 ID 속성이 포함되어 있는지, 그리고 해당 Subject가 어떤 Property를 가지고 있으며, Property의 값으로 어떤 value값을 취하는 것에 관한 정보를 포함한다. 마지막으로 Proof(s)에는 VC에 대한 진위 여부 검증에 필요한 값이 포함되는데 여기에는 다양한 암호 기법이 사용될 수 있다[12][13].

3.4 VC 구성요소

VC에 포함되는 데이터 항목은 @context, id, type, issuer, issuance Date, credential Subject, proof로 구성되어 있다. 플랫폼의 VC 구성요소는 그림 6과 같다. @context는 통신하는 서로 간의 정확한 통신을 위해 데이터가 어떤 값을 가지는지 정의를 내리는 역할을 한다. 플랫폼에서 사용한 @context 항목은 하나로, 해당 URL은 VC의 공식 콘텍스트가 명시된 위치를 가리킨다. W3C 표준에 따르면 VC 데이터를 원활하게 교환하기 위해서는 VC 공식 콘텍스트를 VC 내에 들어가야 한다. type 항목은 어떤 데이터가 사용될 것인지 명시하는 항목이다. 속성 중 Verifiable Credential의 역할은 VC는 공식 콘텍스트 내 정의된 VC 데이터 기본 구조에 따라 VC를 생성할 것이라는 의미이고, Project Contract Credential은 자체 제작한 콘텍스트 내 정의된 데이터 구조에 따라 프로젝트 및 체결에 필요한 데이터를 생성할 것이라는 의미이다. proof에는 공식 콘텍스트 내 정의된 RsaSignature2018 암호화 방식에 따라 검증 데이터를 생성할 것이라고 명시된다. id 항목은 VC 내 다양한 종류의 식별자가 들어갈 수 있다.


```
{
"@context": [ "https://www.w3.org/2018/credentials/v1" ],
"id": "https://us-central1-gsmc-b674f.cloudfunctions.net/vc-get/vcId", // vc url
"type": ["VerifiableCredential", "ProjectContractCredential"],
"issuer": "did:gsmc:0xcc68f4bbaf25b170e3e43b244ff0e2ced4ddeab6", // gsmc admin did
"issuanceDate": "2020-01-01T19:73:24Z",
"credentialSubject": {
"id": "did:gsmc:0x0000000000000000000000000000000000000000",
"project": {
"sn": "PROJECT_SERIAL_NUMBER",
"hash": "HASH_VALUE"
},
"contract": {
"sn": "CONTRACT_SERIAL_NUMBER",
"hash": "HASH_VALUE"
}},
"proof": {
"type": "RsaSignature2018",
"created": "2020-06-18T21:19:10Z",
"proofPurpose": "assertionMethod",
"verificationMethod": "gsmc-firebase-functions.com/vc-proof", // gsmc admin verify
"jws": "eyJhbGciOiJQUzI1NiIsImI2NCI6ZmFsc2UsImNyaXQiOlsiYjY
0Il19DJBMvvFAIC00nSGB6Tn0XKbb9XrsaJZREWvR2aONYTQQxnyXirtXnlew"
}}
```


그림 6. 계약 및 프로젝트 VC 구조 예시
Fig. 6. Examples of contracts and project VC structure

그림에서 볼 수 있는 VC 구조에서는 두 종류의 id가 사용되고 있다. 첫 번째로 나오는 id는 VC를 식별하기 위한 일련번호이다.VC에는 VC를 식별하기 위한 id가 존재한다. id에는 다양한 종류의 URI가 사용될 수 있는데, 첫 번째 명시된 id는 HTTP 기반의 URL이 사용되었고, 두 번째 명시된 id는 DID 기반의 URI가 사용되었다. 다음으로 issuer는 VC를 발행한 사람 혹은 기관을 뜻한다[12][13].

아래와 같이 DID 기반의 URI 형식으로 나타낼 수 있고, 다른 URI를 식별자로 사용할 수 있다. credential Subject는 Claim(s) 데이터를 포함하고 있는 항목으로 VC가 가리키는 객체의 실질적인 ID 속성이 포함되어 있다. 마지막으로 proof는 VC의 무결성을 검증할 수 있는 항목이다. proof는 VC를 발급할 때 필수적으로 포함되어야 하는 항목이다. VC의 무결성을 보장하기 위해 VC 정보에 대한 발행인의 서명이 proof 항목에 추가된다. proof에 포함된 verification Method 속성은 발행인의 공개키 등 디지털 서명을 검증할 수 있는 값이 위치하는 URL을 나타내고 jws 속성은 해당 VC 디지털 서명 값을 나타낸다. 예시와 같이, 계약 및 프로젝트 VC 뿐만 아니라 VC를 통해 원하는 정보로 목적에 맞게 VC를 만들 수 있다[13].

3.5 SSI 기반 인증 및 생성 시나리오

DID 사용자가 직접 DID를 생성해야 하는 것은 아니다. DID로 식별되는 사용자 대신 다른 사람이 DID를 생성하고 인증하는 것이 가능하다. 플랫폼에서 사용하는 DID는 당사자가 직접 DID를 생성하는 것이 아닌, 플랫폼에서 사용자의 DID를 생성하고 인증해주는 방식이다.

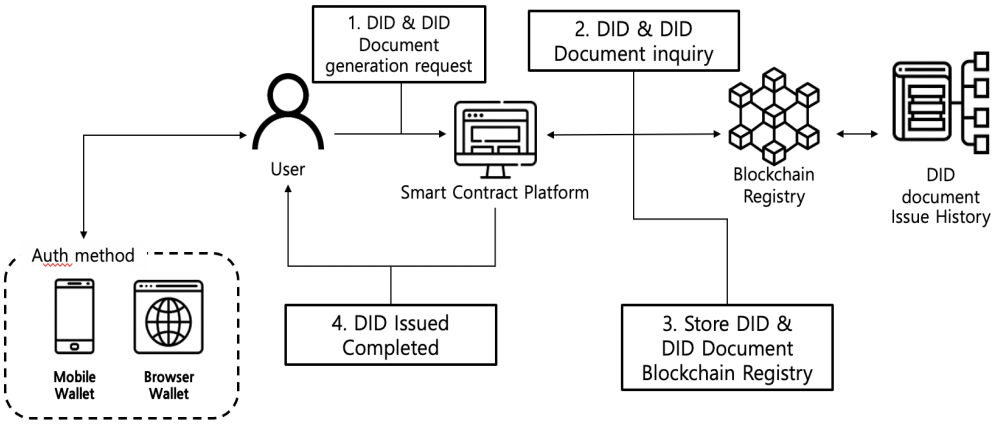

그림 7. SSI 기반 인증 및 생성 시나리오
Fig. 7. SSI-based authentication and creation scenarios

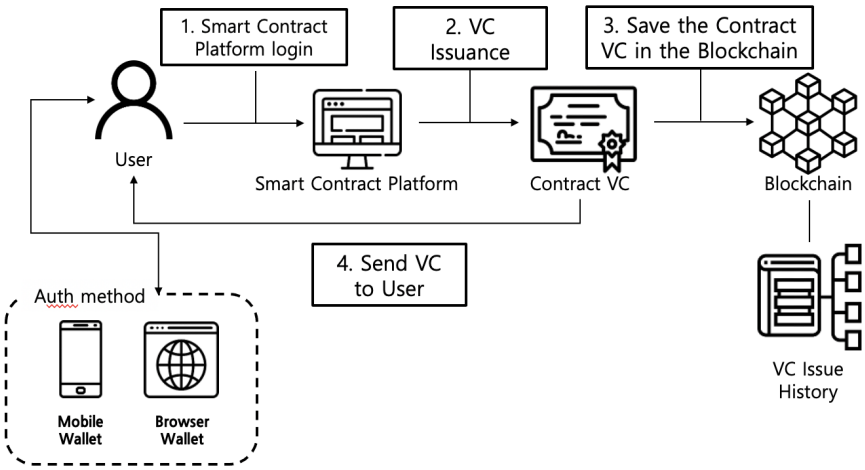

그림 8. SSI 기반 VC 발급 시나리오
Fig. 8. SSI-based verifiable credentials creation scenarios

플랫폼에서 DID 생성은 대신 해두더라도, DID 관리는 사용자가 직접 모바일 전자지갑 애플리케이션이나 브라우저 전자지갑을 통해 관리를 할 수 있다.

그림 7에서 사용자는 플랫폼 서비스에 DID와 DID document를 생성 요청한 후. 플랫폼 서비스에서 DID와 DID document를 생성을 완료하고 DID document는 DID method에 명시된 저장소에 저장한다. 그리고 사용자가 로그인 요청을 하고 DID에 명시된 저장소와 DID 내 Method-specific identifier가 가리키는 위치에 저장된 DID document를 통해 로그인을 가능하다. 그림 8은 기존의 생성된 DID를 통해 웹 플랫폼에 로그인을 하여 플랫폼에 VC 발급을 받는 과정이다. 여기서 VC의 기능은 플랫폼 서비스의 기능을 요청하거나 서비스 제공이 필요할 때만 일회성 통신으로 데이터를 교환하며 사용자 본인이 직접 데이터를 소유하고 관리하는 인증서이다.

Ⅳ. 결 론

본 논문에서는 신원인증 기술에 대한 분석과 기존 기술들의 취약점을 분석했다. 그리고 기존 신원인증 기술에서 토큰 가로채기나 인증기관의 보안 문제로 데이터가 쉽게 유출될 수 있기 때문에, 중앙화된 환경이 아닌 사용자 본인이 직접 데이터의 주체가 되어 데이터를 관리할 수 있으며 정보가 유출되어도 안전할 수 있는 방안을 제시했다. DID, DID document 및 VC 구조를 구축함으로서 DID라는 사용자가 직접 관리할 수 있는 식별 수단이 생성되고 DID document를 통해 식별자의 소유권을 증명할 수 있는 인증 수단이 포함되어 있으며 해당 DID document는 분산 원장 환경에 저장되어 있기 때문에 DID 정보가 유출되더라도 DID document에 기록된 개인키(서명) 인증을 통해 해당 DID의 소유권을 검증할 수 있기 때문에, DID가 유출되는 경우를 예방할 수 있다. 제안된 기법은 W3C의 공식 콘텍스트를 따르면서, DID 주소가 노출되더라도 타인이 정보를 탈취할 수 없으며, 적합한 정보를 가지고 목적에 맞는 VC를 생성할 수 있다. VC 역시 분산원장 환경인 블록체인 저장소에 기록되며 사전에 발급한 DID와 DID Document를 통해 VC의 소유권 확보가 가능하며 해당 VC를 사용하여 플랫폼 서비스를 사용할 수 있는 환경을 구축하였다.

향후 계획은 플랫폼 서비스에서 사용되는 VC를 사용자가 원하는 데이터만 추려 교환할 수 있는 VP(Verifiable Presentation)을 설계할 예정이며 해당 체계를 고도화하고 실제 구현까지 할 수 있도록 개발할 예정이다. 본 논문을 제시한 기법을 구현한 시스템을 보급함으로써 안전하게 신원인증을 할 수 있는 환경이 조성될 것으로 보인다.

References

[1] DongHee Kim and JinTak Choi, "A Study on the Efficient Authentication Management Technique of SSO Foundation", The Journal of KIIT, Vol. 48, No 3, pp. 55-63, Jun. 2006.

[2] YoungJae Maeng and DaeHun Nyang, "An Analysis of Replay Attack Vulnerability on Single Sign-On Solutions", Journal of KIISC, Vol. 18, No 1, pp. 103-114, Feb. 2008.

[3] Hwa-jeong Seo, Tae-hwan Park, and Ga-ram Lee, "Technology for Implementing Symmetric Key Cipher on Lightweight Internet of Things Platform", Journal of KIISC, Vol. 27, No. 6, pp. 15-20, Dec. 2017.

[4] Brian Trzupek, "How DIY PKI often negates the promise of public key encryption", Network Security, Vol. 2020, No 3, pp 14-17, Nov. 2020.

[5] Sungwon Cho, Hyunsang Choi, Gyu Heo, Sanghyun Cho, and Young-Gab Kim, "A System for SSL/TLS Vulnerability Detection of Servers", Journal of KIISC, Vol. 28, No. 1, pp. 145-153, Feb. 2018.

[6] Andreas Bogner, Mathieu Chanson, and Arne Meeuw, "A Decentralised Sharing App Running a Smart Contract on the Ethereum Blockchain", Association for Computing Machinery, Vol. 6, pp. 177-178, Nov. 2016.

[7] Chul-Jin Ki, "A Static and Dynamic Design

Technique of Smart Contract based on Block Chain", Journal of the Korea Academia-Industrial, Vol. 19, pp. 100-119, Jun. 2018.

[8] Alexander Mühle, Andreas Grüner, Tatiana Gayvoronskaya, and Christoph Meinel, "A survey on essential components of a self-sovereign identity", Computer Science Review, Vol. 30, pp. 80-86, Oct. 2018.

[9] Decentralized Identifiers (DIDs) v1.0 https://www.w3.org/TR/did-core/ [accessed: Nov. 01, 2020]

[10] A Primer for Decentralized Identifiers, https://w3c-ccg.github.io/did-primer/ [accessed: Nov, 01, 2020]

[11] Decentralized Identifier Resolution(DID Resolution) v0.2 https://w3c-ccg.github.io/did-resolution/ [accessed: Nov. 21, 2020]

[12] Yoon Dae-geun, "Self-Sovereign Identity Analysis Report", pp.80-98 Jul. 2020.

[13] Verifable Credentials Data Model 1.0 https://www.w3.org/TR/2019/REC-vc-data-model-20191119/ [accessed: Nov. 01, 2020]

저자소개

유 수 민 (Su-Min Yoo)

2019년 2월 : 울산과학대학교 컴퓨터정보학부 소프트웨어 개발전공(공학전문학사)
2019년 6월 ~ 현재 : (재)경주 스마트미디어센터 연구원
관심분야 : 블록체인, 인공지능

유 수 빈 (Soo-Bin Yoo)

2012년 2월 : 동국대학교 정보통신공학과(공학사)
2016년 8월 : 동국대학교 전자통신공학과(공학석사)
2016년 12월 ~ 현재 : (재)경주스마트미디어센터 주임연구원
2020년 3월 ~ 현재 : 동국대학교 전자통신공학과(공학박사)
관심분야 : 인공지능, 머신러닝, 데이터통신

조 정 화 (Jung-Hwa Jo)

2015년 2월 : 대구대학교 사회학과(사회학사)
2019년 6월 ~ 현재 : (재)경주 스마트미디어센터 연구원
2020년 9월 ~ 현재 : 동국대학교 전자통신공학과(공학석사)
관심분야 : 블록체인, 인공지능, 데이터통신

손 애 선 (Ae-Seon Son)

2018년 2월 : 금오공과대학교 컴퓨터공학과(공학사)
2019년 6월 ~ 현재 : (재)경주 스마트미디어센터 연구원
2020년 9월 ~ 현재 : 동국대학교 전자통신공학과(공학석사)
관심분야 : 블록체인, 인공지능