사용자 프라이버시 보호를 위한 BBS+서명 기법 기반 시뮬레이션 환경구축

윤태연[1], 이정륜[1]
[1]블록체인기술연구소
yoonty.ibct@gmail.com, martin@ibct.kr,

BBS+ Signature Environment Simulation for User Privacy Protection

Tae-Yeon Yoon[1], Jong-Ryun Lee[1]
[1]Institute of Blockchain Technology and Service

요 약

디지털 전환 시대를 맞아 일상생활 대부분이 온라인으로 이동하면서 온라인상에서 자신을 나타내는 신뢰할 수 있는 신분증의 필요성이 커지게 되었다. 신원 확인 방법은 중앙 집중식 모델에서 현재는 자기주권신원 모델로 변화하는 과정에 있으며 사용되는 핵심 기술은 탈중앙 식별자 DID(Decentralized Identifier)이다. DID는 기존 신원 체계와 달리 개인의 데이터 소유권을 개인에게 돌려줘 데이터 주권을 지킬 수 있게 해줌으로써 개인의 정보 공유 범위를 결정하는 SSI(Self Sovereign Identity)를 실현하는 기술이다. DID를 이용하면 데이터의 무결성, 투명성을 보장하는 자격 증명(Verifiable Credential, Verifiable Presentation) 발급이 가능하며 이를 검증하는 데이터는 모두 블록체인에 올라가 있는 것이 특징이다. 본 논문에서는 실제 서비스와 유사한 시뮬레이션 환경을 구축하여 자격 증명의 사용자 프라이버시를 보호하는 방법인 BBS+서명 기법에 대해 알아보고자 한다.

1. 서론

과거로부터 신원 인증은 본인확인을 위해 반드시 이루어져야 하는 과정이었다. 기존 인증 방식으로는 주민등록증, 운전면허증이 있었으며 1999년 전자서명법 제정으로 공인인증서가 국내 인터넷/모바일 뱅킹, 전자 정부 등 전자 상거래에서 신원확인 수단으로 본격적으로 활용되기 시작했다. 하지만 개인정보 보호의 중요성이 강조되면서 공인인증서의 폐지가 본격적으로 논의되기 시작했다[1]. 온라인상에서의 본인 인증 수단은 제 3의 기관을 거치지 않을 수 없었으나 현재는 자기주권 신원 모델이 등장하여 내가 “나”임을 증명하는 차세대 디지털 신원 인증 기술인 DID가 등장하게 되었다.

DID를 이용한 자기주권신원 모델은 3개의 참여자로 구성된다. Verifiable Credential(이하, 자격 증명) 발행자인 Issuer, 자격 증명의 소유자인 Holder, 자격 증명의 활용자인 Verifier로 구성되며 이들은 자기주권신원의 10원칙을 준수해야한다[2]. 그 중에서 최소화의 원칙(Minimalization)을 준수하기 위한 방법으로 0지식 증명(Zero Knowledge Proof), 선택적 노출(Selective Disclosure)이 있는데, 본 논문에서는 BBS+ 서명 기법을 이용한 선택적 노출에 대해 소개하고자 한다.

2. 자격 증명과 개인정보 보호 기법

전통적인 전자서명 기법을 통해 데이터의 무결성과 서명자에 대한 신뢰성을 확보할 수 있다. 하지만 이 방법은 원본 메시지 전체를 공개해야 검증이 가능하다는 점에서 불필요한 사용자 프라이버시가 노출될 수 있다는 문제가 있다. 0지식 증명은 이러한 프라이버시 문제를 해결하기 위한 기법 중 하나이다. 0지식 증명이란 사용자가 알고 있는 비밀값(secret)을 노출하지 않으면서 비밀값을 알고 있음을 증명하는 방법이다. 0지식 증명을 구현한 다양한 기법이 있으나 W3C국제 표준은 자격 증명의 0지식 증명의 도입 장벽을 낮추기 위해 CL 서명 기법 이외에 BBS+ 서명 기법 적용을 고려하고 있다[3].

2019년 하이퍼레저(Hyperledger)는 사용자의 신분 보호 및 프라이버시 보호를 위해 CL 서명 기법을 기반으로 한 Anonymous Credential1.0를 설계하였다[4]. CL 서명 기법은 Jan Camenisch와 Anna Lysyanskaya가 설계한 서명 기법으로 RSA비대칭키

알고리즘을 기반으로 한다. 하지만 CL 서명 기법의 여러 가지 장점에도 불구하고 다음과 같은 단점이 있다. 첫 째, 서명/검증에 사용하는 키의 크기가 너무 크다. 둘 째, 생성된 자격 증명의 크기가 너무 크다. 셋 째, 자격 증명의 상태 정보 관리에 사용되는 데이터의 크기가 너무 크다. 즉, 시간적으로나 자원적으로나 CL 서명 기법이 비효율적임을 의미한다[4]. 이에 따라 하이퍼레저는 CL 서명 기법 대신 BBS+ 서명 기법 도입을 고려하며 Anonymous Credential2.0을 설계하게 된다[5].

BBS+ 서명 기법은 BBS서명 기법에서 파생되었다. BBS 서명 기법은 서명자의 프라이버시를 보호하기 위해 처음 나온 개념이다[6]. BBS를 발전시켜 설계된 BBS+ 서명 기법은 BLS12-381과 같은 페어링 친화적 곡선(Pairing Friendly Curve)상의 키를 이용해 복수개의 메시지를 서명하여 단일 서명값을 생성하는 서명 기법이다[7,8]. BBS+ 서명 기법은 CL 서명 기법과 동일한 기능을 제공함(predicate 기능 제외)과 동시에, 키와 서명의 크기가 작고 연산 속도가 빠르다는 장점을 보유하고 있다. 표 1은 BBS+와 CL 서명 기법의 성능을 비교한 것이다[5].

	종류	BBS+ (BLS12-381)	CL (RSA)
크기	개인키	32Byte	256Byte
	공개키	96Byte	(771+256/message) Byte
	서명값	112Byte	672Byte
	증명값	368+(32/hidden message) Byte	696+74/message Byte
성능	키 생성	2.69ms	8.8sec
	서명 생성	1.43ms	93ms
	증명 생성	5.61ms	13ms
	검증 수행	4.61ms	11ms

<표 1> BBS+와 CL 서명기법 비교표

자격 증명의 표준을 준수하면서 선택적 노출을 지원하기 위한 방안으로 BBS+서명 기법과 링크드 데이터(Linked Data)가 함께 결합된 BBS+ Signatures2020의 초안이 공개되었다[9]. BBS+ Signatures2020를 사용하여 자격 증명에 선택적 노출 작업을 수행해도 첫 째, 자격 증명의 의미론적/구조론적 측면을 유지하면서 둘 째, 자격 증명의 무결성을 유지하면서 셋 째, 선택된 하위 속성값들만 포함하는 새로운 자격 증명에 대한 증명값을 생성할 수 있다. 예를 들어 경찰청에서 발행한 운전면허 자격 증명에 해당 기법을 적용하면 생년월일만 보여지는 새로운 형태의 운전 면허 자격 증명을 생성할 수 있으며 이는 경찰청의 공개키로 여전히 검증 가능하다.

JSON-LD(JSON-Linked Data)의 Frame연산을 이용하여 사용자가 원하는 구조의 자격 증명 구조를 생성할 수 있으며, BBS+ 서명을 이용해 다중 메시지 서명을 수행하여 발행 기관의 개인키 없이도 유효한 새로운 증명값을 만들 수 있게 된다. 다음 절에서는 BBS+ 서명 기법과 링크드 데이터를 결합한 방법을 이용한 시뮬레이션에 대해 설명한다.

3. BBS+ JSON-LD기반 자격 증명 시뮬레이션 환경 구축

전통적인 전자서명 기법과 달리 BBS+ 서명 기법은 선택적 노출을 이용한 사용자 개인정보 노출 방지를 목표로 한다. BBS+ 서명 기법은 첫 째, 복수개의 메시지에 서명하여 단일 서명값을 출력하고 둘 째, 복수개의 메시지 중 선택한 값만을 이용한 새로운 증명값을 생성한다. 아래의 수식은 BBS+에 따라 서명값과 증명값을 생성하는 방법을 기술 한 것이다.

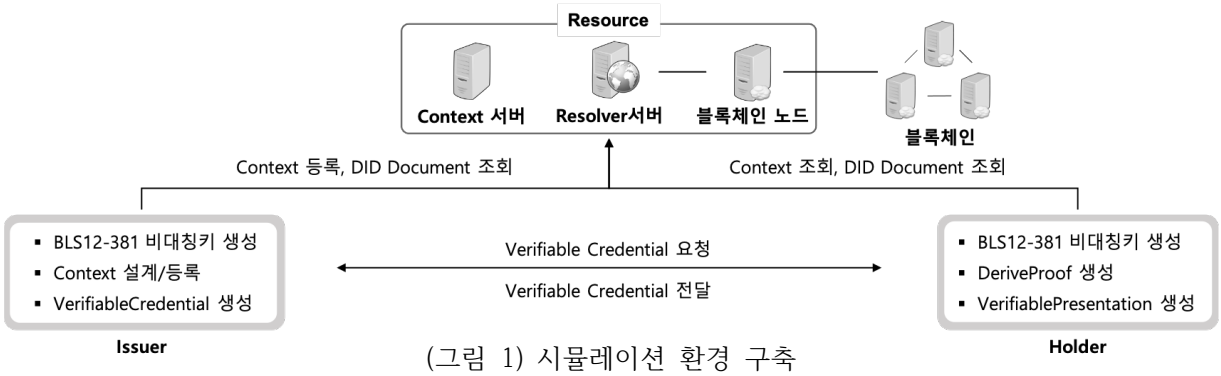
수식(1)에 따라 자격 증명 발행기관은 사용자 정보($claim_0$,$claim_1$,...,$claim_n$)를 모아 자신의 개인키($Issuer_{privKey}$)로 서명한다. *signature*는 n개의 사용자 정보에 대한 BBS+ 서명 값이다. 그 다음, 수식(2)에 따라 사용자는 공개할 정보($claim_x$,$claim_y$,...)를 선택하여 *signature*와 발행기관의 공개키($Issuer_{pubKey}$)를 연산하여 *proof*를 생성한다. 자격 증명을 검증하는 방법은 수식(3)과 같다. 선택된 사용자 정보($claim_x$,$claim_y$,...)와 *proof*를 발행기관의 공개키($Issuer_{pubKey}$)로 검증하면 참/거짓으로 유효성을 판단할 수 있다. *signature*는 원본 자격 증명에 포함되며 *proof*는 새로 생성한 자격 증명에 포함된다.

$$bbsSign([claim_1,claim_2,claim_3,...,claim_n])_{Issuer_{privKey}} = signature \quad (1)$$

$$bbsDeriveProof([claim_x,claim_y,...],\ signature)_{Issuer_{pubKey}} = proof \quad (2)$$

$$bbsVerify([claim_x,claim_y,...],\ proof)_{Issuer_{pubKey}} = True/False \quad (3)$$
$$(1 \le n),\ (1 < x \le n),\ (1 < y \le n)$$

BBS+서명 기법은 발행기관의 공개키를 이용하여 새로운 증명값을 만들 수 있다는 것이 특징이다. 서명값과 증명값 모두 동일한 공개키로 검증이 가능하다는 점에서, 발행기관이 발행한 원본 자격 증명을 사용자가 소유하고 있음을 증명하고 원본 자격 증명



(그림 1) 시뮬레이션 환경 구축

의 무결성이 보호됨을 알 수 있다.

자격 증명의 국제 표준을 준수하면서 BBS+서명 기법을 적용한 자격 증명을 발행하기 위해 그림 1, 표 2와 같은 시뮬레이션 환경을 구축한다.

분류	설명
블록체인 노드	DID Document를 등록한 블록체인 노드
Resolver 서버	DID를 이용해 DID Document를 조회하는 서버
Context 서버	자격 증명의 구조와 claim에 대한 정의를 명시한 서버로, JSON-LD 연산 시 필요한 Context를 조회하는 서버
Issuer	자격 증명을 생성하는 발행 기관
Holder	원본 자격 증명(Verifiable Credential)을 저장하고, 새로운 구조의 자격 증명(Verifiable Presentation)을 생성하는 사용자

<표 2> 시뮬레이션 참여 주체 역할

그림 1의 Resource는 Issuer, Holder가 모두 공통적으로 접근할 수 있는 자원을 의미한다. BBS+ 서명값을 검증하는데 필요한 공개키는 블록체인에 DID Document 형태로 등록하였으며, DID를 입력하여 Resolver서버에서 조회한다. DID Document에는 BLS12-381곡선에서 생성한 96바이트의 공개키를 등록한다. 자격 증명 생성에 사용될 컨텍스트(context)는 자격 증명의 스키마 구조를 의미하며, 본 시뮬레이션에서는 사용자에게 운전면허증을 발행한다고 가정하여 운전면허증의 속성값을 담는다. 본 논문에서 Resource에 대한 구현 방안은 생략한다.

BBS+기반의 운전면허증 자격 증명을 발행하기 위해 Mattr에서 개발한 라이브러리를 이용하여 시뮬레이션을 진행했다[10]. 발행 기관은 사용자에게 원본 자격 증명인 운전면허증을 표 3과 같이 발행한다. 표 3의 credentialSubject는 사용자 정보(Claim)들의 집합이며 원본 자격 증명에는 사용자에 대한 정보가 저장된 것을 볼 수 있다. 원본 자격 증명의 서명값은 proofValue로 표현되며 이를 검증하는데 사용되는 공개키는 verificationMethod에 기재되어 있다. 표 3의 proofValue는 수식(1)의 $signature$이다. 원본 자격 증명을 검증하기 위해서 Resolver서버를 통해 DID Document에 등록된 101번째 공개키($Issuer_{pubKey}$)를 조회한다. 표 3은 수식(1)이 적용된 자격 증명이다.


```
{
  "@context": […],
   "id": "https://issuer.police.go.kr/credentials/83627465",
  "type": ["VerifiableCredential", "DriverCredential"],
  "issuer": "did:lit:Aqu1wif79W85PYmbMTS8xQ",
  "credentialSubject": {
    "id": "did:lit:PCKeVwajdP7rWB7jwhpgWh",
    "type": "DriverLicense",
    "name": "마석대",
    "address": "서울 마포 상암 438",
    "driverId": "서울 11-18-174133-01",
    "registerNum": "890812-1111111",
    "renewalPeriod": "2024-12.31"
  },
  "proof": {
    "type": "BbsBlsSignature2020",
    "created": "2022-04-06T02:42:50Z",
    "proofPurpose": "assertionMethod",
    "proofValue": "tGcNG3JbUz/Zl7fTYQ9hk…",
    "verificationMethod":
"did:lit:Aqu1wif79W85PYmbMTS8xQ#101"
  }
}
```


<표 3> 발행된 원본 자격 증명

사용자는 월렛 또는 다른 저장소에 표 3의 자격 증명을 저장한다. 이후 사용자는 자격 증명을 필요로 하는 곳(예, Verifier)에 제출하기 위해 공개할 값을 선택하여 새로운 자격 증명을 생성한다. 본 시뮬레이션은 사용자가 자신의 이름을 공개하는 것을 가정한다. 표 4는 수식(2)가 적용된 새로운 자격 증명으로 이름만 공개된 것을 확인할 수 있다. 표 4 자격 증명의 proofValue는 수식(2)의 *proof*를 의미한다.


```
{
  "@context": […],
  "id": "https://issuer.police.go.kr/credentials/83627465",
  "type": ["DriverCredential","VerifiableCredential"],
  "credentialSubject": {
    "id": "did:lit:PCKeVwajdP7rWB7jwhpgWh",
    "type": "DriverLicense",
    "name": "마석대"
  },
  "issuer": "did:lit:Aqu1wif79W85PYmbMTS8xQ",
  "proof": {
    "type": "BbsBlsSignatureProof2020",
    "created": "2022-04-06T03:09:55Z",
    "nonce": "0t75Wn2ZnH93hpZxQOF…",
    "proofPurpose": "assertionMethod",
    "proofValue": "AA4+T4zzBOhel7i2riGW/…",
    "verificationMethod":
"did:lit:Aqu1wif79W85PYmbMTS8xQ#101"
  }
}
```


<표 4> 선택적 노출을 수행한 새로운 자격 증명

시뮬레이션을 통해서 표 3과 표 4의 자격 증명을 비교해보았다. 표 3과 표 4 자격 증명의 credentialSubject, proofValue값이 상이하나, 자격 증명을 검증하는데 사용되는 verificationMethod가 같은 것을 볼 수 있다. 표 4와 같은 새로운 자격 증명을 생성한 것을 통해 사용자가 원본 자격 증명을 보유하고 있음을 증명할 수 있고, 발행 기관이 사용자에게 원본 자격 증명을 발행했음을 의미한다. 또한 BBS+ 서명 기법을 통해 사용자의 개인정보 노출을 최소화하는 방안을 확인할 수 있었다.

4. 결론

본 논문에서는 DID기반 자격 증명의 개인정보 보호 방안인 BBS+서명 서비스 환경에 대하여 설명하였다. BBS+서명 기법을 사용하면 자격 증명을 활용하는 기관(예, 서비스 제공자)은 불필요한 데이터를 처리하는 서버 부하를 줄일 수 있으며 사용자 개인정보를 관리하는 부담도 줄어든다. 또한 필요한 사용자 정보만 담은 자격 증명을 제공하므로 사용자 개인정보 노출을 최소화할 수 있다. 하지만 DID기반의 자격 증명을 실생활에 도입하기 위해서는 자격 증명의 확장성 및 범용성을 가진 기술 개발이 요구된다. 따라서 어플리케이션, 자격 증명, 통신, 통합 DID플랫폼 등에 대한 추가적인 연구가 필요할 것으로 보인다.

참고문헌

[1] 정충식, 디지털 신원인증, 본인확인 수단의 변천 과정 분석, SPRI FOCUS, 2021

[2] C. Allen, “The path to self-sovereign identity,” 2016

[3] Verifiable Credentials Working Group, https://w3c.github.io/vc-wg-charter/

[4] hyperledger/indy-hipe, 2019, https://github.com/hyperledger/indy-hipe/tree/master/text/0109-anoncreds-protocol

[5] Sovrin Alliance, Webinar: Anoncreds 1.0 & 2.0 Update

[6] Boneh, Dan, Xavier Boyen, and Hovav Shacham. "Short group signatures." Annual international cryptology conference. Springer, Berlin, Heidelberg, 2004.

[7] Au, Man Ho, Willy Susilo, and Yi Mu. "Constant-size dynamic k-TAA." International conference on security and cryptography for networks. Springer, Berlin, Heidelberg, 2006.

[8] Camenisch, Jan, Manu Drijvers, and Anja Lehmann. "Anonymous attestation using the strong diffie hellman assumption revisited." International Conference on Trust and Trustworthy Computing. Springer, Cham, 2016.

[9] BBS+ Signatures 2020, http://w3c-ccg.github.io/ldp-bbs2020/

[10] mattrglobal/jsonld-signatures-bbs, http://github.com/mattrglobal/jsonld-signatures-bbs