BBS+ 서명을 이용한 선택적 공개 과정의 검증에 대한 연구

황진주, 김근형*

동의대학교 응용소프트웨어공학과,

*동의대학교 게임공학과

mfdo7722@gmail.com, *geunkim@deu.ac.kr

A Study on Verification of Selective Disclosure Process based on BBS+ Signature

Jin-Ju Hwang, Geun-Hyung Kim*

Department of Applied Software Engineering, Dong-eui University

*Department of Game Engineering, Dong-eui University.

요 약

사용자 검증을 위해 정보를 제시하는 과정에서 불필요한 개인 정보를 제외하는 것은 사용자의 개인정보보호 측면과 자원 활용의 효율에서도 중요하다. 본 논문에서는 자격 증명의 서명과정에 BBS+ 서명을 도입하고, 자격증명 소유자의 선택적 공개를 이용한 증명과정을 보인다. 또한, 시스템이 데이터의 위변조 상황에 대해서도 올바른 검증을 도출해냄을 실험한다.

Ⅰ. 서 론

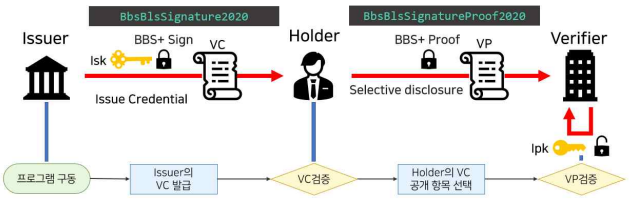

그림 1. 선택적 공개를 이용한 검증 과정

선택적 공개를 이용한 검증 과정의 참여자는 VC(Verifiable Credential)를 발급하는 Issuer, 자격 증명의 보유자이자 제출자인 Holder, Holder가 생성한 VP(Verifiable Presentation)를 검증하는 Verifier로 구성된다. Issuer는 Holder에게 VC를 발급하고, Holder는 VC를 통해 Verifier가 요구하는 검증 속성을 모아 VP로 만들어 제출한다. Verifier는 Holder가 제출한 VP를 검증한다.

검증의 큰 틀은 비대칭키 방식을 따른다. Issuer는 자신이 발급한 VC에 대해 자신의 개인키로 서명하여 데이터의 무결성을 검증하도록 한다. Holder는 자신이 생성한 VP에 대한 Proof를 생성해 값을 검증하도록 한다. Verifier는 Issuer의 공개키를 이용해 VP에 대한 검증을 수행한다.

검증을 위한 공개키는 탈중앙화된 식별자인 DID (Decentralized Identifier)를 통해 블록체인에서 획득한 DID document에서 얻을 수 있다. Isssuer, Holder, Verifier는 자신의 고유 DID를 갖는다.

BBS+(Boneh-Boyen-Shacham) 서명은 영지식 증명 속성인 선택적 공개를 제공한다[1]. 선택적 공개를 이용한다면 Holder는 VC 내부 속성 중 일부만 선택해 VP로 만들어 Verifier에게 제출할 수 있다[1]. BBS+ 서명은 영지식 증명을 제공하는 다른 서명에 비해 작은 Key와 증명값을 가져 작은 비용의 거래를 요구하는 블록체인 환경에 적합하다[2].

본 논문에서는 선택적 공개를 이용한 증명 과정을 보인다. 이후 제작한 시뮬레이터를 이용해 데이터 위변조 상황과 같은 이상 상태에 대해 올바른 처리를 수행하는가에 대해 실험한다.

Ⅱ. 데이터 구조

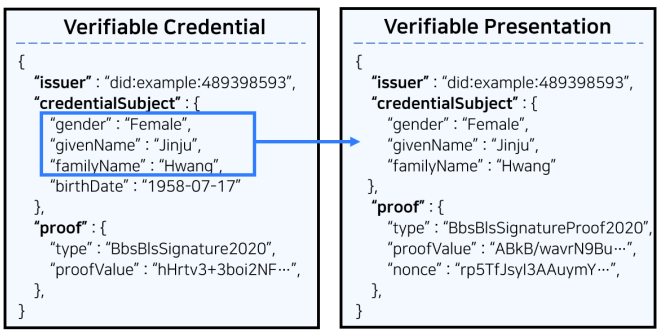

그림 2. VC와 VP의 구조

그림 2는 VC, VP의 데이터의 단순화된 구조를 보인다. 참여자 간 전달되는 모든 데이터는 JSON-LD 형태로 W3C의 데이터 무결성 규약을 따른다[3].

시뮬레이터가 구동되면 정의된 속성 정보 파일을 참조해 DID document를 생성한다. 발급 과정에서 Issuer가 VC에 자신의 개인키를 이용하여 서명하면 "proof" 항목에 증명 방식과 서명 값 등이 추가된다. "type"은 증명 방식이 기입되고 "proofValue"는 해당 방식을 통해 생성된 검증을 위한 실질적인 값이 들어간다.

Holder의 개인 정보는 "credentialSubject"에 담겨 발급된다. Holder가 이 값을 Verifier가 요구하는 항목만을 선택해 VP를 제시하는 것이 선택적공개이다. 그림 2에서 VC에 선택적 공개가 수행되어 "credential Subject"의 속성이 변화됨을 볼 수 있다. Holder가 Proof를 생성하면 Issuer 증명을 포함하여 proofValue에 담긴다. "Issuer"에는 VC를 발급한 Issuer의 DID를 담아 Verifier가 명시된 Issuer의 DID를 통해 공개키를 획득하도록 한다.

Ⅲ. 검증 테스트

시뮬레이터를 이용한 검증 과정의 흐름은 그림 1의 과정과 동일하다. 다만, 테스트에서는 각 참여자 간의 자격 증명 전송 과정에서의 데이터를 조작하여 이상상태를 발생시키고 그 대응을 확인한다.

데이터 조작은 데이터 무결성, 발급 기관 검증을 기준으로 자격 증명의 값을 변화시켰다. 데이터 무결성은 변질, 누락, 조작 시기에 차이를 두어 확인한다. 각 속성에 서명이 되어 속성이 누락될 때 서명이 함께 누락되며 검증을 통과하는 경우에 대한 대응을 확인하기 위해 변질과 누락을 구분하였다.

변질이란 자격 증명의 “givenName”요소의 값이 “Jinju”에서 “Chacha”로 변경되는 것처럼, 내용이 변화된 경우를 의미한다. 누락이란, “given Name”요소 자체가 내용 자체에서 사라지는 상황을 의미한다. 조작 시기는 Holder의 Proof 생성을 기준으로 한다. 변질/누락된 데이터에 Proof가 생성된 경우와 Proof 생성 이후 변질/누락된 경우를 Proof를 통해 값의 인위적 변화를 알아챌 수 있음을 확인하기 위함이다. 따라서, 검증하고자 하는 상황은 아래와 같다.

가. 데이터 무결성 검증
- case1 : Issuer가 발급한 VC의 요소가 변질/누락되어 전달된 경우
- case2 : Holder의 Proof 생성 전 VC 요소를 변질/누락시킨 경우
- case3 : Holder의 Proof 생성 후 VC 요소를 변질/누락시킨 경우

나. 발급 기관 검증
- case4 : VC의 Issuer가 존재하지 않는 경우
- case5 : VC의 Issuer가 존재하는 다른 Issuer로 변질된 경우

Ⅳ. 검증 결과

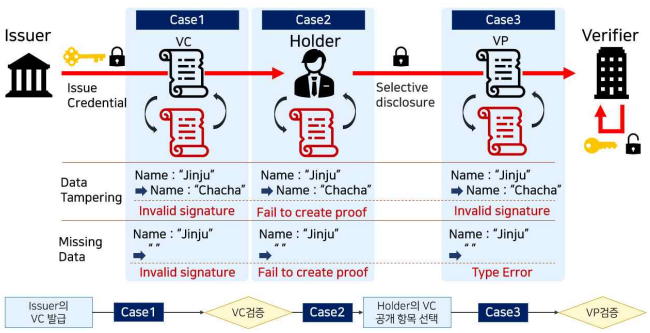

그림 3. 무결성 검증 시기와 변형 방법에 따른 결과

case1의 검증은 Issuer가 Holder에게 VC를 발급한 이후 VC의 “givenName” 요소를 “Jinju”에서 “Chacha”로 변질시켰다. 그 결과 검증 함수에서 “Invalid signature” 오류를 도출해 데이터 변질에 응답하였다. “givenName” 요소 자체를 삭제한 누락 테스트의 경우에도 동일한 대응을 하였다. 따라서 발급된 VC에 대해 데이터가 변질/누락된 경우에 Holder는 VC가 유효하지 않다는 것을 확인할 수 있다.

case2는 Holder의 속성 선택 이전에 변질/누락을 적용하였다. Holder의 Proof 생성이 영향을 미치기 전의 데이터 조작에 Verifier의 대응을 알아보기 위함이다. 그 결과 두 상황 모두 Proof 생성 과정의 중의 연산에서 “Fail to create proof” 에러가 발생한다. 따라서 Proof가 생성되지 않아 VP 제출 자체가 불가하게 된다.

case3에서는 Holder의 Proof 생성 이후 조작된 VP에 대해 Verifier의 대응을 확인한다. 데이터를 변질한 결과에서는 case1과 동일한 “Invalid signature” 오류를 도출한다. 누락의 경우는 Verifier의 Proof Request 요구사항을 충족하지 못하기 때문에 Type Error를 일으킨다.

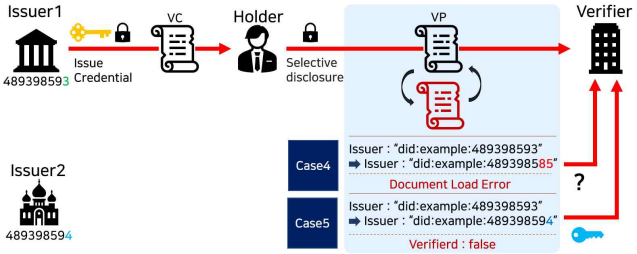

그림 4. Issuer의 유효성에 따른 결과

발급 기관의 검증은 Issuer의 유효성과 검증 시의 공개키의 활용 여부를 확인하기 위해 시행된다. case4는 VC를 발급한 Issuer가 더 이상 유효하지 않거나, 잘못 기입된 경우에 대한 검증이다. 테스트를 위해 VP에서 issuer 속성에 기입된 DID 번호 끝자리를 “93”에서 존재하지 않는 번호인 “85”로 변경했다. 해당 VP로 검증 시, 변질된 DID를 통해서는 DID Document를 로드 할 수 없어 검증에 실패한다.

case5는 verifier의 검증 시, Issuer의 공개키 참조 여부를 확인한다. Issuer가 하나인 이전과 달리 VC를 발급할 Issuer1과 임의의 Issuer2가 존재한다. case4는 존재하지 않는 Issuer의 DID에 접근했기 때문에 DID Document 로드 문제가 발생했다. 하지만 case5에서는 존재하는 다른 임의의 Issuer DID에 접근해 해당 문제를 해결한다. Issuer1에게 발급받은 VC로 생성된 VP를 조작한다. VP의 “issuer” 속성과 “proof” 속성에 기입되는 DID 번호의 끝자리를 “3”에서 Issuer2의 DID인 “4”로 변경하여 검증을 요청한다. 이 경우 Verifer는 변경된 Issuer2의 DID로 접속해 Issuer2의 공개키로 VP를 검증하게 된다. 그 결과 검증 성공 여부를 알리는 “verified” 속성이 “false”로 도출되며 검증에 실패한다.

Ⅴ. 결론

본 논문에서는 BBS+ 서명을 이용한 선택적 공개 과정에서의 데이터 무결성과 발급 기관의 유효성을 확인했다. 선택적 공개를 활용해 제시 과정의 불필요한 개인 정보 전달을 제한했으며, 다른 영지식 증명에 비해 작은 공간을 요구하는 BBS+ 서명을 활용해 자원 소모를 줄였다. 이러한 성능의 확장은 블록체인을 이용한 검증 기술의 실질적 도입에 이바지할 수 있을 것으로 보인다.

ACKNOWLEDGMENT

이 논문은 2021년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(No. NRF-2021R1F1A1047573).

참 고 문 헌

[1] J. Doerner, Y. Kondi, E. Lee, a. shelat and L. Tyner, "Threshold BBS+ Signatures for Distributed Anonymous Credential Issuance," in 2023 2023 IEEE Symposium on Security and Privacy (SP) (SP), San Francisco, CA, US, 2023 pp. 2095-2111. doi: 10.1109/SP46215.2023.00120

[2] T. Looker, V. Kalos, A. Whitehead, and M. Lodder, “The BBS Signature Scheme”, IETF draft-irtf-cfrg-bbs-signatures-01, October 2022

[3] Longley, D. Sporny, M. "Verifiable Credential Data Integrity 1.0" 2022,(https://w3c.github.io/vc-data-integrity).doi:10.1109/SP46215.2023.00120