탈중앙화 환경을 위한 IoT 디바이스 간 통신 메커니즘 설계 및 구현

최규현, 김근형

동의대학교

giry8647@gmail.com, geunkim@deu.ac.kr

Design and Implementation of Communication Mechanism between IoT Devices for Decentralized Environment

Gyu-Hyun Choi, Geun-Hyung Kim

Dong-eui University

요 약

데이터의 자기주권을 위해 탈중앙화된 데이터의 관리 체계가 필요하다. 최근 표준화된 DID(Decentralized Identifier)는 신원 데이터의 자기주권을 가능하게 한다. 탈중앙 환경에서 DID를 사용하는 엔티티간 상호 안전한 통신 채널을 확보할 수 있는 DIDComm도 표준화 중이다. 본 논문은 IoT 디바이스의 식별자로 DID를 적용하고 DIDComm을 사용한 피어 간 탈 중앙화된 통신 채널을 확보하고 통신하는 메커니즘을 제안하고 구현한 결과를 보인다.

Ⅰ. 서 론

현재 블록체인 기술을 활용한 다양한 기술이 개발되고 있다. 그 중 블록체인에 인증 기술을 활용한 Decentralized Identifier(이하 DID)는 자기주권 신원인증을 구현하기 위해 개발된 기술로 제 3자의 간섭 없이 본인의 신원을 증명한다. DID를 사용하면 중앙 집중형 신원관리 시스템이 가지는 문제를 해결할 수 있으며 자신의 신원정보를 스스로 관리하고 사용할 수 있다[1]. DID 기술은 신원증명 기술로 다양한 요소들에 접목시킬 수 있으며 IoT 센서기술에도 적용이 가능하다. IoT 기술은 기존에 있는 다양한 사물 및 센서에 통신 기능을 넣어 인터넷에 연결시키는 기술이다. 이러한 기술은 우리가 다양한 사물들을 통신으로 제어할 수 있게 해주며 각각의 사물들을 연결시켜 상황에 맞게 자동적으로 동작한다. 본 논문은 탈중앙 환경에서 IoT 디바이스의 데이터를 전송하기 위해 DIDComm 메커니즘을 설계 및 구현하였다.

Ⅱ. DID와 DIDComm

DID는 W3에서 표준화된 기술로 중앙기관을 통한 신원 인증 방식의 문제점을 해결하는 탈중앙화된 인증 기술이다. DIDComm은 DID를 사용한 통신을 위해 만들어진 기술로 제 3자를 통하지 않은 직접적인 통신을 제안하고 있으며 메시지 암호화 및 복호화, 서명에 대한 기준을 제시하고 있다. DID 및 DIDComm에 대한 자세한 내용은 해당 문헌에 자세히 나와있다[1][2]. DIDComm은 아직 완전한 표준화가 진행되지 않았으며 최근 버전 2가 나왔다. 기존의 버전 1은 Peer To Peer 통신 최적화를 위해 단일 DID 통신에 최적화된 기능을 제공하였다. 그러나 이러한 방식은 메시지 통신 방법에 따라 DIDComm을 사용하는 방법이 명확하지 못했으며 이를 위해 단일 DID 이외의 통신에서도 사용하기 쉽도록 개선되었다. 이외에도 라우터를 통한 전송 시 속성 추가, 특정 상황에 따른 프로토콜의 추가 등 일부 변경사항이 있다[3]. DIDComm의 표준화는 아직 진행 중이며 이후에 또 다른 변경사항이 추가될 수 있다.

Ⅲ. 테스트

IoT 디바이스들의 특성 상 디바이스에서 발생하는 데이터들을 관리하기 위해 외부로 통신을 하게 되고 이는 필연적으로 본인의 센서 데이터가 노출될 수 있다. 본 논문은 위에서 언급한 문제를 해결하기 위한 방법으로 DID 및 DIDComm을 사용하였다. 이전의 연구를 통해 DID 및 DIDComm의 활용 가능성을 확인하였으며 이를 IoT 디바이스에 적용하기로 하였다[4][5]. IoT 디바이스에 DID를 적용시킬 경우 디바이스에서 발생하는 데이터의 제어를 사용자가 스스로 할 수 있다. IoT 디바이스에 DID를 적용시킬 경우 다음과 같이 동작할 수 있다. 각각의 디바이스에 DID를 매핑시켜 정보를 저장하고 있으며 다른 IoT 디바이스에 데이터를 전달할 때에 생성된 DID를 기반으로 DIDComm을 진행하여 안전하게 데이터를 전달한다. 이러한 방식은 각각의 디바이스들이 DID를 소유하여 사용자가 이를 컨트롤 할 수 있게 도와주며 제 3자를 거치지 않기 때문에 데이터의 보안성을 확보할 수 있다.

표 1. 개발 환경

속성	설명
IoT 디바이스	라즈베리 파이 4
센서	터치 센서, 초음파 센서
사용 언어	Java
사용 패키지	didcomm-jvm(DIDComm)[6], pi4j(Java 기반 센서 제어)[7]

개발 환경은 표 1과 같으며 테스트 환경은 그림 1과 같다. 그림 2는 실제

구현한 IoT 디바이스의 사진으로 왼쪽이 터치센서, 오른쪽이 초음파 센서이다.

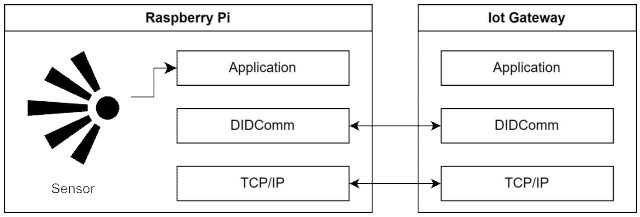

그림 1. 테스트 환경

그림 2. IoT 디바이스 구현 사진

해당 테스트의 경우 IoT 디바이스 및 다른 IoT 디바이스가 각각 DID 및 DIDDoc를 가지고 있으며 상대방의 DID 및 DIDDoc를 가지고 있다. 다음은 실제 테스트 동작이다. 각각의 IoT 디바이스를 실행시켜 통신이 가능한 환경을 만든다. 이후 센서를 가동시켜 데이터를 가져온다. IoT 디바이스는 주기적으로 센서 데이터를 가져와 다른 IoT 디바이스에게 보내게 되며 이때 DIDComm을 사용한다. 전송을 위해 가지고 있는 상태 정보를 토대로 그림 3과 같은 일반 메시지로 만들게 되며 해당 형식은 DIDComm의 일반 메시지 형식을 따른다. 이후 DIDComm을 사용한 암호화를 진행하며 이때 다른 IoT 디바이스의 DIDDoc에 있는 공개키 정보를 활용하여 암호화를 진행한다. 그림 4는 이를 통해 만들어진 암호화 메시지이다.

didcommMessage : {"id":"507917","typ":"application\/didcomm-plain+json","type":"http:\/\/example.com\/protocols\/lets_do_lunch\/1.0\/proposal","from":"did:iot:iot_device_touch_1","to":["did:iot:gateway"],"created_time":1673331699439,"expires_time":1706943600,"body":{"message":"{\"controller\":\"did:deu:alice1234\",\"id\":\"did:iot:iot_device_touch_1\",\"body\":\"Touch!\"}"}}

그림 3. 암호화 전 일반 메시지

encrypted message : {"ciphertext":"ZqY6xDP75_Olh4Zc9Xf6F04mAmJr0CSztEYRFwlNPfHy0f_R1Mdw7ck6x-vYjUF5bkeunPSgFB8WH5jbtYiK_Hs-6cCqhmpxURVngFwYcLz8a6RSQcf6xOAUUTZKcb5dBBlL86fzwoyQ0kpjZvXw_bPFH5Peyst6_0uC3hA3s4lo-eAFEsfgrtEdYO9BQ0G0yC-1orWwRI8N6fF5_DrOOnrDq2AC3wXrouig1T9lyhB3pPTotSybgH9EXXaodRwUAJYK9s4wBff0Kq5B7tZ1STKoKyuMa3pDTXJtb1xaILZIn3jeVlIAmJ3PFDUUPRRM1196sqPrVivNCbA8E7yDRz7UbfTk57FCPx-OsWkYD4rcAktHK1V_exTWpuaFCIQzKyrRGcrjkmt0SzQ5-yfat_CoemL4PnqgpA_G3KFrqi6SsNaaDI7ImY-ieNlE7sjPj-9T6vAy7WwpbwYOS2utGmzTQ6XwE_h5nA8oTMASgNg","protected":"eyJlcGsiOnsia3R5IjoiT0tQIiwiY3J2IjoiWDI1NTE5IiwieCI6IjB6MlNQRVRTM05xWGhLUi1rVFpqYkN2T3RZX1RnQTBmeGM5WUlSZVJHQnMifSwiYXB2IjoiRWtMOTdRTUVUazVmWHJMRzktd2RabkR3TVFhbkZ6V2NJMXI3YXQwYlRZRSIsInNraWQiOiJkaWQ6aW90OmlvdF9kZXZpY2VfdG91Y2hfMSNrZXktMiIsImFwdSI6IlpHbGtPbWxzZERwcGIzUmZaR1YyYVdObFgzUnZkV05vWHpFamEyVjVMVEkiLCJ0eXAiOiJhcHBsaWNhdGlvblwvZGlkY29tbS1lbmNyeXB0ZWQranNvbiIsImVuYyI6IkEyNTZDQkMtSFM1MTIiLCJhbGciOiJFQ0RILTFQVStBMjU2S1cifQ","recipients":[{"encrypted_key":"fXAcPiDQIvobq5RF8Qs4JT7TgsNwlneGspWsAtdwBjNd5x7sMYcdo6yeXZxQH79PqmUL2riyay6IunmysnAHQ2nmnJDCx6Wv","header":{"kid":"did:iot:gateway#key-2"}}],"tag":"JmH1_T2sDQCW-lTM2b9TuMmjE3xG-FkExjA7AlD2flI","iv":"D7fTe57s-gtXlaZdea7FMw"}

그림 4. 암호화 메시지

암호화 메시지가 만들어지면 이를 다른 IoT 디바이스에게 전달한다. 다른 IoT 디바이스는 전달받은 암호화 메시지를 DIDComm을 통해 복호화하며 최종적으로 그림 5와 같은 복호화 메시지를 확인한다.

decrypt message : {"id":"507917","typ":"application\/didcomm-plain+json","type":"http:\/\/example.com\/protocols\/lets_do_lunch\/1.0\/proposal","from":"did:iot:iot_device_touch_1","to":["did:iot:gateway"],"created_time":1673331699439,"expires_time":1706943600,"body":{"message":"{\"controller\":\"did:deu:alice1234\",\"id\":\"did:iot:iot_device_touch_1\",\"body\":\"Touch!\"}"}}
message : {"message":"{\"controller\":\"did:deu:alice1234\",\"id\":\"did:iot:iot_device_touch_1\",\"body\":\"Touch!\"}"}

그림 5. 복호화 메시지

터치하지 않은 상태일 경우에도 위와 동일한 수순을 통해 메시지가 전달되며 다른 IoT 디바이스에서 동일하게 메시지를 확인할 수 있다.

Ⅳ. 결론

DID를 사용한 IoT 디바이스는 사용자가 관리할 수 있다는 장점을 지닌다. 그리고 DIDComm을 통해 디바이스 간의 안전한 통신도 가능하다. 본 논문은 이러한 IoT 디바이스에 DID를 활용한 DIDComm을 적용시켜 테스트하는 것으로 사용자가 데이터를 직접 제어할 수 있는 IoT 디바이스를 확인했다. 이후 해당 기술들을 사용해 사용자가 데이터 주권을 가지는 IoT 환경을 연구할 예정이다.

ACKNOWLEDGMENT

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

참 고 문 헌

[1] M. Sporny, D. Longley, M. Sabadello, D. Reed, O. Steele and C. Allen, “Decentralized Identifiers (DIDs) v1.0 Core architecture, data model, and representations,” W3C PR. Aug. 2021.

[2] Decentralized Identity Foundation, “DIDComm Messaging,” [Internet], https://identity.foundation/didcomm-messaging/spec/.

[3] Decentralized Identity Foundation, “What’s New?,” DIDComm V2 Guidebook, [Internet], https://didcomm.org/book/v2/whatsnew/.

[4] 최규현, 김근형. "자기주권 신원 생태계를 위한 신뢰할 수 있는 통신 방법." 정보처리학회논문지. 컴퓨터 및 통신시스템 11.3 (2022): 91-98.

[5] 최규현, 김근형. "DIDComm 메시지의 암호화 및 서명 알고리즘." 한국멀티미디어학회 추계학술발표대회 논문집. 25.2 (2022): 122-125.

[6] Decentralized Identity Foundation, “didcomm-jvm,” Github, [Internet], https://github.com/sicpa-dlab/didcomm-jvm.

[7] Pi4J, “pi4j-v2,” Github, [Internet], https://github.com/Pi4J/pi4j-v2.