2022년도 한국멀티미디어학회 추계학술발표대회 논문집 제25권 2호

DIDComm 메시지의 암호화 및 서명 알고리즘 분석

최규현*, 김근형*
*동의대학교 게임공학
e-mail : giry8647@gmail.com, geunkim@deu.ac.kr

An Analysis of Encryption and Signature Algorithms for DIDComm Message

Gyu-Hyun Choi*, Geun-Hyung Kim*
*Game Engineering Major Dong-eui University

요 약

DIDComm은 DID와 DPKI를 기반으로 안전한 통신 채널을 설정하는 것을 목적으로 만들어진 기술이며 DID 문서를 통한 공개키 배포 및 메시지 암호화로 안전한 메시지 전달 및 개인 키로 인한 서명 또한 가능하다. 이러한 DIDComm의 사용을 위해 구현해야 할 기능들을 분석하며 DIDComm이 제공하는 패키지의 사용 및 키 호환성 테스트를 통해 DIDComm 모듈이 제대로 작동하는지를 테스트한다.

1. 서론

최근 블록체인을 활용한 다양한 기술들이 연구되고 있으며, 그 중 이미 실용화되어 우리가 모르는 사이에 사용중인 블록체인 기술들 또한 있다. 이 중 분산 식별자라고 불리는 Decentralized Identifier(이하 DID) 또한 블록체인 기술 중 주목받고 있는 기술이다. DID는 기존의 중앙에서 인증받아 ID를 발급받는 형태에서 벗어나 본인 스스로 ID를 만들어 인증받는 형태로 바꿀 수 있게 해주며, 이는 특정 중앙에 개인 정보가 몰리는 상황 및 개인 스스로가 개인 정보를 관리할 수 있는 상황을 만들어 주어 앞으로 다가올 정보화 사회에 매우 유용한 기술이다[1].

DIDComm은 이러한 DID를 통신에 활용하여 높은 보안을 가진 통신이 가능하도록 개발된 기술이다. DIDComm은 아직 개발 중이며 상용화되어 있지 않으나 앞으로 다가올 탈중앙화 시대에 사용하기 좋은 기술이 될 것이다.

2. 관련 기술

2.1 DID

DID란 자기주권신원 증명을 가능하게 해주는 기술이며 W3C에서 표준화한 식별자이다. 분산 식별자를 사용하면 중앙 기관에 상관없이 자유롭게 식별자를 만들 수 있으며, 신원 증명 과정에서 제3 자의 개입 없이 증명할 수 있다[1]. 이는 분산 식별자의 인증을 위한 비 공개키 쌍과 분산 네트워크 때문에 가능하다. 사용자가 분산 식별자를 만들면 비 대칭키 쌍이 만들어지며 개인의 지갑에 저장된다. 여기서 지갑은 개인이 소유한 기기로 본인 핸드폰 또한 개인 컴퓨터를 말한다. 다음으로 앤드 포인트를 만드는데, 본인 인증을 위한 통신 방법을 위해 만든다. 이렇게 만들어진 일부의 정보를 분산 네트워크에 저장한다. 분산 네트워크에 저장된 정보들은 분산 식별자를 통해 조회할 수 있으며 이를 DID 문서라고 부른다. DID 문서는 DID를 통해 조회할 수 있으며 해당 정보를 통해 인증을 진행하게 된다.

2.2 JWT 및 JWK

JSON Web Token(이하 JWT)은 통신을 위해 사용하는 토큰으로 JSON 형식의 Web 토큰이다. JWT는 사용에 따라 JSON Web Signature(이하 JWS)와 JSON Web Encryption(이하 JWE)으로 나뉜다. JWS는 서명이 들어간 토큰으로 데이터에 대한 무결성과 토큰 발행자의 인증을 위해 사용된다. 데이터의 무결성 및 인증은 보내려는 데이터를 해시 함수에 적용 및 개인키로 암호화하여, 기존의 데이터와 함께 보내는 것으로 확인할 수 있다[2]. JWE는 암호화가 들어간 토큰으로 데이터의 유출을 막기 위해 사용한다. 송신자는 수신자의 공개키를 사용해 메시지를 암호화하는 것으로 수신자의 개인키로만 복호화가 가능하도록 만든다. 이는 제3 자에 의해 메시지가 탈취당하더라도 암호화에 사용한 공개키와 쌍을 이루는 개인키를 소유하지 않을 경우, 복호화를 할 수 없기 때문에 메시지의 정보를 확인할 수 없다[3]. JSON Web Key(이하 JWK)는 위와 같은 JWT을 위해 사용하는 키들을 JSON 형식으로 표준화했으며 JWT 사용 외에도 키 정보 전달을 위해 사

용한다[4]. <표 1>은 JWK에 대한 형식을, <그림 1>은 JWK의 예시 그림이다.

표 1. JWK 형식

속성	설명
kty	해당 키를 사용할 알고리즘
d	개인키 정보
crv	사용한 타원 곡선
x, y	공개키 정보, 사용한 곡선에 따라 y 값이 없을 수도 있다.


```
{
"kty":"EC",
"d":"tSANmJSE49KEUj9yvU8gUYJtiIBkajk9D3j_HetQHA4",
"crv":"P-256",
"x":"t2mj5_LHQ4twBgYH8HNfVW3DOHIJT76pfCmErFWxEQY",
"y":"CpD7E-1u2v6DGUebsRR0ofyCbLEpzkLhoNwnT2Dwvlg"
}
```


그림 1. JWK의 예시

3. DIDComm

3.1 DIDComm 표준

DIDComm이란 DID 시스템에서 안전한 개인 통신을 제공하기 위해 등장했다. DIDComm은 DID 시스템을 활용한 통신 메커니즘을 제공하며, 이는 분산된 환경에서도 안전한 통신이 가능하다[5]. DIDComm을 사용하기 위해선 분산 식별자가 필요하며, 기존에 만들어진 키 쌍을 활용해 메시지를 암호화 및 서명하는 것으로 메시지의 탈취로 인한 정보 유출을 막거나 메시지의 무결성 및 인증 등에 활용한다. DIDComm의 메시지들은 JWT의 형식을 따르고 있으며, 암호화 및 서명을 위한 키 또한 JWK의 형식을 따른다. DIDComm은 사용자 간의 암호화 키 및 암호화 방식의 통일을 지향하고 있으며 가능한 암호화 키 및 암호화 방식들을 제한하고 있다. 암호화 키의 경우 X25519, P-384, P-256의 타원 곡선을 지원하며 옵션으로 P-521까지 지원한다. 암호화 알고리즘은 A256CBC-HS512를 지원하며, 키 래핑의 경우 위에서 만든 키를 기준으로 ECDH-ES, ECDH-1PU를 지원한다. 서명 키의 경우 P-256, secp256k1, Ed25519를 지원한다[5].

DIDComm은 총 3가지 종류의 메시지를 지원하며 각각 일반 텍스트 메시지, 서명 메시지, 암호화 메시지이다. 보낼 내용에 맞추어 일반 텍스트 메시지를 작성한 뒤, 서명과 암호화 여부에 따라 서명 메시지나 암호화 메시지로 나뉜다[6]. 각각의 메시지들은 JSON 기반의 메시지들로 JWT의 규칙들을 따른다. 암호화 메시지는 JWE, 서명 메시지는 JWS 기반으로 만들어진다. <그림 2>는 일반 메시지의 예시이다.


```
{
"id":"1234",
"typ":"application₩/didcomm-plain+json",
"type":"http:₩/₩/example.com₩/protocols₩/lets_do_lunch₩/1.0₩/proposal",
"from":"did:example:alice",
"to":[
"did:example:alice"
  ],
"created_time":1546521,
"expires_time":1543215,
"body":
        {
        "message":"Hello World"
        }
}
```


그림 2. 일반 메시지

DIDComm을 사용한 통신 과정은 다음과 같이 이루어진다. 먼저 통신하고 싶은 상대방의 DID를 확인한다. 확인한 DID를 사용해 DID 문서를 조회하여 메시지 암호화를 위한 공개 키와 메시지 전달을 위한 앤드 포인트를 가져온다. 사용자는 전달하고자 하는 내용 및 문서를 DIDComm 규격에 맞추어 작성을 한 뒤, 가져온 DID 문서의 공개키를 사용해 암호화 메시지를 만들어 앤드 포인트로 전달한다. 암호화 메시지를 받은 상대방은 공개 키와 쌍을 이루는 개인 키를 사용하여 복호화한 뒤, 메시지의 내용을 확인한다. 답장의 경우 메시지에 작성된 DID를 사용해 위와 같은 과정을 거치는 것으로 메시지를 전달할 수 있다[5].

3.2 DIDComm 패키지

DIDComm은 기능 구현을 위해 다양한 언어에 맞는 패키지들을 제공하고 있다. 본 논문은 해당 패키지들 중 JAVA 기반으로 만들어진 'didcomm-jvm'을 기준으로 분석하여 작성하였다.

'didcomm-jvm'은 암호화를 위한 키 생성 및 DID 문서 조회 등의 기능은 지원하지 않으며, 오직 키를 사용한 암호화 기능 및 메시지의 포맷 생성 기능을 제공한다. 메시지 암호화 및 서명을 위한 키 정보, 메시지 수신자 정보 등이 필요하며 'didcomm-jvm'에서는 위와 같은 정보들을 사용하기 위한 Interface를 요구하고 있다. DID 문서를 관리하는 DIDDocResolver와 개인 키를 관리하는 SecretResolver가 각각 구현되어 있어야 'didcomm-jvm'의 기능을 사용할 수 있다. DIDDocResolver는 상대방의 공개키 및 통신을 위한 앤드 포인트 등 수신자의 정보가 필요할 때 저장된 DID 문서를 탐색하는 기능을 가지고 있다. 이를 위해 사용자는 DID 문서를 가져오는 기능이 있어야 하며 DIDDocResolver 규격에 맞추어 DID 문서를 저장해야 한다. SecretResolver는 서명 및 메시지 복호화를 진행할 때 사용되며, 이때 개인 키를 참조할 수 있어야 한다. 사용자는 개인키를 'didcomm-jvm'에서 제공하는 Secret 규격에 맞춰 저장해야 하며, 저장된 Secret을 SercetResolver가 읽어올 수 있도록 구성하여야 한다. 여기서 각각의 키들은 JWK 형식으로 구성되어 있어야 하며, DIDDocResolver

및 SecetResolver의 Interface 사용에 문제가 없어야 한다.

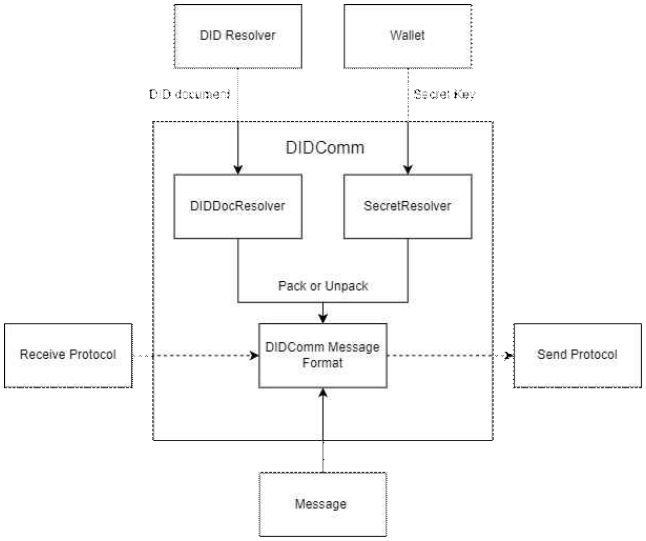

그림 3. DIDComm 기능 블록

4. DIDComm 암호화 및 서명 알고리즘 검증

'didcomm-jvm' 패키지는 JAVA 언어에서의 사용을 중점으로 만들어졌으며, 해당 패키지를 사용해 JAVA 언어로 DIDComm 기능을 사용할 수 있다. 본 논문은 'didcomm-jvm' 패키지가 DIDComm 에서 지원하는 키들을 정말로 지원하는지 직접 테스트 함으로써 'didcomm-jvm' 패키지가 제대로 구현이 되었는지 검증했다.

표 2. 테스트에 사용한 패키지 및 설명

구현 기능	패키지 이름	설명
DIDComm	didcomm-jvm	메시지 생성, 암호화 및 서명 기능
JWK	NimbusJOSE+JWT	키 생성 및 JWK 화

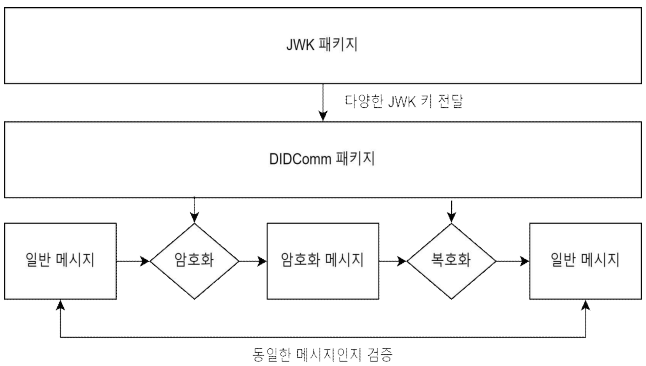

그림 4. 암호화/복호화 알고리즘 검증 환경


```
{
"ciphertext":"c8sKw_B0DIbZvizYW4oL-5-TBXuzU-8gT0g33FwpIH6jDhJkOJrUTozGuC0SarQyvZXslCE
WA38qgD_U6aJNYgvhP1-xNI3usJY_o05jXNsRnw4-6HLSGUUjO3_Pg2gwEuUKXVNaRZ7nYwkYpgF9
a9uozlkttthN8oetf1sPBh1guHR5elMTJ8ZRNe_ZbbNKR3rK4YCo-eZd9I4E9OJd_B2BDR_tNwJTOuljX-
Of8jwjPlnAWO0-20-SUV0TS9r8M1W8YQduj-jFWD6O2c5LAiigvl_mGqmqDDCmmlVmTuyYZDGL6X
HC_z_OBD5tPJZb7ZBeumNwbRtRXelOhSuB1Ng",
"protected":"eyJlcGsiOnsia3R5IjoiT0tQIiwiY3J2IjoiWDI1NTE5IiwieCI6IjZ0aTd0aGc0UVVRN1hvMUZO
OUs0VXJ0TzluNkk3MTNHblg0ajdZZUUtVncifSwiYXB2IjoiWEE3UkQ3dks0d0M4MnV5WkhXQnVrR1F
QTHNHRUlIX0dTQ1UzQi1jRUY5QSIsInNraWQiOiJkaWQ6ZXhhbXBsZTphbGljZSNrZXktMiIsImFwdSI6
IlpHbGtPbVY0WVcxd2JHVTZZV3hwWTJVamEyVjVMVEkiLCJ0eXAiOiJhcHBsaWNhdGlvblwvZGlkY29
tbS1lbmNyeXB0ZWQranNvbiIsImVuYyI6IkEyNTZDQkMtSFM1MTliLCJhbGciOiJFQ0RILTFQVStBMjU2
S1cifQ",
"recipients":[
        {
        "encrypted_key":"3OL_B_qlTu_tMBFurMX0ko8XssOqjqJbsl_0mQ-YPOs3dlheDvCH1SKS91
        X-8HbZaxH43MJPJxxn4qTDzW4kOQufkfZ4GdEw",
        "header":
                {
                "kid":"did:example:alice#key-2"
                }
        }
    ],
"tag":"bd5paoDPXiEKrb4Si4I7d4IZ5ZFJ_M4A7IfZVU8ulGY",
"iv":"AGlDBqQQ8AypBhbFE04Wsg"
}
```


그림 5. 암호화 메시지


```
{
"id":"1234",
"typ":"application₩/didcomm-plain+json",
"type":"http:₩/₩/example.com₩/protocols₩/lets_do_lunch₩/1.0₩/proposal",
"from":"did:example:alice",
"to":[
"did:example:alice"
],
"created_time":1546521,
"expires_time":1543215,
"body":
        {
        "message":"Hello World"
        }
}
```


그림 6. 암호화 메시지 복호화 결과

표 3. 암호화 테스트에 사용한 키 별 암호화/복호화 결과

kty	crv	y값 여부	ciphertext 크기	메시지 복호화 비교
EC	P-256	O	363	동일
EC	P-384	O	363	동일
EC	P-521	O	363	동일
OKP	X25519	X	360	동일


```
{
"payload":"eyJpZCI6IjEyMzQiLCJ0eXAiOiJhcHBsaWNhdGlvblwvZGlkY29tbS1wbGFpbitqc29uliwidHlw
ZSI6Imh0dHA6XC9cL2V4YW1wbGUuY29tXC9wcm90b2NvbHNcL2xldHNfZG9fbHVuY2hcLzEuMFwvc
HJvcG9zYWwiLCJmcm9tIjoiZGlkOmV4YW1wbGU6YWxpY2UiLCJ0byI6WyJkaWQ6ZXhhbXBsZTphbGlj
ZSJdLCJjcmVhdGVkX3RpbWUiOjE1NDY1MjEsImV4cGlyZXNfdGltZSI6MTU0MzIxNSwiYm9keSI6eyJtZ
XNzYWdlIjoiSGVsbG8gV29ybGQifX0",
"signatures":[
        {
        "protected":"eyJ0eXAiOiJhcHBsaWNhdGlvblwvZGlkY29tbS1zaWduZWQranNvbiIsImFsZyI6
        IkVkRFNBIn0",
        "signature":"lEz26twgqfLp-LYWOHC0JfL6OkZQg_aCbcKs95pCU_iKFpL4UAD8oluR26siQQr
        i8KuHjM0SG-NxH6V4RFuqBw",
        "header":
                {
                "kid":"did:example:alice#key-1"
                }
        }
    ]
}
```


그림 7. 서명 메시지

```
{
"id":"1234",
"typ":"application₩/didcomm-plain+json",
"type":"http:₩/₩/example.com₩/protocols₩/lets_do_lunch₩/1.0₩/proposal",
"from":"did:example:alice",
"to":[
"did:example:alice"
],
"created_time":1546521,
"expires_time":1543215,
"body":
        {
        "message":"Hello World"
        }
}
```


그림 8. 서명 메시지 복호화 결과

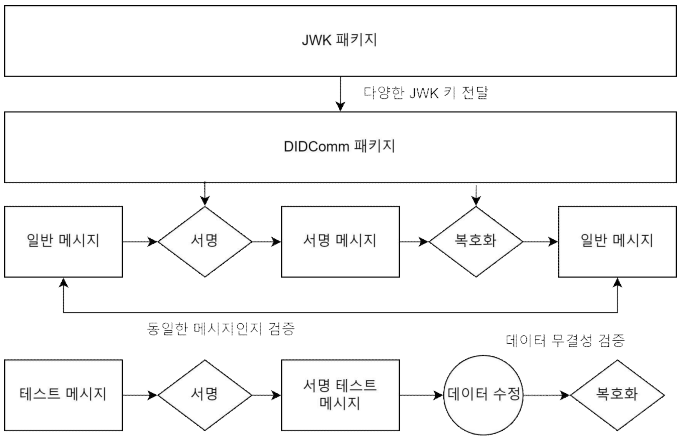

그림 9. 서명 알고리즘 검증 환경

표 4. 서명 알고리즘에 사용한 키 별 검증 결과

kty	crv	y값 여부	서명 결과 크기	메시지 복호화 비교
EC	P-256	O	86	동일
EC	secp256k1	O	86	동일
OKP	Ed25519	X	86	동일

그림 10. 데이터 무결성 검증 결과

5. 결론

DIDComm은 아직 개발 및 연구 중인 기술이며 DID를 활용한 기술이기 때문에 DID가 상용화되지 않으면 사용에 어려움이 있다. 하지만 DID가 상용화되면 그만큼 큰 장점을 지닌 기술이다. 대부분의 기술들이 중앙에서 벗어난 탈중앙화를 위해 연구되고 있으며, 이러한 흐름은 DID가 인터넷에 사용될 환경을 만들어 줌과 동시에 DIDComm을 사용할 수 있는 환경 또한 만들어 준다. 최근 우리나라는 탈중앙화와 관련된 여러 가지 기술들이 유입되고 있으며, 특히 DID의 경우 이미 다양한 회사에서 이를 활용한 제품들을 개발하고 있다. 이러한 상황에서 DIDComm은 충분한 활용성을 지닌 기술이며 충분히 상용화 될 수 있는 기술이다. 본 논문은 이러한 DIDComm에 대한 분석을 통해 이를 알리는 것으로 DIDComm을 더 쉽게 이해할 수 있도록 돕기 위해 작성되었다.

ACKNOWLEDGMENT

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

참고문헌

[1] 윤대근 “자기주권 신원증명 구조 분석서” 제이펍 2020
[2] Jones, M., Bradley, J., and N. Sakimura, "JSON Web Signature (JWS)", RFC 7515, DOI 10.17487/RFC7515, May 2015, <http://www.rfc-editor.org/info/rfc7515>.
[3] Jones, M. and J. Hildebrand, "JSON Web Encryption (JWE)", RFC 7516, DOI 10.17487/RFC7516, May 2015, <http://www.rfc-editor.org/info/rfc7516>.
[4] M. Jones, “JSON Web Key (JWK)”, RFC 7517, DOI 10.17487/RFC 7517, May 2015, <https://www.rfc-editor.org/rfc/rfc7517>.
[5] DIF Ratified Specification, “DIDComm Messaging,” Computer Security and the Data Encryption Standard, <https://identity.foundation/didcomm-messaging/spec/>.
[6] 최규현,and 김근형. "자기주권 신원 생태계를 위한 신뢰할 수 있는 통신 방법." 정보처리학회논문지. 컴퓨터 및 통신시스템 11.3 (2022): 91-98.