Journal of the Korea Convergence Society
Vol. 11. No. 5, pp. 19-25, 2020

ISSN 2233-4890 / eISSN 2713-6353
https://doi.org/10.15207/JKCS.2020.11.5.019

OpenSource를 이용한 FIDO 인증 시스템에 관한 연구

이현조[1], 조한진[2], 김용기[3], 채철주[4*]
[1]전북대학교 컴퓨터공학과 연구원, [2]극동대학교 에너지IT공학과 교수,
[3]전주비전대학교 IT융합시스템과 교수, [4]한국농수산대학 교양공통과 교수

A study on the FIDO authentication system using OpenSource

Hyun-Jo Lee[1], Han-Jin Cho[2], Yong-Ki Kim[3], Cheol-Joo Chae[4*]
[1]Researcher, Dept. of Computer Engineering, Jeonbuk National University
[2]Professor, Dept. of Energy IT, Far East University
[3]Professor, Dept. of IT Convergence System Engineering, VISION College of JeonJu
[4]Professor, Dept. of General Education, Korea National College of Agriculture and Fisheries

요 약 모바일 기기 사용자가 증가함에 따라서 민감한 개인정보를 보호하기 위해 다양한 사용자 인증 방식에 대한 연구가 활발하게 진행되고 있다. 지식기반 기법들은 인증 수단 노출이 쉬워 보안성이 저하되는 단점이 존재하며, 소유기반 기법들은 서비스를 사용하기 위한 구축비용 증가 및 사용자 편리성이 낮은 문제점이 존재한다. 이러한 문제를 해결하기 위해 본인의 스마트 기기를 활용하는 사용자 인증 기법인 FIDO 인증 시스템이 제안되었다. FIDO 인증 시스템은 사용자의 생체 정보기반 인증을 수행하기 때문에 인증 수단이 유출되는 위험이 낮으며, 아울러 사용자의 스마트 기기에 인증정보를 저장하기 때문에, 서버 해킹에 의한 사용자 정보가 노출되는 문제점을 해결한다. 이를 통해 서비스의 보안 수준에 맞는 사용자 인증기술을 선정하고 활용할 수 있다. 논문에서는 FIDO 인증 시스템에 대해 소개하고, FIDO UAF 클라이언트-서버 개발에 필요한 주요 부분을 설명하고 실제 ebay에서 제공하는 UAF 오픈소스를 활용한 구현 예제를 보여준다.

주제어 : FIDO, OpenSource, ebay FIDO, 인증, 보안

Abstract As the number of mobile device users increases, research on various user authentication methods has been actively conducted to protect sensitive personal information. Knowledge-based techniques have the disadvantage that security is deteriorated due to easy exposure of authentication means, and proprietary-based techniques have a problem of increasing construction cost and low user convenience to use the service. In order to solve this problem, a FIDO authentication system, which is a user authentication method using a smart device, has been proposed. Since the FIDO authentication system performs authentication based on the biometric information of the user, the risk of the authentication means being leaked is low, and since the authentication information is stored in the user's smart device, the user information due to server hacking is solved. Through this, it is possible to select and utilize user authentication technology suitable for the security level of the service. In this paper, we introduce the FIDO authentication system, explain the main parts required for FIDO UAF client-server development, and show examples of implementation using UAF open source provided by ebay.

Key Words : FIDO, OpenSource, ebay FIDO, Authentication, Security

*This research was supported by Basic Science Research Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Education(NRF-2017R1D1A1B03032876)
*Corresponding Author : Cheol-Joo Chae(chae.cheoljoo@gmail.com)
Received March 31, 2020 Revised May 4, 2020
Accepted May 20, 2020 Published May 28, 2020

1. 서론

최근 사용자들은 인터넷뿐만 아니라, Online-to-Offline 서비스를 통해 많은 활동을 온라인으로 수행할 수 있게 되었다. 이에 따라 사용자가 서비스를 사용하기 위해 입력한 검색 키워드, 소셜 네트워크 활동 내역, 온라인 결제 정보 등이 서비스 제공자의 서버에 저장되어 활용된다. 이러한 사용자 입력 데이터는 사용자의 프라이버시를 포함하는 민감한 정보이기 때문에, 이에 접근하는 사용자가 시스템에 등록된 적합한 사용권한을 지닌 사용자인지 판단하는 사용자 인증 기법이 필수적이다. 이를 위해 비밀번호, PIN 번호와 같은 사용자의 지식에 기반한 인증 기법, 일회용 비밀번호 OTP(one time password)나 공인인증서 등을 이용하는 사용자가 소유한 인증 매체 기반의 인증 기법 등 다양한 사용자 인증 기법이 활용되었다[1, 2]. 그러나 비밀번호나 PIN 번호는 사용자의 기억에 의존하기 때문에 대다수의 사용자들이 기억하기 쉬운 간단한 문자열을 사용할 뿐만 아니라, 서로 다른 온라인 서비스에도 동일하거나 유사한 비밀번호를 사용한다. 따라서 추측이나 엿보기 등 간단한 공격만으로 노출되기 쉽다. 한편, 사용자가 소유한 인증매체를 활용하면 인증매체 없이 접근이 제한되기 때문에 보안성이 향상된다. 그러나 인증매체의 구조가 복잡하여 인증매체 개발 및 이를 활용한 인증 시스템 구축비용이 증가하고, 신분확인을 위한 서비스 제공자 방문이 요구된다. 아울러 온라인 서비스 이용 시 인증매체를 항상 소유하고 있어야 하므로 편리성이 낮다는 단점이 있다. 또한 기존 사용자 인증 기법은 서버에 사용자에 관련한 정보를 저장하고 입력된 정보와 비교하여 사용자의 적합성을 판단하기 때문에 항상 서버 해킹에 의해 민감한 개인정보가 유출될 위험이 존재한다[4-7].

이러한 문제를 해결하기 위해 비밀번호 없이 본인의 스마트 기기를 통한 인증 후 온라인 서비스를 이용할 수 있는 FIDO(Fast Identity Online) UAF(Universal Authentication Framework)이 제안되었다[8]. 개인용 스마트 기기는 항상 휴대하기 대문에 추가적인 인증매체가 필요하지 않고, 타인이 이용할 가능성이 낮아 잘못된 사용자가 인증을 사용하는 문제점이 감소된다. 또한 최신 스마트 기기의 경우, 지문인식, 홍채인식 등 첨단 생체 인식 기술을 탑재하였기 때문에, 별다른 사용자 입력 없이 보안성이 높은 인증을 수행할 수 있다. 사용자의 생체 정보를 스마트 기기에 저장하고, 인증이 필요한 경우 스마트 기기 내의 인증 앱을 통해 사용자를 판단하여 암호화된 값을 생성한다. 암호화 값은 은행 등의 온라인 서비스에 전송하여 확인함으로써 인증을 수행한다. 따라서 사용자의 개인 정보가 서버에 저장 및 관리되지 않아 정보 노출의 위험이 낮고, 지문, 홍채인식 등의 생체 정보를 활용하기 때문에 인증의 보안성이 높다.

현재 다양한 기업들이 FIDO UAF를 구현하고 상품 서비스를 제공하고 있으나, 연구를 위한 UAF 개발 시 참조할 수 있는 오픈 소스나 가이드라인이 많지 않아 UAF 클라이언트-서버 개발이 어려운 실정이다. 이러한 문제를 해결하고자 본 논문에서는 FIDO UAF에 대한 간략히 소개하고, FIDO UAF 클라이언트-서버 개발에 필요한 주요 부분을 설명하고 실제 ebay에서 제공하는 UAF 오픈소스를 활용한 구현 예제를 보여준다.

2. FIDO 인증 기술

FIDO 인증 기술은 UAF(Universal Authentication Framework) 프로토콜로 사용자의 스마트폰에 탑재된 인증수단(예: 지문인식 센서)을 온라인 서비스와 연동하여 사용자를 인증하는 기술과 U2F(Universal 2nd Factor) 프로토콜로 기존 패스워드를 사용하는 온라인 서비스에서 2차 인증요소로 토큰기반 인증을 사용자 로그인 시에 추가할 수 있는 기술로 구성된다. U2F의 경우, 현재 크롬, 윈도우 등에서 지원되고 있는 장점이 있다. UAF의 경우 클라이언트, 서버, 두 개체 간에 주고받는 프로토콜로 구성되어 있다. 이때, FIDO의 인증을 위한 토큰 생성 및 관리 API를 만족하면 다양한 종류의 인증 방법을 사용할 수 있다[8, 9].

FIDO UAF 아키텍처는 크게 사용자 기기와 사용자에게 서비스를 제공하는 웹 서버로 분류된다. 사용자 기기, 즉, 스마트 폰의 프레임워크는 서비스를 제공하는 RP 어플리케이션(Relying-Party Application), FIDO 클라이언트, FIDO ASM (Authenticator Specific Module), FIDO 인증장치(Authenticator)로 구성된다. FIDO 클라이언트는 FIDO 프로토콜을 수행하며 서비스 응용의 인증정책에 부합하는 인증장치가 사용되도록 지원한다. FIDO ASM은 FIDO 클라이언트가 FIDO 인증장치에 접근할 수 있도록 하며, FIDO 인증장치는 FIDO 프로토콜에서의 인증장치 등록/인증/탈퇴 및 실질적인 사용자 인증을 수행한다. 웹 서버는 서비스 응용의 인증정책을 관리하며 사용자 등록/인증/탈퇴에 필요한 프로토콜을 수행한다.

FIDO 기술은 사용자의 스마트폰에 저장된 인증수단을 이용해 인증하고, FIDO 인증장치가 표준화된 인증 프로토콜을 이용해 서버와 원격 인증을 수행한다. 따라서 서비스 기업은 해당 보안 요구수준에 적합한 인증 수단들을 선정하여 사용자에게 제공할 수 있으며, 사용자는 익숙한 인증 수단을 통해 서비스를 제공받을 수 있다. 특히 FIDO 기술은 사용자의 민감한 생체정보가 사용자의 스마트폰에만 저장되기 때문에, 인증 서버를 통해 생체정보가 유출될 수 있는 문제를 방지한다. 따라서 서비스 제공자 측에서 대규모 DB 유출 사고가 발생하더라도 다른 사이트로 피해가 확산되는 것을 막을 수 있다. Fig. 1은 FIDO UAF 구조를 보여주고 있다.

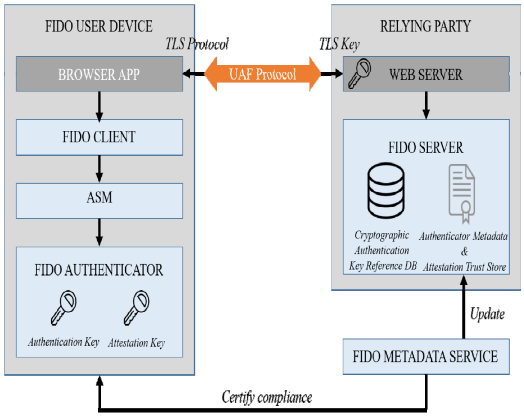

Fig. 1. FIDO UAF High-Level Architecture

3. FIDO UAF 인증 시스템

3.1 FIDO UAF Client

FIDO UAF 클라이언트는 표준화된 인증장치 드라이버 ASM(Authenticator Specific Module)을 통하여 인증 서비스를 제공하는 뱅킹, 공인인증, 신용카드 결제 등의 다양한 RP 어플리케이션(Relying Party Application)에서 요구하는 인증 장치를 검색하고 연동을 수행한다. FIDO 클라이언트는 ASM의 API를 활용하여 인증장치 특성 정보조회, 인증장치 등록, 인증, 인증장치 해지, 인증장치 등록조회, 인증장치 환경 정보 설정 등을 수행할 수 있다. FIDO UAF는 장치에 따라 크게 Web을 위한 DOM, 안드로이드를 위한 인텐트(Android Intent), iOS를 위한 커스텀 URL(Custom URL) 방식의 API를 지원한다. 본 논문에서는 안드로이드 상에서의 클라이언트 구현을 위해 인텐트 기반의 API를 사용하였다[10].

안드로이드 기반의 FIDO UAF 클라이언트를 구현할 때, 주의해야할 사항은 안드로이드 버전이다. 기본적으로 FIDO UAF는 안드로이드 버전 5.0 이상에서 구현하는 것을 권하고 있다. 만약 안드로이드 5.0 이전 버전을 사용하는 경우 'org.fidoalliance.uaf.permissions.FIDO_CLIENT'을 이용한 권한 획득(permission) 및 사용 승인(uses-permission)을 반드시 선언해야 한다. 이를 이용한 사용 승인 예제는 Fig. 2와 같다. 다만 최근 개발된 스마트 기기 또는 안드로이드 에뮬레이터의 경우 버전이 5.0 이상이기 때문에 위의 권한 획득 및 사용 승인을 선언할 필요가 없다.

```
<permission
    android:name="org.fidoalliance.uaf.permissions.FIDO_CLIENT"
    android:label="Act as a FIDO Client."
    android:description="This application acts as a FIDO Client."
    android:protectionLevel="dangerous"
/>
<uses-permission android:name="org.fidoalliance.uaf.permissions.FIDO_CLIENT"/>
```

Fig. 2. Example of uses-permission

UAF 클라이언트는 'message'라는 이름으로 설정한 'extra' 메시지를 송수신함으로써 통신을 수행한다. 사용자는 한 번에 하나의 안드로이드 앱을 선택하여 실행할 수 있다. 이때 클라이언트의 인텐트 식별자는 반드시 'org.fidoalliance.intent.FIDO_OPERATION'로 설정되어야 하며, 인텐트 타입은 'fido.uaf_ client+json'으로 설정해야 한다. 메시지를 설정하고 startActivityForResult() 함수를 호출하여 전송한다. 메시지의 응답은 onActivityResult() 함수를 통해 수신된다. 인텐트의 'extra' 메시지 내에 어떠한 요소(elements)를 삽입하였는가에 따라 다양한 기능을 수행할 수 있다. 인텐트의 'extra' 메시지는 Fig. 3과 같은 요소를 지닌다.

```
dictionary Extra {
    String UAFIntentType;
    String discoveryData;
    String componentName;
    Short errorCode;
    String message;
    String origin;
    String channelBindings;
    Short responseCode;
};
```

Fig. 3. Elements of extra

- UAFIntentType : 안드로이드 인텐트의 타입을 결정하는 변수로써, 사용가능한 인증장치의 종류를 확인하고, 해당 인증장치가 FIDO UAF 보안정책을 준수하는지 확인하기 위해 사용된다. 아울러 클라이언트에서 수신한 요청메시지를 처리하기 위해 사용되기도 한다.
- discoveryData : 클라이언트에서 접속가능한 인증장치에 대한 정보를 표현하며, UAF 프로토콜 버전, 클라이언트 제작사, 클라이언트 버전 및 사용가능한 인증장치 등의 정보를 포함한다.
- componentName : 클라이언트 앱의 구성 요소명을 나타내며, ComponentName.flattenString() 함수에 의해 반드시 일련화가 이루어져야 한다.
- errorCode : 요청 실행시 발생한 에러에 대한 확인 코드를 나타낸다. 기본적으로 NO_ERROR부터 UNKNOWN까지 9가지 에러코드가 정의되어 있다.
- message : UAF 프로토콜을 통해 요청하거나 응답하는 메시지를 나타낸다.
- origin : 'org.fidoalliance.permissions.ACT_AS_WEB_BROWSER'을 사용하여 WEB_BROWSER 사용 권한을 획득하였을 때 이를 나타낸다.
- channelBindings : 요청에 대한 channel binding을 나타낸다.
- responseCode : 현재 메시지에 대한 처리 진행 상황을 나타낸다.

클라이언트는 사용자 요청을 받을 때마다 메시지를 분석하고, 각 메시지의 인텐트 타입에 따른 처리를 수행한다. 처리 결과에 따라 서버에 응답 또는 오류 메시지를 전송한다. 이 과정은 크게 Fig. 4의 인증장치 확인 및 연결을 수행하는 Discover와 Fig. 5의 연결된 인증 장치가 현재 보안 정책에 적합한지 확인하는 Check_policy 단계로 구성된다.

앞에서 언급한 Extra의 discoveryData에 저장되는 FIDO UAF 클라이언트 관련 정보는 Discover, Check_policy 수행을 통해 확인 및 저장된다. 사용자가 직접 안드로이드 버전, 사용가능한 인증장치 등을 확인하고 소스코드 내에 하드코딩을 할 필요성이 없다.

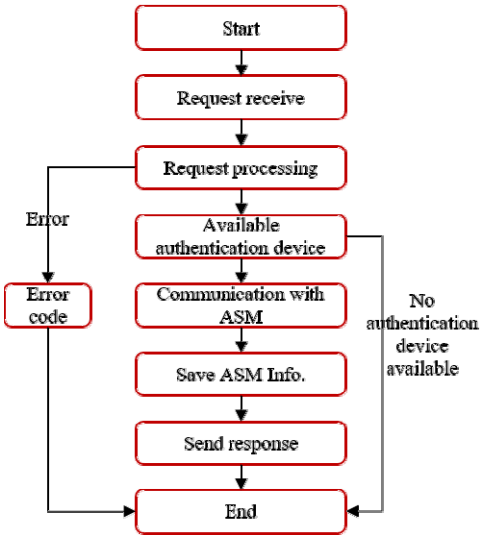

Fig. 4. Process of Discover

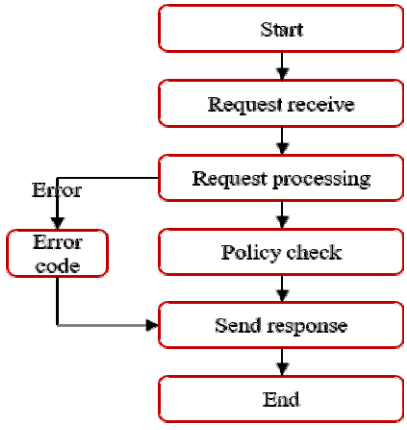

Fig. 5. Process of Check_policy

3.1.1 ebay UAF client implementation

이베이의 오픈소스[11]에 포함된 FIDO UAF Client는 안드로이드에서 구동가능한 앱으로써, 안드로이드 5.0 이상에서 실행가능하다. 크게 fido.uaf 및 fidouafclient 패키지로 구성되어 있다. fido.uaf 패키지는 안드로이드 모바일에서 동작하기 위한 코어 패키지이며, fidouafclient 패키지는 HTTP 관련 요청 및 응답 프로토콜, 서버 주소 설정, 안드로이드 시스템 리소스 할당 등 외적인 부분을 구현한 패키지이다. 이 중 fido.uaf 패키지 내의 msg 및 tlv에는 FIDO 클라이언트 외에도 RP 응용, 인증장치, ASM 등의 역할이 모두 통합되어 구현되어 있다. 따라서 해당 클라이언트 소스코드를 실행하면 스마트 기기에

FIDO UAF 클라이언트와 인증장치가 포함되어 있지 않더라도 테스트를 수행할 수 있다. 실행을 위한 메인 파일은 org.ebayopensource.fidouafclient.util에 있는 MainActivity.java이다. Fig. 6, Fig. 7은 FIDO UAF 클라이언트를 실행한 화면을 나타낸다.

Fig. 6. MainActivity

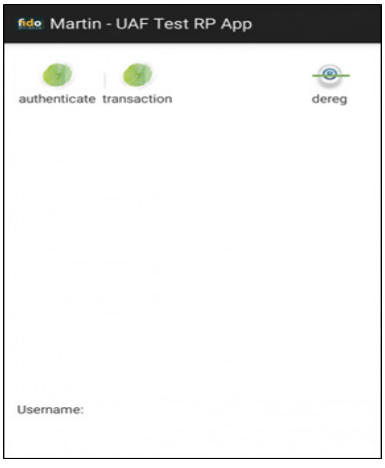

Fig. 7. RegisteredActivity

3.2 UAF server

FIDO UAF 서버는 RP 서버(Relying Party Server)의 사용자 인증 요청을 받아 인증을 처리하고 결과를 반환하는 역할을 수행한다. 서버의 주요 역할은 크게 네 가지로 구성된다[12-14]. 첫째, FIDO 얼라이언스에서 제공하는 메타데이터 서비스(metadata service)나 인증장치 제작사의 웹사이트에서 접속하여 메타데이터를 다운로드하고 이를 로컬 스토리지에 저장하여 정기적으로 업데이트를 수행한다. FIDO 얼라이언스로부터 검증된 인증장치 제작사는 메타데이터 서비스에 자체 제작한 인증장치 메타데이터를 업데이트 할 수 있다. 메타데이터 서비스의 정보는 디지털 서명을 통해 정보의 변조를 방지하고 있으며, RP서버는 인터넷을 통해 가장 최신의 메타데이터를 다운로드 받아 사용할 수 있다. 둘째, 보안 정책을 정의하고 UAF 클라이언트와 통신하기 위한 프로토콜 메시지를 처리한다. 셋째, 서명을 확인하고 카운터를 점검하여 인증된 사용자가 적합한지를 검증한다. 넷째, 사용자 계정에 대해 등록된 UAF 인증자의 정보를 관리한다.

FIDO UAF 서버를 위해, 메타데이터를 관리할 수 있는 기능을 포함하는 메타데이터 관리 라이브러리(UAF metadata library)와 UAF 요청/응답 메시지를 처리하는 메시지 라이브러리(UAF server library)를 구현한다. 각 라이브러리에서 제공하는 함수 API는 다음과 같다.

- UAF metadata library

· update_metadata() : FIDO 얼라이언스에 의해 검증된 인증장치 메타데이터를 최신 정보로 업데이트 한다. 다운로드 받는 URL은 'https://mds.fidoalliance.org'이며, 다운로드된 데이터는 JSON 웹 서명[15, 16] 기능을 통해 분석하고 검증한다. 만약 적합한 메타데이터로 검증이 완료되면 로컬 데이터베이스를 업데이트하고 타임스탬프를 웹 서버로 반환한다.

· get_metadata($aaid) : 인증장치의 고유 아이디를 사용하여 데이터베이스에서 데이터를 탐색한다. 아이디 탐색에 성공하면 인증장치에 대한 모든 정보를 포함하는 배열을 반환한다.

- UAF server library

· construct_registration_request($registered_token, $appID, $username) : 사용자 이름 및 모바일 앱을 UAF에 등록한다. 이를 위해 웹 서버에서 미리 정의된 보안 정책 개체를 읽어들인 다음 기존에 등록된 토큰을 보안 정책 개체의 미허용 배열 부분에 저장한다. 이후 appID, 사용자 이름을 메시지에 할당하여 요청한다.

· process_registration_response($response, $request) : 검증된 인증서를 사용하여 디지털 서명을 검증하고, 서명이 유효하다면 요청된 사용자 이름 및 모바일 앱을 등록한다. 이후 인증장치의 aaid, keyID, 공개

키, 카운터 및 검증된 인증서가 포함된 배열 객체를 반환하여 응답한다.

· construct_authentication_request($appID) : 등록된 앱에 대한 사용자 인증을 요청한다.

· process_authentication_response($registered_token, $response, $request) : 사용자 인증 결과를 처리한다. 등록된 토큰을 response 내의 키핸들에서 검색한다. 이후 공개키를 사용하여 디지털 서명을 검증하고 카운터 값을 확인한다. 만약 유효하다면 인증장치의 aaid, keyID, 카운터가 저장된 배열 객체를 반환한다.

· construct_deregistration_request($registered_token, $appID) : 모바일 앱에 등록된 사용자 및 인증장치에 대한 해제를 요청한다.

3.2.1 ebay UAF server implementation

이베이의 UAF 오픈소스[11] 중 서버 파트는 핵심 기능을 구현한 Core 파트 및 실제로 구동 및 이용이 가능한 Demo 파트로 구성된다. 먼저 core 파트(core.uaf)는 6개 패키지로 나누어진다. 각 패키지는 다음과 같다.

- Crypto : 인증서 검증 및 암호화 수행 클래스
- Msg : 등록, 검증, 해지를 위한 메시지(Reg, Dereg, Auth)를 주고 받을 때 쓰기 위한 컨테이너
- Ops : 서버 측 수행 명령 구현 클래스
- ri.client : 라이브러리 테스트 서버로써 주요 파라메터들이 미리 하드코딩 되어 있음
- storage : 인증장치와 사용자 디바이스를 관리하기 위한 튜플 및 인터페이스
- tlv(Tag Length Value) : UAF 메시지 태그 처리 및 검증 클래스

한편, Demo 파트는 Core 파트를 참조하여 실제 구동 가능한 서버를 구현한 부분이다. org.ebayopensource.fidouaf.res 내의 FidoUafResource.java 파일이 메인 실행 파일이다. Demo 파트를 구성하는 패키지는 다음과 같다.

- Facets : RP응용 또는 RP서버 확인을 위한 FacetID를 저장하기 위한 튜플
- RPserver.msg : core파트의 mag와 동일하며, RP 서버 실행을 위해 TokenType와 Token이 추가적으로 정의되어 있음
- Stats : 로그 저장 클래스. 등록, 검증, 해지 명령(Auth, Reg, Dereg)이 처리된 내역을 해시맵으로 저장
- Res : Notary, Storage 인터페이스를 구현한 클래스 및 Response, Dereg Request처리 클래스

이베이의 Demo 서버 파트는 RP서버와 FIDO서버의 기능을 하나의 서버에서 수행하도록 구성되어 있다. 특히 주요 파라메터가 하드코딩 되어 있기 때문에 테스트 이외의 환경에서 사용하기 위해서는 저장소 및 관련 서버 내용을 따로 구현해야 한다. 아울러 RP 서버와 FIDO 서버를 분리해야 하기 때문에, 실제 응용 환경에서 사용하기 위해서는 각각의 역할에 따라 별도로 구현해야 한다.

4. 결론 및 향후 연구

모바일 기기가 다양해지고 모바일 기기를 이용한 인증 기술에 대한 방법들이 다양해짐에 따라서 사용자는 오히려 개인정보노출의 위험이 증가하는 문제가 발생하고 있다. 이를 위해 다양한 사용자 인증 기술이 개발되었으며, FIDO 프레임워크를 활용하여 이러한 인증기술을 활용할 수 있다.

본 논문에서는 FIDO UAF 서버 및 클라이언트 프레임워크를 구현 및 테스트할 때 중요한 파라메터 및 API 부분에 대해 설명하고, 이베이에서 오픈소스로 제공하는 FIDO UAF 구현 코드를 간략하게 소개하였다. 구현한 UAF 클라이언트는 단순한 메시지 송수신 기능만이 구현되어 있으며, 서버의 경우에도 최소한의 기능만이 구현되어 있어 실제 응용환경에서 활용하기 위해서는 추가적인 기능 구현이 필수적이다. 향후 연구로는 현재 ebay UAF DEMO server를 바탕으로, FIDO alliance에서 제시한 표준을 따르는 RP 서버 및 UAF 서버를 각각 구현하고 이를 통한 인증을 수행하도록 한다.

REFERENCES

[1] T. H. Park, G. R. Lee & H. W. Kim. (2017). Survey and Prospective on Privacy Protection Methods on Cloud Platform Environment. *Journal of the Korea Institute of Information Security and Cryptology, 27(5),* 1149-1155.

[2] T. Y. Kim, H. J. Jun & T. S. Kim. (2018). An Analysis

on Intention to Use Information Service for Personal Information Breach. *Journal of the Korea Institute of Information Security and Cryptology, 28(1),* 199-213.

[3] S. J. Kim & S. S. Yeo. (2013). A Study on Secure Data Access Control in Mobile Cloud Environment. *Journal of Digital Convergence, 11(2),* 317-322.

[4] H. T. Chae & S. J. Lee. (2014). Security Policy Proposals through PC Security Solution Log Analysis (Prevention Leakage of Personal Information). *Journal of the Korea Institute of Information Security & Cryptology, 24(5),* 961-968.

[5] S. Yun. (2017). The Biometric Authentication Scheme Capable of Multilevel Security Control. *Journal of the Korea Convergence Society, 8(2),* 9-14.

[6] S. Khandelwal. (2016). QRLJacking—Hacking Technique to Hijack QR Code Based Quick Login System, The Hacker New(Online). https://thehackernews.com/2016/07/qrljacking-hackingqr-code.html

[7] J. H. Jeon. (2016). A Study on Security Risk according to the activation of Bio-Authentication Technology. *Convergence security journal, 16(5),* 57-63.

[8] https://fidoalliance.org/

[9] J. Kim. (2015). Study on the password-free certification system using the FIDO (Fast IDentity Online). Communications of the Korea Information Science Society, *KIISE, 33(5).*

[10] FIDO Alliance. (2016). FIDO UAF Application API and Transport Binding Specification v1.0, https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-client-api-transport-v1.0-ps-20141208.html

[11] npesic et al. (2016). UAF — Universal Authentication Framework. https://github.com/eBay/UAF

[12] FIDO Alliance. (2016). FIDO UAF Architectural Overview. https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-overview-v1.0-ps-20141208.html

[13] FIDO Alliance. (2016). FIDO UAF Protocol Specification v1.0. https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-protocol-v1.0-ps-20141208.html

[14] FIDO appID and facet specification. https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-appid-and-facets-v1.0-ps-20141208.html

[15] JSON, http://www.json.org/

[16] RFC 7515 - JSON Web Signature, https://tools.ietf.org/html/rfc7515

이 현 조(Hyun-Jo Lee) [정회원]

· 2008년 2월 : 전북대학교 컴퓨터공학과(공학석사)
· 2014년 2월 : 전북대학교 컴퓨터공학과(공학박사)
· 2015년 9월 ~ 2016년 8월 : 한국과학기술정보연구원 선임연구원
· 2018년 6월 ~ 현재 : 전북대학교 시간강사 및 연구원
· 관심분야 : 시공간 데이터베이스, 정보보호, 사용자 인증
· E-Mail : o2near@gmail.com

조 한 진(Han-Jin Cho) [종신회원]

· 1999년 2월 : 한남대학교 컴퓨터공학과(공학석사)
· 2002년 8월 : 한남대학교 컴퓨터공학과(공학박사)
· 2002년 3월 ~ 현재 : 극동대학교 에너지IT공학과 교수
· 관심분야 : 정보보호, 스마트폰 보안, 모바일 콘텐츠
· E-Mail : hanjincho@hotmail.com

김 용 기(Yong-Ki Kim) [정회원]

· 2005년 2월 : 전북대학교 컴퓨터공학과(공학석사)
· 2011년 2월 : 전북대학교 컴퓨터공학과(공학박사)
· 2012년 4월 ~ 2016년 3월 : 한국과학기술정보연구원 선임연구원
· 2016년 4월 ~ 현재 : 전주비전대학교 IT융합시스템과 교수
· 관심분야 : 데이터베이스, 정보보호, 네트워크 보안
· E-Mail : braves0815@gmail.com

채 철 주(Cheol-Joo Chae) [종신회원]

· 2009년 8월 : 한남대학교 컴퓨터공학과(공학박사)
· 2009년 9월 ~ 2013년 4월 : 한국전자통신연구원 선임연구원
· 2013년 4월 ~ 2016년 8월 : 한국과학기술정보연구원 선임연구원
· 2016년 9월 ~ 현재 : 한국농수산대학 교양공통과 교수
· 관심분야 : 정보보호, 바이오 보안, 네트워크 보안
· E-Mail : chae.cheoljoo@gmail.com