FIDO 1.0 패스코드 인증장치 구현

조영섭*, 김석현*, 조상래*, 김수형*
*한국전자통신연구원 인증기술연구실

e-mail : {yscho, ksh4uu, sangrae, lifewsky}@etri.re.kr

FIDO 1.0 Passcode Authenticator Implementation

Young-Seob Cho*, Seok-Hyun Kim *, Sang-Rae Cho *, Soo-Hyung Kim*
*Authentication Research Section, Electronics and Telecommunications Research Institute

요 약

FIDO(Fast IDentity Online)는 인증 프로토콜과 인증수단을 분리하여 지문, 홍채, 스마트카드 등 다양한 인증 기술을 쉽게 수용할 수 있도록 하여 인증강도를 높이면서 사용자의 편리성도 제고할 수 있는 기술로 기존 패스워드 인증 방식의 문제를 해결하며 최근 모바일 결제 등으로 그 활용의 폭이 급격히 높아지고 있다.

본 논문은 FIDO 1.0 을 준용하는 패스코드 인증장치의 설계 및 구현에 대하여 기술한다. 본 인증장치는 2015 년 4 월에 개최된 FIDO UAF 1.0 상호운용성(IOP) 테스트를 통과한 FIDO Certified 인증장치로 향후 모바일 결제, 온라인 서비스 로그인 등 다양한 분야에서 활용될 것으로 예상된다.

1. 서론

패스워드는 초창기 컴퓨팅 환경부터 최근의 인터넷 환경까지 가장 광범위하게 사용되는 사용자 인증 방식이다. 이것은 패스워드가 제공하는 보편적인 사용의 편의성 때문이다. 그러나 패스워드는 약한 보안강도를 제공하는 문제점, 도용의 문제, 여러 사이트에 중복 사용하는 문제점 등을 가지고 있다. 또한 최근 스마트폰이 확산됨에 따라 패스워드 입력의 불편함이 발생하여 사용의 편의성에서도 문제가 발생하고 있다. 이러한 패스워드 문제를 해결하기 지난 수십 년 동안 위한 다양한 인증 기술이 연구되어 왔다[1].

FIDO(Fast IDentity Online) Alliance[2]는 온라인 환경에서 바이오 인식기술을 활용한 인증방식에 대한 기술표준(De Facto)을 정하기 위해 2012 년 7 월 설립되었다. FIDO 는 서버에서 사용자를 직접 인증하던 기존 인증 방식과 달리 사용자 단말에서 사용자를 인증하는 로컬 인증과 서비스 제공 서버에서 수행하는 원격 인증으로 인증을 분리하고 있다. 로컬 인증은 지문, 홍채 등의 생체인증뿐만 아니라 스마트카드, 패스코드 등 다양한 인증 방식을 지원한다. 원격인증은 공개키 방식의 인증 프로토콜을 규정하고 있다. 이와 같은 인증의 분리는 서비스 제공자가 원격인증만을 지원하면, 사용자 단에서 어떠한 방식의 로컬 인증을 사용하더라도 서비스 제공자 부분을 수정할 필요가 없게 된다. 즉 서비스 제공자는 사용자에 대한 직접적인 인증 기술이 변경되더라도 수정할 필요가 없게 되며, 사용자의 생체 정보, 패스워드 등과 같은 민감한 정보가 서버에 집적되어 발생하는 문제를 회피할 수 있게 된다. FIDO 는 2014 년 12 월에 FIDO UAF(Universal Authentication Framework) 1.0 규격을 발표하였으며 FIDO 규격을 준용하는 제품에 대한 상호운용성(IOP) 테스트를 통해 FIDO Certification 을 발급하고 있다. 현재 국내에서도 ETRI 를 포함한 다수의 업체 제품이 FIDO 인증을 받았다. FIDO 1.0 을 이용한 대표적인 서비스로는 현재 삼성 갤럭시 S6 스마트폰에서 FIDO 인증을 받은 지문 인증장치를 이용하여 모바일 결제 서비스를 제공하는 삼성 페이가 있다.

본 논문은 FIDO 1.0 을 준용하는 패스코드 인증장치의 설계 및 구현에 대하여 기술한다. 본 논문의 구성은 다음과 같다. 2 장에서 FIDO UAF 1.0 의 개요 및 FIDO UAF 1.0 인증장치 규격에 대하여 기술한다. 3 장에서 FIDO 1.0 패스코드 인증장치의 설계에 대하여 기술하고 4 장에서 구현에 대하여 기술한다. 마지막으로 5 장에서 결론을 맺는다.

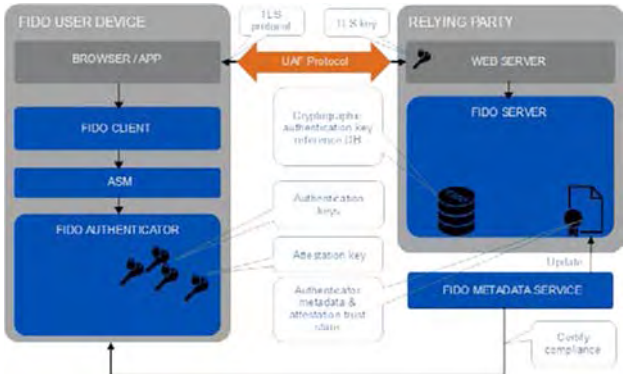
2. FIDO UAF 1.0

본 장에서는 FIDO UAF 1.0 의 기본 개요와 인증장치의 구조에 대하여 기술한다.

2.1 FIDO UAF 1.0 개요

FIDO UAF 1.0 의 구조는 아래 그림 1 과 같다[3].

본 연구는 미래창조과학부 및 정보통신기술진흥센터의 정보통신• 방송 연구개발 사업의 일환으로 하였음 [B0126-15-1007, "상황인지 기반 멀티팩터 인증 및 전자서명을 제공하는 범용 인증 플랫폼 기술 개발"].



(그림 1) FIDO UAF 구조

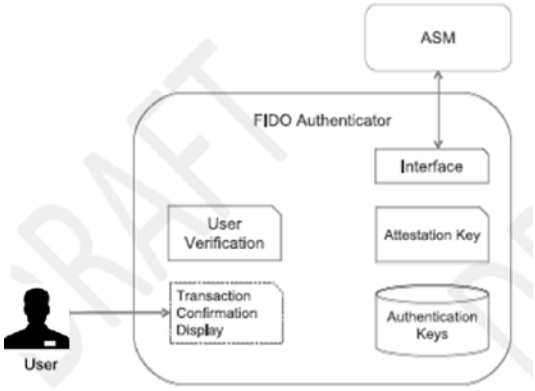
FIDO UAF 는 FIDO Server, FIDO Client, ASM(Authenticator Specific Module)[4], FIDO Authenticator, FIDO Metadata Service 로 구성되며 그림에서 표기된 방식으로 상호작용을 수행한다. FIDO Server 는 FIDO Client 에서 전달한 인증토큰을 검증하고 관리하는 기능을 제공한다. FIDO Client 는 FIDO 서버의 요청을 분석하여 적절한 인증장치에 그 요청을 전달하고 인증장치 응답을 FIDO 서버에게 전달한다. ASM 은 FIDO Client 의 요청을 인증장치에게 전달하고 그 응답을 FIDO Client 에게 전달한다. FIDO Authenticator 는 실제 사용자의 Credential 정보를 생성하고 관리하며 FIDO 서버의 인증장치 등록과 인증 요청을 처리한다. FIDO Metadata Service 는 인증장치에 대한 최신 메타데이터 조회 서비스를 제공한다.

FIDO Server 와 FIDO Client 는 표준화된 UAF Protocol 을 통해 통신한다. UAF Protocol 은 다음과 같은 4 가지 메시지를 정의하고 있다[3]. Authenticator Registration 은 사용자가 이용하는 인증장치를 FIDO 서버에 등록하는 기능에 사용되며, User Authentication 은 Relying Party 에서 사용자를 인증하는데 사용된다. Secure Transaction Confirmation 은 사용자 인증뿐만 아니라 전달된 메시지에 대한 사용자 확인 정보를 Relying Party 에 제공할 때 사용된다. Authenticator Deregistration 은 사용자의 서비스 탈퇴 등과 같은 경우 등록된 인증장치를 해지하는 기능에 사용된다.

2.2 FIDO UAF 인증장치

FIDO UAF 1.0 인증장치의 논리 구조는 그림 2 와 같다[5].

FIDO 인증장치는 ASM 으로부터 요청 메시지를 전달받아 이를 처리하여 응답을 ASM 에게 전달한다. 사용자가 인증장치를 FIDO 서버에 등록할 때, 인증장치는 추후 인증을 위해 공개키 쌍을 생성하여 개인키는 인증장치에서 관리하고 공개키는 FIDO 서버에 전달하여 FIDO 서버에서 인증장치에서 생성한 인증 토큰을 검증하는데 사용할 수 있도록 한다. 하나의 FIDO 서버는 다수의 Relying Party 들을 지원할 수 있는데 이 경우 Relying Party 별로 인증장치 등록시 공개키 쌍을 생성한다. Authentication Keys 는 이와 같이 생성된 개인키들에 대한 키 정보를 나타낸다.



(그림 2) FIDO UAF Authenticator 논리 구조

Attestation Key 는 인증장치 제조사에서 인증장치를 제조할 때 인증장치의 신뢰를 증명(Attestation)할 수 있도록 설치하는 비대칭키의 개인키 정보이다. 인증장치 등록시 생성된 공개키 정보를 Attestation Key 로 서명하여 FIDO 서버에 전달하고, FIDO 서버는 인증장치 제조사가 배포한 Attestation Key 에 대응되는 공개키 인증서를 통해 인증장치가 전달한 인증장치 등록 결과를 검증할 수 있게 된다.

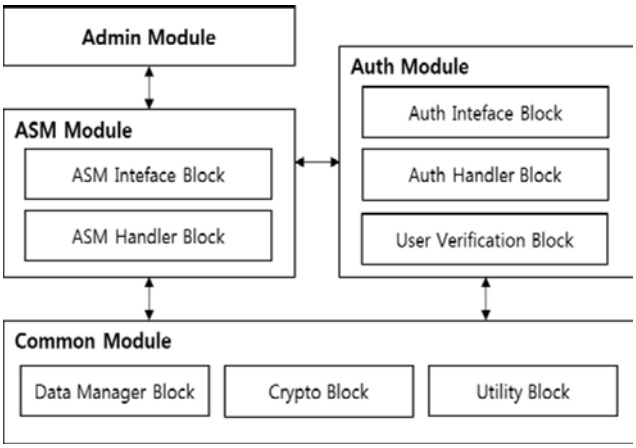
Transaction Confirmation Display 는 결제, 자금 이체, 서비스 등과 같이 사용자에게 반드시 확인을 받아야 하는 메시지가 있는 경우, 사용자에게 해당 메시지를 출력하여 확인을 받는 기능이다. 이 기능은 인증장치에서 선택적으로 제공할 수 있는 기능이다.

User Verification 은 인증장치에서 사용자를 인증하는 기능을 제공하며 FIDO 의 로컬 인증에 해당한다. User Verification 은 패스코드, 키 패턴, 음성, 화상, 지문, 홍채 등 다양한 인증 기술을 사용할 수 있다.

3. FIDO 1.0 패스코드 인증장치 설계

본 장에서는 ETRI 에서 개발한 FIDO 1.0 패스코드 인증장치의 설계에 대하여 기술한다.

3.1 FIDO 1.0 패스코드 인증장치 구조



(그림 3) FIDO 패스코드 인증장치 구조도

FIDO 1.0 패스코드 인증장치의 구조는 그림 3 과 같이 여러 모듈로 구성된다.

ASM 모듈은 ASM 에서 관리하는 인증장치 정보를 제공하는 GetInfo, 인증장치를 등록하는 Register, 서버에서 요청한 메시지를 서명하는 Authenticate 기능, 인증장치를 해지하는 Deregister 기능, 현재 ASM 에 등록한 사용자 정보를 제공하는 GetRegistration 기능 및 인증장치를 설정하는 OpenSettings 기능을 제공한다. ASM 모듈은 ASM Interface 블록과 ASM Handler 블록으로 구성된다. ASM Interface 블록은 Admin Module 이나 FIDO Client 에 의해 호출되며 ASM 모듈이 지원하는 기능을 제공한다. ASM Handler 블록은 ASM 이 제공하는 기능 각각을 실질적으로 구현하는 핵심 모듈이다.

Auth 모듈은 인증장치 정보를 제공하는 GetInfo, 인증장치를 등록하는 Register, 서버에서 요청한 메시지를 서명하는 Sign 기능, 인증장치를 해지하는 Deregister 기능, 인증장치를 설정하는 OpenSettings 기능을 제공한다. Auth 모듈은 Auth Interface 블록, Auth Handler 블록과 User Verification 블록으로 구성된다. Auth Interface 블록은 ASM 모듈에 의해 호출되며 Auth 모듈에서 지원하는 기능을 제공한다. Auth Handler 블록은 Auth 모듈이 제공하는 기능 각각을 실질적으로 구현하는 핵심 모듈이다. User Verification 블록은 사용자가 패스코드를 등록하고 Register 와 Sign 기능을 처리할 때 사용자를 로컬 인증하는 기능을 제공한다.

Common 모듈은 ASM 모듈과 Auth 모듈에서 공통으로 사용되는 데이터베이스 관리 기능, 암호 연산 기능, 기타 유틸리티 기능을 제공한다. Data Manager 블록은 인증장치 등록, 인증 요청을 처리할 때마다 필요한 데이터를 조회하고 생성된 정보를 저장 관리하는 기능을 제공한다. Crypto 블록은 AES, SHA256, RSA, ECC 등 키 생성 및 서명 등에 필요한 암호 연산 기능을 제공한다. Utility 블록은 프로토콜 메시지 인코딩/디코딩, 안드로이드 파일 입출력, 안드로이드 Asset 관리 등 다양한 유틸리티 기능을 제공한다.

Admin 모듈은 패스코드 인증장치를 제어하는 기능을 제공한다. 인증장치의 설치 및 삭제, 패스코드를 설정, 사용자 검증 토큰의 유효기간 설정 등과 같은 기능을 제공한다.

3.2 주요 기능 설계

■ 등록/인증 Assertion

등록 Assertion 은 인증장치에서 인증장치 등록을 수행한 후, 등록을 요청한 FIDO 서버에 전달하는 인증장치 등록 정보이다. 등록 Assertion 은 TLV(Tag Length Value) 형식으로 구성되며, ASM 에서 base64url[6]로 인코딩하여 FIDO 서버에 전달된다.

등록 Assertion 은 9 자로 구성된 인증장치 고유 식별자인 AAID(Attestation Authenticator ID), 서명 알고리즘, 공개키 정보, 서버에서 제공한 Challenge 정보, 인증장치에서 지금까지 등록한 횟수, 인증장치에서 생성한 공개키 정보와 이들 정보의 신뢰성을 확보할 수 있는 서명 정보로 구성된다.

제조사에서 인증장치에 직접 설치한 Attestation Key 가 있는 경우에는 이 키로 서명 값을 생성한다. 만약 Attestation Key 가 없는 경우에는 인증장치 등록시 생성한 공개키에 대응되는 개인키로 서명 값을 생성한다.

인증 Assertion 은 FIDO 서버에서 요청한 내용에 대해 인증장치 등록시 생성한 개인키로 전자서명 값을 생성하여 FIDO 서버가 사용자를 인증하도록 하는데 사용된다. 인증 Assertion 은 등록 Assertion 과 같은 방식으로 ASM 을 통해 FIDO 서버에 전달된다.

인증 Assertion 은 9 자로 구성된 인증장치 고유 식별자인 AAID, 서명 알고리즘, 인증장치에서 생성한 nonce, 서버에서 제공한 Challenge 정보, 사용자에게 확인 받은 메시지가 있는 경우 메시지에 대한 해시 값, 서명에 사용되는 KeyId, 지금까지 서명한 횟수 정보와 이들 정보에 대한 전자서명 값으로 구성된다.

■ KeyHandle 과 KeyId

인증장치 등록을 수행하게 되면 인증장치에서는 사용자에 대한 공개키쌍을 생성한다. FIDO 에서는 인증장치가 시스템 리소스 제약이 많은 하드웨어 모듈로 구현되는 상황을 고려하여 개인키 정보를 인증장치에 저장하지 않고 인증장치에 함께 설치되는 대칭키인 WrapKey 로 암호화하여 ASM 에 저장할 것을 권고하고 있다. FIDO 에서는 암호화되기 이전 키 정보를 RawKeyHandle 로 정의하고 RawKeyHandle 을 WrapKey 로 암호화한 정보를 KeyHandle 로 정의하고 있다. 또한 인증장치는 KeyHandle 의 해시 값이나 Random 값으로 구성한 KeyId 를 ASM 에 전달하여 ASM 이 RawKeyHandle 과 KeyId 를 관리할 수 있도록 한다. ASM 은 KeyId 를 FIDO 서버에 전달하여 향후 FIDO 서버가 향후 인증/해지 등을 요청할 때 정확한 키 정보를 인증장치에 전달할 수 있도록 한다.

본 패스코드 인증장치는 KeyHandle 과 KeyId 를 다음과 같이 정의한다.

KeyId = Random(32)

RawKeyHandle = (KHAccessToken | UPrivateKey | UserName | KeyId)

$KeyHandle = E_{wk}(RawKeyHandle)$

■ 접근제어

패스코드 인증장치는 사용자 로컬 인증과 KHAccessToken(Key Handle Access Token) 두 가지 방식으로 접근을 제어한다.

인증장치는 인증장치 등록과 인증 요청시마다 사용자에 대한 로컬 인증을 수행한다. 로컬 인증은 사용자로부터 입력 받은 패스코드가 인증장치에 등록된 패스코드와 일치하는지 확인하여 수행된다. 만약 사용자 로컬 인증이 성공하면, Auth 모듈은 Random 값으로 구성된 사용자 검증 토큰을 발급하고 이를 ASM 에 전달한다. ASM 은 인증방치 등록과 인증 요청 명령 메시지를 생성하여 Auth 모듈에 전달할 때마다 사용자 검증 토큰 값을 명령 메시지에 설정하여 Auth

모듈이 사용자를 검증 결과를 확인할 수 있도록 한다.

KHAccessToken 은 인증장치 등록마다 생성되는 KeyHandle 에 대한 접근을 제어하는 토큰이다. KHAccessToken 은 AppID, PersonaID, ASMToken, CallerID 정보를 연결(concatenate)한 정보에 대한 해시 값으로 구성된다.

AppID 는 FIDO 서버에게 인증장치 등록을 요청한 Relying Party 의 아이디로 보통 해당 서비스의 URI 로 구성한다. PersonaID 는 인증장치가 설치된 OS 상에서의 사용자의 계정정보로 구성한다. ASMToken 은 ASM 모듈에서 random 하게 생성하여 관리하는 비밀 정보이다. CallerID 는 ASM 모듈을 호출한 FIDO Client 의 식별자로 안드로이드에서는 FIDO Client 앱을 서명하는 인증서의 해시 값으로 설정한다.

KHAccessToken 은 인증장치 등록을 요청한 Relying Party, FIDO Client 와 ASM 만이 Auth 모듈에서 생성한 Key Handle 을 접근할 수 있도록 해 준다.

4. FIDO 1.0 패스코드 인증장치 구현

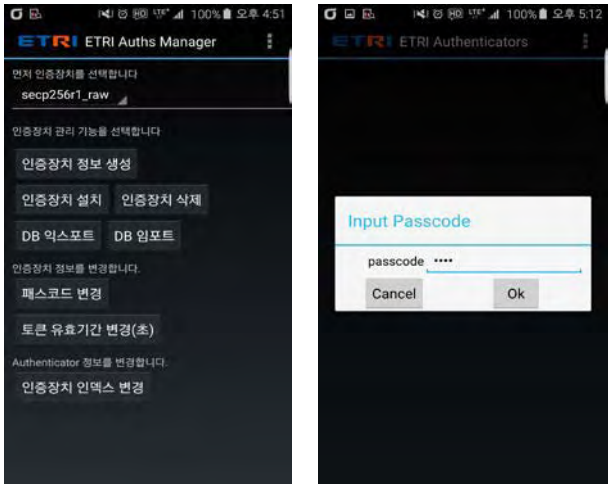
본 인증장치의 구현환경은 <표 1>과 같다.

<표 1> FIDO 1.0 패스코드 인증장치 구현 환경

항목	내용
운영체제	Android 4.0 이상(스마트폰) Window 7(PC)
개발툴	Eclipse Java EE Juno (4.4)
DB	Android Built-in SQLite
라이브러리	GSON

본 인증장치는 안드로이드 4.0 이상에서 동작하며, Window 7 PC 에서 Eclipse Java EE Juno 사용하여 개발하였다. DB 는 안드로이드에서 제공하는 SQLite 를 사용하며 프로토콜 메시지의 인코딩/디코딩을 위해 GSON 라이브러리를 사용하였따.

다음 그림 4 는 FIDO 패스코드 인증장치 실행 예를 보인다.



(그림 4) FIDO 패스코드 인증장치 실행 예

그림 4 의 왼쪽 화면은 인증장치 관리자 프로그램이다. FIDO 에서 권고하는 6 가지 공개키 알고리즘을 모두 지원하며 각각의 알고리즘 별로 인증장치를 생성할 수 있다. 인증장치 정보 생성은 인증장치 설치에 필요한 메타데이터, Attestation Key 등을 생성한다. 인증장치 설치는 인증장치 설치 정보를 이용하여 장치를 설치하는 기능을 수행한다. 인증장치 삭제는 설치된 인증장치를 제거한다. DB 익스포트, DB 임포트 기능은 인증장치에서 관리하는 DB 를 스마트폰으로 익스포트 또는 임포트하는 기능을 제공한다. 패스코드 변경은 사용자가 등록한 패스코드를 변경하는 기능을 제공한다. 토큰 유효기간 변경(초)은 사용자를 패스코드로 인증한 후 발급되는 사용자 검증 토큰의 유효기간을 설정하는 기능이다. 인증장치 인덱스 변경은 인증장치별로 할당된 정수 값인 인증장치 인덱스를 강제로 변경하는 기능을 제공한다.

그림 4 의 오른쪽 화면은 FIDO Client 를 통해 인증장치 등록과 인증 요청이 패스코드 인증장치로 전달되었을 때 사용자의 로컬 인증을 위해 사용자에게 패스코드 입력을 요청하는 화면이다.

5. 결론

본 논문은 FIDO 1.0 규격을 준용하는 패스코드 인증장치의 설계 및 구현에 대하여 기술하였다. 본 패스코드 인증장치는 추가적인 하드웨어 없이도 FIDO 1.0 에서 제시하는 높은 보안성과 동시에 사용자 편의성을 제공하고 있어 다양한 분야의 인증 기술로 활용될 것으로 예상된다. 향후 FIDO 기술을 플랫폼에 통합시키는 FIDO 2.0 규격이 개발되면 본 인증장치도 FIDO 2.0 을 지원하도록 개발하는 것이 필요하다.

참고문헌

[1] Joseph Bonneau et al., “The Quest to Replace Passwords: A Framework for Comparative Evaluation of Web Authentication Schemes, IEEE Symposium on Security and Privacy, 2012, pp.553–567.

[2] FIDO Alliance, http://fidoalliance.org

[3] Salah Machani et al., “FIDO UAF Architectural Overview”, FIDO Alliance, 2014, https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-overview-v1.0-ps-20141208.html

[4] Rolf Lindemann et al., “FIDO UAF Authenticator-Specific Module API”, FIDO Alliance, 2014, https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-asm-api-v1.0-ps-20141208.html

[5] Rolf Lindemann et al., “FIDO UAF Authenticator Commands v1.0”, FIDO Alliance, 2014, https://fidoalliance.org/specs/fido-uaf-v1.0-ps-20141208/fido-uaf-authnr-cmds-v1.0-ps-20141208.html

[6] The Base16,Base32,and Base64 Data Encodings. IETF. October 2006. RFC 4648. Retrieved March 18, 2010.