Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.2, Apr. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.2.373

OAuth2.0을 변형한 금융권 통합인증 프로토콜*

정 규 원,† 신 혜 성, 박 종 환‡
상명대학교

Integrated Authentication Protocol of Financial Sector that Modified OAuth2.0*

Kyu-Won Jung,† Hye-seong Shin, Jong Hwan Park‡
Sangmyung University

요 약

현재 국내 금융거래에서는 공인인증서를 중심으로 하는 다양한 사용자 인증방식이 사용되고 있다. 이러한 공인인증서 방식은 사용자가 개별 금융사의 웹서버에 접속할 때마다 서로 다른 보안모듈을 설치해야 하는 문제가 있다. 금융사 중심의 이러한 인증방식은 앞으로 생체인증 등의 차세대 인증방식이 새롭게 도입될 때마다 금융사마다 새로운 보안모듈을 추가적으로 설치해야 하는 문제가 발생한다. 이러한 문제를 해결하기 위해 본 논문에서는 금융거래 시 각 금융사를 대신하여 사용자 인증을 전담하는 통합 인증기관을 상정하고, 이를 중심으로 사용자 및 금융사 웹서버 삼자 간에 안전한 사용자 인증을 처리하는 통합인증 프로토콜을 제안한다. 새로운 인증 프로토콜은 OAuth2.0을 변형하여 안전성과 효율성을 증가한 프레임워크로써, 인증서버 및 금융사 웹서버 간 사전에 공유된 비밀키로 도전-응답 프로토콜을 수행하는 것이 특징이다. 이를 통해 사용자는 편리하고 안전한 통합인증(SSO: Single-Sign-On) 효과를 누리게 된다.

ABSTRACT

Currently, various types of user authentication methods based on public certificates are used in domestic financial transactions. Such an authorized certificate method has a problem that a different security module must be installed every time a user connects an individual financial company to a web server. Also, the financial company relying on this authentication method has a problem that a new security module should be additionally installed for each financial institution whenever a next generation authentication method such as biometric authentication is newly introduced. In order to solve these problems, we propose an integrated authentication system that handles user authentication on behalf of each financial institution in financial transactions, and proposes an integrated authentication protocol that handles secure user authentication between user and financial company web server. The new authentication protocol is a modified version of OAuth2.0 that increases security and efficiency. It is characterized by performing a challenge-response protocol with a pre-shared secret key between the authentication server and the financial company web server. This gives users a convenient and secure Single Sign-On (SSO) effect.

Keywords: Authentication framework, Integrated user authentication, Financial sector, SSO, OAuth2.0

I. 서 론

사용자 인증은 안전한 전자금융거래 환경에서 핵심적으로 선결되어야 하는 문제이다. 사용자는 금융사가 제공하는 웹서버에 본인임을 인증하고 금융사가 제공하는 웹서비스를 이용할 수 있다. 사용자를 인증하는 방법으로는 패스워드, 일회용 패스워드(OTP: One-Time Password), 인증서

Received(02. 27. 2017), Modified(03. 23. 2017), Accepted(03. 27. 2017)
* 본 연구는 2016년도 상명대학교 교내연구비를 지원받아 수행하였습니다.
† 주저자, jkw14g@naver.com
‡ 교신저자, jhpark@smu.ac.kr(Corresponding author)

(certificate), 생체인증 등이 있다. 현재 국내 금융거래에서는 인터넷뱅킹 등 온라인 금융거래 시 패스워드, OTP, 인증서를 주된 수단으로 이용하여 있으며, 온라인 쇼핑 등 전자상거래에서는 패스워드, 인증서, 생체인증 등으로 사용자를 인증하고 있다. 또한 금융거래의 중요성에 따라 단일(single factor) 인증이나 다중(multi factor) 인증을 사용한다.

특히 국내 금융거래에서는 공인인증서를 이용하여 사용자를 인증하므로 공인인증서 관련 보안모듈을 사용자 측에 추가로 설치해야 하는 문제가 있다[1]. 이를 위해 금융사 웹서버는 Active-X 형태나 exe 파일 형태로 보안모듈을 사용자 측에 설치하는 일이 빈번히 발생한다. 더 큰 문제는 각 금융사마다 독자적인 보안모듈을 설치하도록 강요하므로 사용자는 몇 개의 거래은행에 접속할 때마다 서로 다른 보안모듈을 설치해야 한다. 또한 이러한 보안모듈은 금융권뿐만 아니라 정부 및 그 산하기관에서 관리하는 웹서버에 접속할 때도 새롭게 설치해야 한다. 즉 공인인증서로 사용자를 인증하는 기관마다 독자적인 보안모듈을 시스템에 탑재하고 이를 사용자가 설치하도록 하는 번거로움이 존재한다. 이러한 문제는 생체인증과 같은 차세대 인증수단이 활성화될 경우 더 큰 문제를 야기한다. 즉, 공인인증서 이외에 별도로 생체인증 보안모듈을 각 금융사마다 준비하고 이를 사용자로 하여금 설치하도록 하는 문제가 발생할 것이다.

본 논문에서는 이와 같은 문제점을 완화하는 새로운 사용자 통합 인증 프로토콜을 제시하고자 한다. 새로운 프로토콜에서 각 금융사들은 사용자 인증업무를 SSO(Single-Sign-On) 인증서버에 위임한다. SSO 인증서버는 각 금융사의 요청이 있을 때마다 사용자를 대신 인증하며 그 인증 결과를 금융사에 돌려준다. 즉 사용자는 SSO 인증서버에 본인임을 인증하고, 그 결과에 따라 금융사 웹서버에 접속하게 된다. 이러한 인증 프레임워크는 현재 사용하고 있는 OAuth2.0[2] 프로토콜과 유사하다. 사용자 입장에서는 금융거래 시 접속하고자 하는 금융사 종류에 관계없이 하나의 SSO 인증서버에서 제시하는 인증수단 만 만족하면 되므로 SSO 효과를 누릴 수 있다. 또한 각 금융사별로 다른 보안모듈을 설치할 필요 없이 SSO 인증서버에서 요청하는 보안모듈만 관리하면 된다. 금융사 입장에서는 인증업무를 SSO 인증서버에 위임함으로써 금융사별 독자적인 인증기법을 도입하거나 시스템을 구축할 필요가 없게 된다.

제안하는 통합인증 프레임워크에서는 OAuth2.0을 그대로 차용하지 않고, 안전성을 강화하는 방향으로 개선한다. 먼저 SSO 인증서버와 각 금융사 웹서버는 비밀키를 교환하고, 이 키를 이용하여 사용자에게 발행한 인증토큰을 검증한다. 이 과정에서 SSO 인증서버와 웹서버는 일방향 인증을 하거나 상호인증을 할 수도 있다. 또한 OAuth2.0에서는 사용자 인증시마다 인증서버와 웹서버 간 통신이 발생했는데, 이는 대규모 사용자를 인증하는 경우 두 서버 간 통신이 과도하게 발생하는 문제가 있다. 제안된 프로토콜에서는 상호 공유된 비밀키를 이용하여 인증토큰을 검증함으로써 인증서버와 웹서버간의 추가적인 통신이 발생하지 않도록 개선한다.

II. SSO 프로토콜의 사전지식

2.1 SSO 프로토콜 구성 객체

제안하는 SSO 프로토콜은 사용자(user), SSO 인증서버(SSO authentication server), 그리고 각 금융사를 대표하는 웹서버들(web servers)로 구성된다. 각 주체별 역할을 다음과 같다.

- **사용자:** SSO 인증서버에 인증한 후 개별 금융사의 웹서버에 접속하는 자로 보통 웹브라우저 형태의 사용자 에이전트(user-agent)를 말함.
- **금융사 웹서버:** 사용자에게 금융사가 제공하는 웹서비스를 관리하는 시스템
- **SSO 인증서버:** 사용자 인증을 처리하고, 웹서버에 접근할 수 있는 인증토큰을 생성하는 시스템

이외에도 SSO 인증서버는 사용자 인증을 위해 SSO 통합사용자 DB를 구축한다. 이 경우 인증서버와 통합DB 서버 간 통신이 발생하는데, 이러한 통신은 안전한 전용망 또는 시스템 내부망을 통해 이루어진다고 가정하자.

2.2 SSO 프로토콜 객체 간 통신에 대한 가정

- **사용자 ↔ SSO 인증서버:** 사용자는 SSO 인증서버에 인증정보를 사전등록을 한 후, SSO 인증서버에 사용자 인증을 수행한다. 이 경우 두 객체는 인증서버가 제공하는 인증서로 SSL[1] 통신을 한다.

1) 현재 TLS v.1.2가 주로 사용되고 있으나, 본 논문에서는

- **사용자 ↔ 웹서버:** 사용자와 웹서버는 웹서버가 제공하는 인증서를 이용해 SSL 통신을 한다.
- **웹서버 ↔ SSO 인증서버:** 웹서버와 SSO 인증서버는 사용자 인증 이전 초기단계에서 비밀키를 교환한다. 사용자 인증 도중에는 HTTP 리다이렉션(redirection)을 통해 사용자를 경유하는 간접 통신을 한다. 이 경우에도 사용자와 인증서버 간, 그리고 사용자와 웹서버 간 SSL 통신을 이용한다.

2.3 사용자 인증에 대한 가정

사용자 인증 방법으로 패스워드, OTP, 사용자 인증서(Certificate), 생체인증, 일회용 도전-응답[2] (challenge-response) 등이 있다. 본 논문에서 제시하는 SSO 프로토콜은 원칙적으로 사용자 인증 방식에 제한을 두지 않는다.

SSO 인증서버나 웹서버는 사용자에게 정상적인 서버임을 인증하기 위해 인증서를 이용한다. 실제 SSL 통신에서는 서버인증서로 서버를 인증하고, 키 교환 및 안전한 통신을 수행한다. 그 안전한 채널 위에서 사용자 인증에 필요한 정보들이 암호화된 상태로 인증서버에 전송된다.

2.4 OAuth2.0 SSO 프로토콜

최근 SSO 인증의 형태로 OAuth2.0[2] 프로토콜이 사용된다. 흔히 소셜 로그인(social login)이라 불리기도 하는데, 페이스북이나 구글, 트위터 등의 계정을 이용하여 사용자를 대체 인증하는 방식이다. [그림 1]은 사용자가 쇼핑몰에 접속하고자 하는 경우, 쇼핑몰 서버에 패스워드로 인증하는 대신 몇 가지의 소셜미디어 계정을 통해서도 인증할 수 있는 OAuth2.0 예를 보여준다. OAuth2.0을 적용한 웹서버가 많아질수록 사용자는 하나의 계정으로 여러 개의 웹서버에 (사용자 인증 후) 접속하는 것이 가능해지므로 SSO 효과를 누릴 수 있다.

[그림 2]를 통해 OAuth2.0 프로토콜을 간단히 설명하면, (1) 먼저 사용자는 OAuth2.0이 적용된 웹서버에서 원하는 소셜 계정을 선택한다. (2) 사용

편의상 SSL 통신으로 표현한다.

2) 이 방법은 이동통신사를 이용한 SMS나 ARS, 또는 이메일 서버를 이용한 이메일을 통해 인증서버에 일회용 난수(도전값)를 전송하고, 그 난수를 받은 사용자는 그 난수(응답값)를 다시 시스템에 입력하여 본인임을 인증하게 된다.

Fig. 1. OAuth2.0-implemented web browser

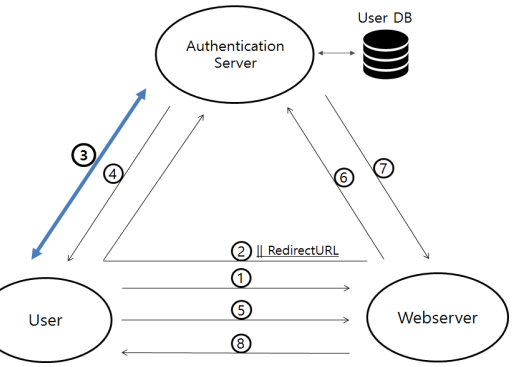

Fig. 2. OAuth2.0 authentication protocol

자가 해당 계정에 본인임을 인증하면, 인증서버(즉, 소셜 사이트에서 관리하는 인증서버)는 사용자에게 (권한인증을 포함한) 인증코드를 발행한다. (3) 사용자는 그 인증코드를 웹서버에게 전달하고, 그 인증코드를 받은 웹서버는 다시 인증서버에 인증코드를 전송한다. (4) 인증서버는 웹서버에게 받은 인증코드가 자신이 발행한 것과 일치하면, 인증토큰을 생성해서 웹서버에게 준다. (5) 웹서버는 인증토큰을 받으면 사용자 인증이 된 것으로 간주하고 사용자에게 접속을 허용한다.

위 과정에서 ⑥과 ⑦의 과정을 생략한 암시적(Implicit) 방식의 인증방식은 사용자의 인증토큰을 검증하지 않는 취약점이 발생하고 이를 악용한 형태의 공격이 가능하다[3,4,5].

III. 제안하는 SSO 프로토콜

프로토콜을 설명하기 위해 몇 가지 가정을 한다. (1) 각 사용자는 SSO 인증서버에 사전등록이 된 상태라고 하자. 각 금융사와 연계된 사용자 정보 및 인증에 필요한 정보 (예를 들어, 인증서, 생체인증 등) 등이 SSO 인증서버에 등록된 상태이다. (2) 각 금융사 웹서버는 (독자적으로 계정관리를 하는 것은 별도로 하고) 인증서버가 관리하는 통합사용자 DB와

연동되어 필요하다면 사용자들의 계정을 확인할 수 있다고 하자.

3.1 SSO 프로토콜 설명

0. SSO 인증서버와 각 금융사 웹서버간의 초기 설정 단계로 (1) 인증서버는 웹서버가 사용할 통합인증 API(Application Programming Interface)를 전송하고, (2) 웹서버는 해당 API를 사용하기 위한 비밀키 K를 설정하고 인증서버와 공유한다. K는 나중에 사용자 인증에서 MAC 기반의 인증토큰을 검증하는데 필수적으로 사용된다. (3) [그림 2]의 ②번 과정에서 필요한 HTTP 리다이렉션 URL 주소 및 관련 도메인 정보를 공유한다. 보안정책에 따라 안전성을 강화하기 위해 두 서버 간 공유된 비밀키는 주기적으로 갱신될 수 있다. 이 과정은 이후의 사용자 인증과정에서 반복되지 않는다.

1. 사용자는 금융거래를 원하는 금융사의 웹서버에 접속하여 로그인을 시도한다. [그림 4]는 기존 패스워드 기반 인증이외에 본 논문에서 제안하는 금융권 통합인증을 선택할 수 있는 화면이다. 사용자는 두 가지 중 하나를 선택할 수 있다. 특히 사용자는 통합인증을 선택할 때 자신이 원하는 인증방식을 선택할 수 있다. 여기서 사용자와 웹서버 간 통신은 웹서버 인증서를 이용한 SSL 통신으로 보호된다. 추후 금융권 통합인증이 활성화되어 인증업무가 각 금융사에서 SSO 인증서버로 대체되면, 각 은행별 ID/패스워드 인증도 사용빈도가 줄어들 것이다.

2. 사용자가 통합인증으로 특정 인증 방식을 지정하

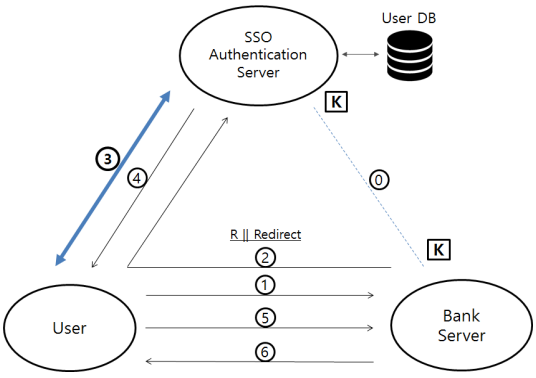

Fig. 3. Proposed SSO protocol

Fig. 4. Web server login via proposed SSO

면, 금융사 웹서버는 (1) 난수 R을 선택하고, (2) HTTP 리다이렉션 주소를 첨부하여, (3) 사용자를 SSO 인증서버로 안내한다. 여기서 난수 R은 웹서버가 생성한 도전값(challenge)으로 사용자 인증을 위해 필요하다. 나중에 ⑤번 과정에서 사용자가 보내온 값이 난수 R에서 대응하는 응답값(response)이 된다. 이제 사용자는 인증서버와 통신하게 되고, 이 과정 역시 인증서버가 보내주는 인증서를 통해 SSL 통신으로 보호된다.

3. 사용자가 SSO 인증서버에게 자신이 선택한 인증방식으로 자신을 인증하는 과정이다. 앞서 설명한 것처럼 사용자 인증을 위해서는 패스워드, OTP, 인증서, 생체인증 등의 방법이 있는데, 인증서버는 인증정책을 설정함으로써 이 방법들 중 한 가지를 이용하거나(즉, single factor 인증) 몇 가지를 이용해서(즉, multi factor 인증) 사용자를 인증할 수 있다.

- 패스워드 방식: 이것은 기존 OAuth2.0 방식과 동일하다. 그러나 금융거래의 특성상 SSO 인증서버에 패스워드를 사전등록 할 때, 패스워드를 거래계좌들과 연동하고 보안성이 강화된 패스워드를 설정하는 등의 정책이 필요하다. 제안 프로토콜에 의하면, 하나의 패스워드를 통해 등록된 금융사에 접근할 수 있는 장점을 가진다.

- OTP 방식: SSO 인증서버가 OTP를 발급할 경우, 제안된 프로토콜에서는 하나의 (통합) OTP[6]만으로 등록된 금융사에 접근할 수 있다. 이 방식도 현재 추진되고 있는 통합 OTP와 연동하는데 별 문제가 없다.

패스워드 인증과 OTP 인증은 실제 운용상에서 몇 가지 차이점이 있다. 첫째, 부가적인 기기 (또는 소프트웨어)가 필요한지 여부로써, 패스워드 인증은 별도의 기기가 필요 없으나, OTP는 인증서버와 사전에 비밀키값 및 OTP 생성모듈을 공유해야 함으로

OTP 기기 등이 필요하다. 둘째, 키보드 보안과 같은 입력값을 보호하는 보안모듈이 추가로 필요한지 여부로써, 패스워드 인증은 입력되는 패스워드가 시스템 내부로 안전하게 이동하도록 키보드 보안모듈이 필요하다. 그러나 OTP 인증에서는 한 번 사용된 패스워드는 그 다음 생성될 패스워드를 계산하는데 도움이 안 되므로 이전 패스워드가 노출되어도 안전하다. 따라서 OTP 인증에서는 일회용 패스워드를 보호하기 위한 별도의 입력장치 보안모듈은 필요하지 않다.

- 인증서 방식: 국내 환경에서 사용하는 공인인증서를 수용할 경우, Active-X 형태로 또는 exe 파일 형태로 사용자 측에 공인인증서 관련 보안모듈을 추가적으로 설치해야 한다. 그러나 제안된 프로토콜이 기존방식과 다른 점은 (1) 각 금융사와 관계없이 SSO 인증서버로부터 한번만 보안 모듈을 내려 받는 과정으로 충분하게 된다. 또한 (2) 각 금융사들은 공인인증서 관련 보안모듈을 독자적으로 관리할 필요가 없으며, 공인인증서를 통한 인증업무를 SSO 인증서버에 위임하면 된다.

사용자와의 금융거래가 완료되면 금융사는 부인방지(non-repudiation)을 위해 사용자에게 거래내역에 대한 전자서명을 요구할 수 있다. 인증서 방식을 사용할 경우 제안 프로토콜에서는 이미 사용자가 SSO 인증서버로부터 보안모듈을 받고 있으므로 전자서명을 생성하는데 큰 문제가 없다. 금융사 역시 사용자로 하여금 전자서명을 위한 보안모듈을 내려받도록 할 필요가 없다. 단 사용자가 생성한 전자서명을 검증할 수 있는 모듈은 필요하다.

- 생체인증 방식: 현재 활발하게 연구되는 생체인증도 사용자 인증의 한 가지 방식으로 포함될 수 있다. 지문, 홍채, 얼굴 인식, 목소리, 정맥 등의 생체정보를 활용한 방식이 모두 가능하다. 중요한 점은 사용자의 생체정보가 인증서버에 그대로 저장되는 것이 아니라, 암호화된 형태 등으로 인증서버가 생체정보를 알 수 없도록 저장되어야 한다는 것이다. 현재 사용 중인 FIDO(Fast IDentity Online)[7]의 경우 사용자와 사용자 기기 간, 그리고 사용자 기기와 인증서버 간 두 단계 인증을 거치게 된다. 따라서 사용자와 사용자 생체정보를 저장한 기기 간 인증이 선행되어야 하므로 사용자는 전용기기를 반드시 지녀야 한다는 단점이 있다. 그 대신 사용자의 생체정보가 인증서버에 직접 전달되지 않으므로 생체정보 보안을 확보할 수 있는 장점이 있다.

그러나 생체정보를 이용한 차세대 인증은 생체정보 보안을 확보하면서도 위와 같은 하드웨어 의존성을 탈피하는 범용 인증플랫폼[8] 방향으로 연구되어야 할 것이다. 그 이유는, 예를 들어, 가족 구성원이 TV 시청을 한다고 하자. 프로그램 중 원하는 아이템이 있어서 구매하고자 하는 경우 하드웨어 의존성이 없다면 공용 TV를 통해 구성원 각자가 생체인증을 하고 원하는 아이템을 즉시 구매할 수 있게 된다. 그러나 현재 FIDO라면 구성원은 자신의 전용기기를 항시 옆에 소지하는 경우에만 즉시 구매할 수 있게 된다. 이처럼 FIDO와 달리 범용 인증플랫폼에서의 생체인증이 가능하게 하려면, 즉, 사용자로부터 기기 의존성을 벗어나게 하려면, 암호화된 생체정보를 직접 인증서버에 등록하는 것이 필요하다. 여기서의 문제는 인증단계의 생체정보와 등록단계의 생체정보는 에러가 발생할 수 있고, 허용된 에러 범위 내에서는 인증이 성공되어야 한다는 것이다. 또한 인증서버에 등록된 생체정보는 암호화된 상태이고, 인증단계에서 암호화된 생체정보와 연산을 수행하여 사용자를 인증해야 한다. 즉 암호화된 두 데이터에 대해 연산이 수행되어야 한다는 것을 의미한다.

이를 해결하기 위해 암호학적 프리미티브 중 하나인 동형암호(homomorphic encryption)[9]를 이용하는 것도 고려할 수 있으나, 동형암호는 암호문 연산결과를 다시 복호화해야 하므로 SSO 인증서버가 동형암호 연산 결과를 복호화하는 과정에서 생체정보가 노출될 위험이 있다. 이와 달리 최근 연구되는 함수암호(functional encryption)[10]는 비밀키 보안이 유지되는 경우, 암호문과 비밀키를 복호화 연산해서 원하는 함수값 만을 도출하는 것으로 복호화 연산과정에서 생체정보가 노출되지 않는 장점이 있다[11].

4. SSO 인증서버가 사용자를 인증하면, 웹서버가 보내온 도전값 R에 대응하는 응답값을 생성한다. 이 응답값은 (1) 먼저 인증서버가 난수 r을 생성하고, (2) 리다이렉션 URL을 통해 웹서버를 확인하고 해당 웹서버와 공유한 비밀키 K를 가져오고, (3) 인증을 통과한 사용자 정보 ID를 확인해서, (4) 응답값으로 t=MAC(K, r||ID||R)을 구한다. 인증서버는 사용자에게 (t, r, ID)을 보낸다. 여기서 MAC 값 t는 웹서버에 대한 사용자의 인증토큰 역할을 한

다. 이 과정에서 인증서버와 사용자 간 통신은 여전히 인증서버가 보낸 인증서 기반 SSL 통신으로 보호된다.

5. SSO 인증서버로부터 (t, r, ID)를 받은 사용자는 그 값들을 금융사 웹서버로 전송한다. 여기서 사용자와 웹서버 간 통신도 웹서버의 인증서 기반 SSL 통신으로 보호된다. 실제 〔그림 2〕의 ④와 ⑤의 과정은 HTTP 리다이렉션 과정으로 한 번에 수행된다.

금융사 웹서버는 사용자가 보내온 (t, r, ID)값을 통해 사용자를 인증한다. 먼저 SSO 서버와 공유한 비밀키 K와 자신이 생성했던 난수 R, 전송된 (r, ID) 값을 이용해서 t'=MAC(K, r||ID||R)값을 구한다. 그 t'값과 전송된 t값을 비교해서 같으면 ID에 대응하는 사용자를 인증하고, 다르면 인증실패로 간주한다. 이 과정에서 사용되는 인증 토큰 검증은 〔그림 5〕에서 표현된 대칭키 기반의 도전-응답 프로토콜에 근거한다. 여기서 웹서버가 SSO 인증서버를 인증하는 일방향 인증을 사용하고, 사용자는 (인증서버와의 인증이 성공하면) 단순히 도전-응답값을 전송하는 역할만 담당한다.

6. 금융사 웹서버는 사용자 인증 성공여부에 따라 사용자에게 웹서버화면을 전송한다. 이 경우에도 웹서버는 서버인증서를 이용, 사용자와 SSL 통신을 수행한다. 이후 사용자는 웹서버에서 제공하는 서비스를 이용한다.

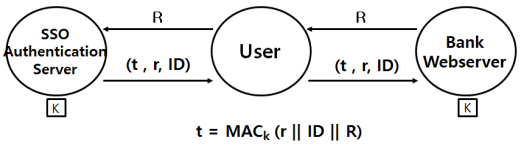

Fig. 5. One-way authentication between SSO server and web server

3.2 강화된 인증토큰 검증방법

사용자 인증 강화: 사용자 인증을 강화하기 위한 방법으로 사용자와 웹서버 간 도전-응답 프로토콜을 추가적으로 수행하는 것도 가능하다. 이 경우 사용자는 인증서버에게 받은 (t, r, ID) 중 (r, ID) 값만 웹서버에게 전송한다. 이를 받은 웹서버는 비밀키 K, 자신이 생성한 난수 R, 사용자에게 받은 (r, ID)을 이용해서 t'=MAC(K, r||ID||R)값을 구한다. 중요한 점은 사용자가 MAC 값인 t를 전송하지 않는다는 것이다. 이제 사용자와 웹서버는 각자 소유한 t와 t'을 대칭키로 해서 대칭키 기반의 도전-응답 인증 프로토콜을 수행한다. 여기서는 사용자만 (웹서버에게) 인증하는 일방향 인증도 가능하고, 사용자와 웹서버가 상호 인증하는 것도 가능하다. 이것은 시스템의 보안정책에 따라 결정될 것이다.

SSO 인증서버와 웹서버 상호인증: 위에서 기술한 인증토큰 검증에는 웹서버가 SSO 인증서버만 인증하는 일방향 프로토콜을 사용하였다. 반대로 인증서버도 웹서버를 인증하기 위해서는 인증서버의 도전값에 대해 웹서버가 응답값을 보내는 과정이 필요하다. 이 문제를 해결하기 위해 타임스탬프(timestamp)를 이용한다. 즉 타임스탬프를 SSO 서버의 도전값으로 보고 이에 대한 응답값을 웹서버가 보내는 것이다. SSO 인증서버와 웹서버는 상호인증을 두 번의 통신만으로 수행한다고 할 때, 이를 해결하는 방법으로 〔그림 6〕의 상호인증 프로토콜을 사용할 수 있다.

〔그림 6〕에서 웹서버는 난수 R을 생성하고 시간정보 time과 R을 상호 약속된 규격으로 포맷한 메시지 m을 만든다. 예를 들어 m=time||00…0||R이라 하자. 이 값을 대칭키 K로 암호화한 값 $\widetilde{R}$을 웹서버의 도전값으로 보낸다. 이를 받은 SSO 인증서버는 (사용자 인증과는 별도로) $\widetilde{R}$를 복호화한 후 메시지가 약속된 규격으로 나오는지 검사한다. 포맷 형식이 맞다면, $\widetilde{R}$를 보낸 주체가 K를 공유한 웹서버임을 인증한다. 이어서 인증서버의 난수 r을 생성하고 응답값으로 t=MAC(K, r||ID||$\widetilde{R}$||m)와 r, 사용자 ID를 전송한다. (t, r, ID)를 받은 웹서버는 t'=MAC(K, r||ID||$\widetilde{R}$||m)를 구하고 t와 비교해서 일치하면 SSO 인증서버임을 인증한다. SSO 인

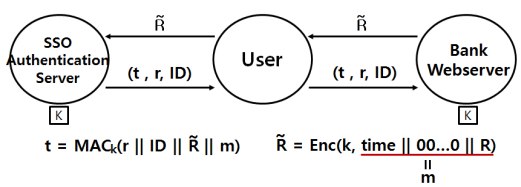

Fig. 6. Mutual authentication between SSO server and web server

증서버와 웹서버가 상호인증하는 과정에서도 안전성을 강화하기 위해 앞서 기술한 사용자 인증 프로토콜을 추가적으로 수행할 수 있다. 이 과정에서도 사용자는 웹서버에게 (r, ID)만 전송하고 t값을 비밀키로 사용해서 사용자와 웹서버 간 도전-응답 프로토콜을 수행한다.

IV. 제안된 SSO 프로토콜과 기존 방식 비교

제안된 금융권 통합대체인증 SSO 프로토콜이 구현될 경우, 현재 사용되는 사용자 인증방식과 어떤 차이점이 있는지 설명한다.

사용자 측면: 첫째, 사용자는 SSO 인증서버로부터 한번만 통합인증 보안 모듈을 내려 받으면 된다. 이에 비해 기존 방식에서는 금융사마다 독자적인 인증시스템을 구축하고 그에 따른 보안모듈을 사용자마다 내려 주는 불편함이 존재한다. 각기 다른 보안모듈 설치는 상호 다른 보안모듈간의 충돌 및 시스템의 성능저하로 이어질 수 있다. 둘째, 사용자는 자신이 선호하는 인증수단 한 가지로 서로 다른 금융사들에 접속할 수 있다. 이것은 현재 기존방식에서도 부분적으로 가능한 것으로써[12], 예를 들어 인증서로 인증하는 경우, 사용자가 금융사마다 자신의 인증서를 등록했다면 이후 하나의 인증서로 서로 다른 금융사에 접속할 수 있다. 여전히 사용자는 다른 금융사에 자신의 인증서를 등록하는 절차를 거쳐야 하는 불편함은 존재한다. 그러나 제안된 프로토콜에서는 금융권 통합 사용자 DB를 가정하므로, 사용자가 하나의 인증서를 SSO 인증서버에 등록할 때마다 사용자가 거래하는 금융사들[3]과 자동으로 연동될 수 있게 된다. 즉 금융사마다 별도로 인증서를 등록하는 과정은 필요 없게 된다.

금융사 측면: 첫째 금융사는 사용자 인증에 필요한 보안모듈 개발 및 배포 문제에서 벗어날 수 있다. 기존 기법에서는 각 금융사마다 독자적인 인증기법을 도입함으로서 그에 따른 보안모듈을 배포(예를 들어, Active-X 형태나 exe 파일 형태)하는 것이 여전히 필요하다. 이것은 OTP 기기나 소프트웨어를 배포하는 것에도 동일하게 적용된다. 별도의 보안모듈을 배포할 필요가 없으므로 사용자와 금융사 웹서버 간 통신은 SSL/TLS와 같은 웹 표준프로토콜로 간단히 구현 가능하다. 둘째, 제안 프로토콜 하에서는 금융사가 생체인증 등 차세대 인증방식을 도입하기 위해 필요한 시스템을 금융사 내부 시스템에 구축할 필요가 없다. 많은 비용 및 복잡한 시스템 구축이 필요한 인증업무를 SSO 인증서버에 위임할 수 있게 됨으로써 인증관련 비용을 획기적으로 절감할 수 있다.

SSO 인증서버 측면: 첫째, 금융권에 사용되는 사용자 통합 DB가 구축되어야 한다. 이것은 기존 방식에서는 없는 기관이지만, 현재 도입되는 통합계좌관리[13] 등의 서비스에서도 구축되고 있는 것으로서 현재 금융권에서 추진되는 방향과도 부합한다. 둘째, 각 금융사로부터 인증업무 대체수행에 따른 위임수수료를 받을 수 있다. 물론 정책적으로 이 수수료는 금융사가 직접 사용자 인증을 처리하는 비용보다 저렴하게 책정되어야 할 것이다. 이러한 수수료는 SSO 인증기관을 관리하고 유지하는 비용으로, 그리고 차세대 인증방식을 도입하는 비용으로 충당될 수 있다.

제안된 프로토콜에서는 하나의 SSO 인증서버를 운영하는 것으로 설계되었다. 이것은 하나의 인증서버가 마비될 경우 전체 금융거래시스템이 중단될 수 있음(즉, single point of failure)을 의미한다. 그러므로 실제 운영에서는 현재 구글 통합인증시스템이 동작하는 것처럼, 다수의 인증서버를 두고 업무를 배분하는 분산시스템(distributed system)을 구축함으로서 그 문제를 완화할 수 있다.

OAuth2.0 프로토콜: 현재 OAuth2.0을 도입한 대부분의 서비스 제공업체는 패스워드를 사용자 인증수단으로 도입하고 있지만, OAuth2.0 프로토콜은 원칙적으로 사용자 인증수단을 특정하지는 않는다. 마찬가지로 제안 프로토콜에서도 사용자와 SSO 인증서버 간 사용자 인증방식을 제한하지 않고 있으며, 향후 도입될 차세대 인증방식을 유연하게 수용할 수 있다.

차이점으로 OAuth2.0에서는 사용자로부터 인증값을 받은 후, SSO 인증서버와 서비스 제공업체가 추가적인 통신을 해서 사용자 인증을 완료한다. 이는 매번 사용자 인증이 수행될 때마다 인증서버와의 통신이 발생하는 것으로 SSO 인증서버와 서비스 제공업체 간 통신 부담이 크다. 그러나 제안 프로토콜에

3) 물론 사용자는 등록하는 인증서가 사용될 수 있는 금융사들(예를 들어, 거래계좌 등)을 선택적으로 지정할 수도 있다.

서는 인증서버와 금융사 웹서버가 사전에 비밀키를 교환하고, 그 키를 이용해서 사용자 인증을 금융사 웹서버 단독으로 완료하므로 SSO 인증서버와의 추가적인 통신이 요구되지 않는 장점이 있다. 또한 인증서버와 웹서버가 사전 공유된 비밀키로 상호 인증하는 프로토콜을 추가하여 안전성을 강화할 수 있다.

V. 결 론

본 논문에서는 금융권 사용자 통합인증을 제공하는 새로운 인증 프레임워크를 제시하였다. 새로운 인증 프로토콜은 OAuth2.0을 변형하여 안전성을 강화하는 방향으로 설계되었다. 가장 큰 차이점은 SSO 인증서버와 금융사 웹서버 간 비밀키를 사전에 교환하고 이 키를 통해 도전-응답 방식의 (상호)인증을 수행하는 것과, OAuth2.0에서 사용자 인증 시마다 발생하는 인증서버와 웹서버 간 통신을 없앨 수 있다는 것이다. 이를 통해 인증서버와 웹서버 간 통신 부담을 획기적으로 줄일 수 있다.

제안된 프로토콜 하에서 사용자는 SSO 인증서버에게만 자신을 인증하면 되고, 인증관련 보안모듈은 오로지 SSO 인증서버로부터만 제공받으면 된다. 따라서 현재 각 금융사마다 매번 다른 보안모듈을 설치하는 문제를 해결할 수 있다. 또한 금융사 웹서버는 인증관련 업무를 SSO 인증서버에 위임함으로써 인증기술 도입 및 시스템 탑재에 대한 부담을 경감할 수 있는 장점이 있다.

본 논문에서 제시한 인증 프레임워크는 차세대 인증방식에 대해 특정 인증방식을 한정하지 않고, 여러 인증 방식을 수용할 수 있으며, 온라인 쇼핑 등의 전자상거래 분야나 전자정부 등의 공공기관 업무 분야에서 사용자를 통합적으로 인증하도록 자연스럽게 확장될 수 있다.

References

[1] Jeong Gi Seog, "A Study on Measures for Improving Obligatory Use of Digital Certificate for Electronic Financial Transactions", Journal of Information and Security, Korea Information Assurance Society, Vol. 13, No. 6, pp.25-33, Dec. 2013

[2] D. Hardt, "The OAuth2.0 Authorization Framework", Internet Engineering Task Force(IETF) RFC 6749, 2012, https://tools.ietf.org/html/rfc6749#page-37

[3] T. Lodderstedt, Ed. Deutsche Telekom AG, M. McGloin, IBM, P. Hunt, Oracle Corp., "OAuth2.0 Threat Model and Security Considerations", Internet Engineering Task Force (IETF) RFC 6819, Jan 2013, https://tools.ietf.org/html/rfc6819

[4] Jun-Kyo Jung, Yong-Min Kim, "Secure Access Token Model of Open Banking Platform using Hash Chain", The Korean Society of Computer And Information, Vol. 24, No.2, pp. 277-280, Jul. 2016

[5] Hyung-Soo Park, Ki-Hyung Kim, "Enhanced OAuth Authentication with Security Code", Ajou University Graduate school, Dec. 2016

[6] Seung-Soo Shin, Kun-Hee Han, "A Study on Integrated ID Authentication Protocol for Web User", Journal of Digital Convergence in The Society of Digital Policy & Management, Vol. 13, No. 7, pp.197-205, Jul. 2015

[7] Submission Request to W3C: FIDO 2.0 Platform Specifications 1.0, https://www.w3.org/Submission/2015/02/

[8] Sangrae Cho, YoungSeob Cho, Soohyung Kim, "FIDO 2.0 Universal authentication technology Introduce" Korea Institute Of Information Security And Cryptology, Apr. 2016

[9] C. Gentry, "Fully homomorphic encryption using ideal lattices", Proceedings of STOC 2009, ACM, pp.169-178, 2009

[10] D. Boneh, A. Sahai, B. Waters, "Functional encryption: definitions and challenges", Proceedings of TCC 2011, Vol. 6597, LNCS, pp.253-273, 2011

[11] Minhye Seo, Jung Yeon Hwang, Soo-hy-

ung Kim, Jong Hwan Park, "Biometric Authentication Protocol using Hidden Vector Key Encapsulation Mechanism", Journal of the Korea Institute of Information and Communication Engineering, Vol. 26, No. 1, pp.69-79, Feb. 2016

[12] Jong Pil Yun, Jonghyun Kim, Kwangsu Lee, "Certificate-based SSO Protocol Complying with Web Standard", Journal of the Korea Institute of Information and Communication Engineering, Vol. 20, No. 8, pp.1466-1477, Aug. 2016

[13] Integrated management services for Account Information, https://www.payinfo.or.kr/payinfo.html

〈저자소개〉

정 규 원 (Kyu-won Jung) 학생회원
2017년 8월: 상명대학교 컴퓨터과학과 졸업예정
〈관심분야〉 인증 및 키 교환, 운영체제 보안

신 혜 성 (Hye-seong Shin) 학생회원
2017년 8월: 상명대학교 컴퓨터과학과 졸업예정
〈관심분야〉 암호 프로토콜, 인증 및 키 교환

박 종 환 (Jong Hwan Park) 정회원
1999년 2월: 고려대학교 수학과 졸업
2004년 2월: 고려대학교 정보보호대학원 석사
2008년 8월: 고려대학교 정보보호대학원 박사
2013년 9월~현재: 상명대학교 컴퓨터과학과 조교수
〈관심분야〉 함수 암호, 브로드캐스트 암호, 암호 프로토콜