Journal of Convergence for Information Technology
Vol. 10. No. 11, pp. 40-46, 2020
e-ISSN 2586-4440
DOI : https://doi.org/10.22156/CS4SMB.2020.10.11.040

다중 채널 기반 오픈 API 보안 프로토콜에 관한 연구

김상근
성결대학교 컴퓨터공학과 교수

A Study on Open API Security Protocol based on Multi-Channel

Sang-Geun Kim
Professor, Department of Computer Engineering, SungKyul University

요 약 금융권 공동 오픈 플랫폼 구축·서비스에 따라 스타트업 생태계에 안전한 보안 기술이 요구되고 있다. 금융권 표준 오픈 API는 상호인증 과정의 핵심 API 인증키 보호를 위해, 결제 관련 핀테크 기업이 추가 보안 기술을 개발/적용하는 것을 권고하고 있다. 본 연구는 다중 채널을 사용하는 강화된 API 보안 프로토콜을 제안한다. 기존 오픈 API 관련 연구의 문제점과 취약점을 추가 분석하고, 이기종 플랫폼의 호환성을 고려하여 설계되었다. 기존 보안 프로토콜의 단일 채널에 추가 보안 채널을 분리하여 은닉하는 방법을 적용했다. 성능 분석 결과 다중 채널의 통신 세션 양방향 안전성과 강화된 인증키의 중간자 공격 안전성을 확인하였으며, 다중 세션에서 지연시간의 연산 성능(1초 이하)을 확인하였다.

주제어 : 오픈 API, 오픈 뱅킹, 오픈 플랫폼, 오픈 인증, 금융 보안

Abstract Safe security technology is required for the startup ecosystem according to the construction and service of a joint open platform in the financial sector. Financial industry standard open API recommends that payment-related fintech companies develop/apply additional security technologies to protect core API authentication keys in the mutual authentication process. This study proposes an enhanced API security protocol using multiple channels. It was designed in consideration of the compatibility of heterogeneous platforms by further analyzing the problems and weaknesses of existing open API related research. I applied the method of concealment to remove the additional security channels into a single channel of the existing security protocols. As a result of the performance analysis, the two-way safety of the communication session of the multi-channel and the security of the man-in-the-middle attack of the enhanced authentication key were confirmed, and the computational performance of the delay time (less than 1 second) in the multi-session was confirmed.

Key Words : Open API, Open Banking, Open Platform, Open Authorization, Financial Security

1. 서론

국내외 금융기관은 개인정보 자기 결정권 보장, 금융 소비자 보호 등을 위한 금융 분야 마이데이터(MyData) 산업 도입을 추진하고 있다[1]. 사용자의 데이터 통합 서비스는 정보 주체가 사용자를 중심으로 전환되어 사업자는 개인정보 보호법과 개인정보 이동권을 정보 주체의 보편적 권리로 명시해야 한다[2]. 공통점은 데이터에 대한 적절한 보호, 이용 편의성, 데이터 투명한 관리 등을 위해 정보 요청/제공 방식이 표준화된 오픈 API를 활용하고 있다는 점이다[3]. 새로운 환경의 등장으로 인해 금융 및 결제 시장은 보안 인프라 및 개인정보 관리에 대한 문제가 이슈화되고 있다. 데이터 통합 서비

*Corresponding Author : Sang-Geun Kim(sgkim@sungkyul.ac.kr)
Received October 14, 2020
Revised November 6, 2020
Accepted November 20, 2020
Published November 28, 2020

스 구조 특성상 기존 개인정보 보호법과는 다른 성격의 정책을 반영하기 때문에 이에 대한 대응책이 요구되고 있다[4]. 금융보안원 오픈 API 보안 점검 가이드라인은 표준 API 보안 강화를 위해 각 사업에 참여하는 업체들이 추가적인 보안 기술 개발/적용할 것을 권고하고 있다[5]. 본 논문에서는 표준 API 인증키 보호를 강화하는 다중 채널 기반 보안 프로토콜을 제안한다. 본 논문의 구성은 다음과 같다. 2장 관련 연구에서 기존 오픈 API 표준과 보안 요구사항을 살펴보고, 기존 보안 프로토콜의 문제점을 살펴본다. 3장 제안하는 다중 채널 기반 오픈 API 프로토콜을 설명한다. 4장 성능 분석에서 보안성과 효율성을 검증하고, 5장 결론으로 마친다.

2. 관련 연구

2.1 오픈 API 표준과 보안 요구사항

국내 초기 핀테크 산업은 금융권 API 개방 정책을 지속 추진하였고, 다양한 핀테크 업체에서 이를 개발하고 활용하는 단계까지 왔다. 오픈 API는 표준에 근거하여 문서로 정의된 소프트웨어 간의 통신을 위한 인터페이스를 의미한다[6]. Fig 1과 같이 사용자가 제 3자 기관(핀테크 업체)을 통해 은행에서 필요한 권한을 허용하는 형태로 동작한다[7]. 금융권을 중심으로 모바일 앱을 연동하는 핀테크 서비스(지급 결제, 송금/전자화폐, 펀딩, 자산관리 등)가 큰 비중을 차지하고 있다[8].

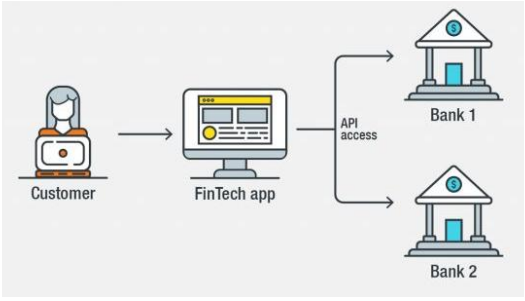

Fig. 1. Banking and Fin-tech Open API Access

금융권 오픈 API는 사용자의 중요정보에 대한 정보를 통합 활용하기 때문에 데이터 유출이나 변조 등 해킹 위협에 대해 노출될 수 있다[9]. 금융보안원은 이용기관 보안 점검을 위해 기술적 보안 유형에 개발보안/암호통제/접근통제/시스템 보안 위협 등 이용기관과 핀테크 서비스에 대한 보안 요구사항을 정의하고 있다[10]. 개발 보안 요구사항 핵심 요소는 오픈 API 프로토콜의 인증 및 접근키의 보호와 접근 권한 관리이다. 표준 오픈 API 키는 OAuth2(Open Authorization, Open Authentication 2)를 사용하여 인증과 권한인가 기능을 수행한다. 인증방식은 인증 코드 권한 부여(Authorization Code Grant Type) 방식 채택하여 사용하고 있다. Fig 2와 같이 OAuth2 보안 프로토콜은 기관, 사용자, 자체 인증에 모두 인증 토큰(Access Token)을 사용한다[11].

Fig. 2. OAuth2 Security Protocol Process

모든 사용자는 핀테크 서비스를 사용하기 위해 로그인을 수행한다. 지문인식, 아이디/패스워드 등을 활용할 수 있다. 정상 세션이 생성된 이후 내부 OAuth2 프로토콜의 인증 절차를 진행하게 된다. 서비스 요청 이후 접근 토큰을 획득하게 되며, 토큰의 세션 유효시간 이내 서비스를 사용할 수 있게 된다. 금융위 권고에 따라 핀테크 서비스 업체는 서비스에 자체적으로 개발한 보안 기술을 적용할 수 있다. 핀테크 분야는 본인인증, 정보 유출, 시스템 마비, 부정거래를 핵심 보안 위협으로 정의했다. 오픈 플랫폼 보안, 통신망 보호, PIN 번호 도입, 이상 금융거래 탐지 시스템 등 다양한 대응 방안을 적용해왔다[12].

2.2 오픈 API 보안 취약점 분석

오픈 API 보안은 OAuth2 프로토콜을 중심으로 위협 영역을 분류하면 크게 두 가지 영역으로 분류된다. Fig 3과 같이 사용자의 내부 응용 소프트웨어와 핀테크 기업 사이의 통신 구간이다[13]. 실제 사용자(User)의 핀테크 앱(웹 포함)은 핀테크 업체(Agency) 사이의 통신 구간의 안정성이 확보되어야 한다. 해커는 세션 생성 이후 정상적인 API 접근키와 인증키를 획득하고 재사용하여 인증 서버의 권한을 획득하는 것이 일반적인 해킹 시나리오이다.

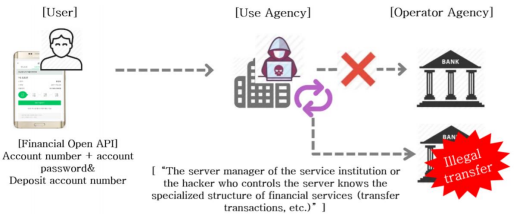

Fig. 3. Banking and Fin-tech Open API

Table 1은 국내 오픈 API 표준의 OAuth 보안 프로토콜 연구 사례의 문제점을 나타낸다.

Table 1. Comparative Analysis of Research problems

Proposed	Problem
B2B2C Model[13]	Server Construction cost Problem, difficulty in implementing the proposed Model
Hash chain based OTP[14]	OTP bypass technique exists(decryption, reverse engineering, reuse, etc.)
Blockchain based trading system[15]	Double Spending, Mining/Pool, Wallet Threat
Blockchain based business model[16]	POC(Proof of Concept) Stage: Lack of technology Maturity
POSCAL Certification Framework[17]	Difficulty in building/development of PKI infrastructure, Wallet Threat
IUWT-based token authentication[18]	New token method verification Problem, non-compliance with standard specifications
OAuth2.0 modified[19]	Server Construction Cost, Safety problem for SSO Server
Stateless token authentication[20]	New Token method verification Problem, non-compliance with standard specifications

OTP, 해시체인, 블록체인, IUWT, 비상태 토큰 연구 등 내부 토큰을 보호하는 방법이 주요 연구이다. 문제점 분석결과 OTP는 다양한 우회기법[21], 블록체인은 외부 환경의 취약성[22], POSCAL의 경우 PKI 기반 공인인증서의 범용성 부족[23] 문제점이 존재했다. 공통점에는 B2B2C 모델, POSCAL, IUWT, OAuth2 변형 연구 등 대부분 연구가 비용에 부담이 큰 인증 서버를 추가하는 형태로 설계되었다. 기타 연구들은 수년 이전의 프로토콜로 규격으로 현재 오픈 API 규격과 호환성이 떨어진다는 문제점이 존재했다. Bayram Doğan는 핀테크 서비스의 주요 해킹 취약점들이 표준 개발 사양에 적절하지 않은 OAuth 프로토콜 구현이 주요 원인이라고 설명했다[24]. IUWT나 OAuth2 변형과 같이 검증되지 않은 프로토콜은 표준 오픈 API와 호환성이 떨어져 신뢰성이 낮았다.

3. 다중 채널 기반 오픈 API

3.1 오픈 API 보안 요구사항 및 대응책

Table 2는 문제점 분석 이후 오픈 API 프로토콜 보안 요구사항을 나타낸다.

Table 2. Open API protocol security requirements

No.	Security List	Requirements
1	Core security area	Between users and fintech companies
2	Internal token security	Enhanced authentication token security
3	Standard protocol vulnerability	Complementing known bypass techniques
4	Functional modularization	Modular extension model design
5	Core Security verification	Local encrypted storage, MITM (Man-In-The-Middle attack)

본 논문은 현재 금융권에서 운영되는 서버 인프라와 통신 구간이 안전하다고 가정한다. 핵심 보안 영역은 사용자 로컬(월렛)과 핀테크 업체 사이의 통신 세션 보안이다. 오픈 API의 표준 프로토콜을 준수하고, 알려진 보안 취약점(우회, MITM 등)들을 보완하기 위한 프로토콜 추가와 인증 토큰의 강화가 요구되었다. 블록체인과 PKI 보안 위협으로 Double Spending, Mining/Pool, Wallet Threat 등은 본 논문의 오픈 API 프로토콜 보안 범위를 벗어나기 때문에 해결 항목에서 제외한다. Table 3은 보안 요구사항을 위한 해결책을 나타낸다. 현재 핀테크 서비스는 전용 APP(웹/앱) 내에서 비밀번호, 지문 생체정보, OTP(OTP, One Time Password) 기능을 제공하는 형태로 구현되어 있다.

Table 3. Open API Protocol Security Requirements

No.	Security Solution
1	Separation and hiding of Secure Session channels using Internal APP
2	Insertion of external Session passphrase (enhanced Authentication Token Security)
3	URL checks, Tokens updates, limited number of Tokens
4	Standard Security Protocol extension and Modularization
5	Security verification against MITM attacks

OAuth 표준 프로토콜은 핀테크 업체를 거쳐 금융권 인증 서버와 안전한 채널을 생성하게 된다. 제안 기법은 안전한 채널을 이중화하여 독립적으로 분리하는 방식을 적용한다.

3.2 다중 채널 프로토콜 전체 흐름도

Fig 4는 표준 오픈 API 동작 과정에 제안하는 대응 기법을 적용한 결과를 나타낸다.

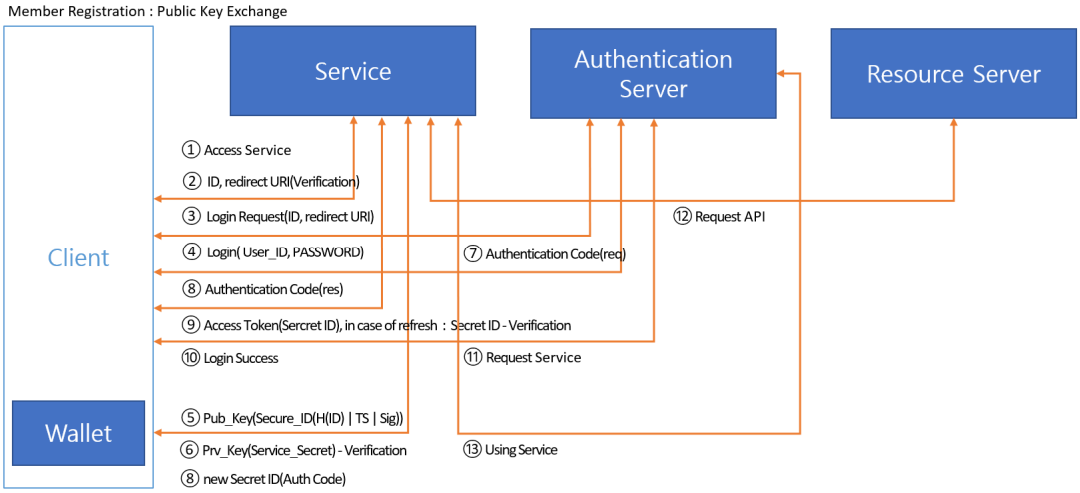

Fig. 4. Dual-channel Scheme Protocol Entire Process

① ② 서비스 접근 및 URI 검사 : 핀테크 서비스 접근과 함께 기존 알려진 취약 대응책으로 URI 검사 부분을 표기하였다. 모든 프로토콜 동작은 기본적으로 TLS 기반 암호화 프로토콜을 사용한다. - 3번 보안 요구사항

③ 로그인 요청 : 초기 생성된 고유 식별자(ID)를 핀테크 서비스를 거쳐 인증 서버로 전송하게 된다.

④ 로그인 : OAuth 표준 보안 프로토콜은 ID, PASSWORD 등 다양한 형태의 식별 파라미터를 전송하여 사용자 인증을 수행한다.

⑤ ⑥ 독립 채널 상호인증 : 사전(가입)에 교환된 공개키 쌍을 사용하여 사용자의 핀테크 서비스 간에 상호인증을 수행한다. 기존 프로토콜은 7계층 URL 기반의 통신을 수행한다. 새로 추가된 독립 세션은 핀테크 월렛 내부 4계층 TCP 통신으로 상호인증을 구현하였다. - 1, 5번 보안 요구사항

⑦ ⑧ 인증 코드 교환 및 암호화 : 정상적인 인증 코드를 교환한다. 비밀 식별자(Secret ID)를 생성하여 두 채널 사이의 암호화 파라미터로 사용한다. - 2번 보안 요구사항

⑨ 접근 토큰 교환 및 검증 : 주기적으로 외부 보안 채널(내부 앱)에서 토큰을 검증한다. 토큰 발급 개수 제한 및 짧은 토큰 갱신 시간을 적용했다. - 3번 보안 요구사항

⑩ 정상 로그인 완료

⑪ ⑫ 서비스 요청/응답 : 초기 과정에서 발급받은 오픈 API 접근키를 이용하여 서비스를 요청한다.

⑬ 정상 서비스 사용

Fig 5는 다중 채널 기반 프로토콜 과정의 특징을 나타낸다. 기존 오픈 API 프로토콜의 TLS 암호화 채널 전체가 안전하다고 가정할 수 없다. TLS 기술은 취약한 보안 설정, 알려진 sslstrip 공격 기법, 취약한 암호 알고리즘 선택 등 지속해서 해킹이 발생해 왔다. TLS 비활성화, 취약성으로 인해 초기 파라미터는 노출될 수 있다.

오픈 API 프로토콜에서 생성된 보안 세션을 독립된 보안 세션으로 안전성을 강화했다. TLS의 경우 서버의 보안 설정에 따라 암호 알고리즘 설정이 다르게 변화하지만, 제안 프로토콜은 비교적 안전하다고 알려진 알고리즘 조합(TLS 1.3 버전 기준)으로 동작을 강제한다. 초기 세션에서 생성된 접근 토큰 내부에 Secret ID를 포함하기 때문에, 접근 토큰이 주기적으로 갱신될 때마다 독립된 채널에서 상호 검증을 수행하는 방식이다.

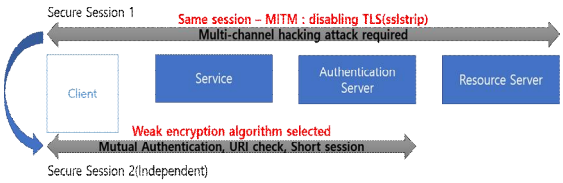

Fig. 5. Features of the Proposed Multi-Channel Security

4. 성능 분석

제안 기법은 기존 OAuth2 프로토콜을 지원하는 실제 리눅스 클라우드 서버를 구축하고, 웹 서버 부하 성능을 애플리케이션 단위로 비교 분석하였다. Table 4는 성능 분석에 사용된 환경설정을 나타낸다.

Table 4. Performance Analysis Environment

	Description
Server	Ubuntu Linux 16.04(x64), Apache Tomcat 8 Server
Framework, API Service	Spring Security OAuth2.0, Google Cloud Platform
Encryption, Hash Algorithm	ECDHE, RSA, SHA256
CPU, Memory	Intel(R) Core(TM) i7-5700Q CPU @ 2.70GHz, 16GB

기본 세션 설정은 Spring security OAuth2.0에 설정된 기본 오픈 API 알고리즘 사양을 준수했다. 독립 보안 세션은 Java 언어 Security 클래스에서 제공하는 타원곡선 디피헬만(Diffie-Hellman) 키 교환, RSA/ECB 상호인증, SHA 해시 알고리즘을 사용했다.

4.1 보안성 검증

표 5는 제안 프로토콜의 보안성 비교 분석결과를 나타낸다.

Table 5. Security Comparison Analysis

	OAuth2	Proposed
Multi Sesstion	Single Session	Session Multiplexing
Security token safety	SSL/TLS	SSL/TLS
	Session Key Dependent	ECDHE(Key Exchange)
		RSA/ECB(Mutual Authentication)
MITM Attack	Possibility of Stealing the Authentication Code of Session 1 (Redirect URI)	Session 1, 2 Associated Verification
		Limit the Number of Session Verifications
		Session Verification Time Limit

① 다중 세션의 양방향 안전성 : 초기 세션 1개의 접근 키에 대해 2개의 세션 키를 생성하여 검증하는 방식이다. 초기 세션이 노출되었을 때, 가입 초기 생성한 개인 키를 알아내야 한다. 독립적인 두 세션에 대한 개인키를 모두 복원해야 하는 강도를 지니기 때문에, 양방향 세션의 안전성 제공한다.

② 접근 키에 대한 안전성 : 접근 키 내부에는 공개키 기반으로 생성된 새로운 Secret ID을 포함한다. sslstrip 공격이 성공하여 실제 SSL/TLS 암호화가 무력화된 경우 Secret ID를 추가로 계산해야 한다. Secret ID를 계산하는 어려움은 기존 RSA 소인수분해의 어려움을 해결해야 한다.

③ 중간자 공격에 대한 안전성 : 다중 세션은 기존 중간자 공격에 방어하기 위한 다른 보안 채널을 추가 검증하는 방식이다. 중간자 공격을 위해서는 사용자 PC가 반드시 첫 서비스 요청 단계 이전부터 악성코드가 감염되어 스푸핑이 선행되어야 한다. 이는 초기 가입 단계부터 공개키 교환 및 통신 세션 파라미터를 모두 가로채야 한다. 또한, 제한된 횟수와 짧은 세션 검증 주기 내에 공개키를 교체해야 하는 어려움을 해결해야 한다.

4.2 다중 채널 성능 비교 분석

분석 항목은 연산 과정에 CPU 처리시간에 초점을 맞추어 실험을 수행하였다. 제안 프로토콜에서 추가된 주요 RSA 연산은 총 2회로 상호인증 1회와 세션 검증 1회이다. Fig 6은 RSA 키 1,024bit 길이 기준 세션 1(기존)과 세션 2(제안)의 총 100회 암호화/복호화 시간 변화를 비교 분석한 결과이다.

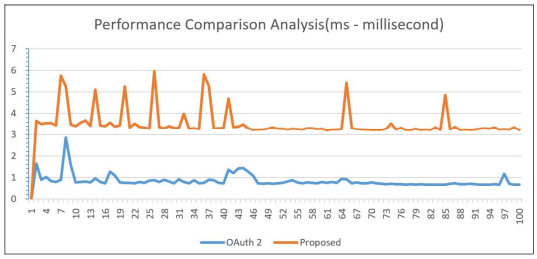

Fig. 6. Performance Analysis Result(Enc/Decrption)

단일 암호화/복호화에서 OAuth 프로토콜은 1초 기

준으로 변환했을 때 평균 0.0083초, 제안 프로토콜은 평균 0.0035초를 유지했다. 암호화/복호화 속도보다 내부 인증 토큰의 키 생성과 서명 생성에 추가적인 오버헤드가 크게 나타났다. 단일 세션 0.0353초는 체감될 정도의 지연시간은 아니다. Fig 7, 8, 9는 사용자 세션 개수 별 전체 처리시간 변화를 나타낸다. 100회 연산에 대하여 10, 50, 100세션으로 분리하여 처리시간의 오버헤드를 비교 분석하였다.

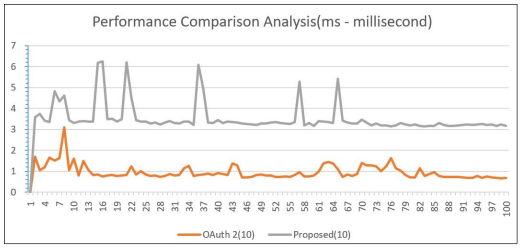

Fig. 7. Performance Analysis Result(10 Session)

10세션 기준 OAuth 프로토콜은 평균 0.0096초, 제안 프로토콜은 평균 0.0352초를 유지했다. 10 세션 수준까지는 처리 성능이 큰 차이가 없음을 나타낸다.

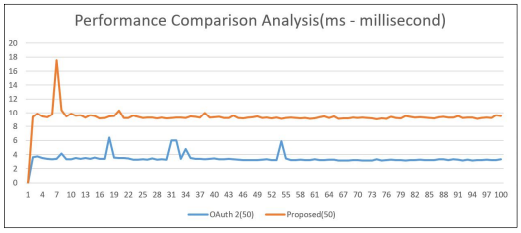

Fig. 8. Performance Analysis Result(50 Session)

50세션 기준 OAuth 프로토콜은 평균 0.0344초, 제안 프로토콜은 평균 0.0942초를 유지했다. 세션 증가로 인해 처리시간이 다소 증가함을 알 수 있다.

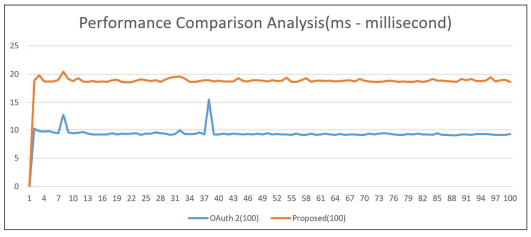

Fig. 9. Performance Analysis Result(100 Session)

100세션 기준 OAuth 프로토콜은 평균 0.0937초, 제안 프로토콜은 평균 0.1886초를 유지했다. 반응형 웹 전문가로 알려진 Jakob Nielsen에 따르면 웹 애플리케이션의 지연시간 1초 이하 속도는 사용자 경험(UX)에 긍정적인 영향을 미치는 것으로 알려졌다[22]. 개인 PC와 동일 사양의 실제 서버 하드웨어를 구축하고 운영했을 때, 500세션 이상 처리 가능함을 예상할 수 있다. 현재 서버 플랫폼의 경우 훨씬 좋은 H/W 사양들이 존재하기 때문에, 실제 서버를 구축/운영하면 충분히 더 많은 세션 처리가 가능할 것으로 예상한다.

5. 결론

국내 금융권은 최근 간편결제 시장의 확대와 함께 은행·저축은행·증권사의 예금된 본인 계좌를 직접 활용할 수 있는 다양한 결제 서비스를 제공되고 있다. 특히, 모바일 기반 생체정보 등록 및 PIN 번호를 인증하는 간편결제 서비스의 비중이 높아지고 있다. 그러나 이러한 생체인증 방법도 앞으로 취약 S/W와 H/W에서 유출된 생체정보 악용에 대한 대응책이 반드시 요구될 것이다. 현재 금융권에서 안전성을 검증받은 PKI 인프라는 대부분 암호화에 X.509 표준 RSA 알고리즘에 의존적이다. 오픈 플랫폼/오픈 API의 활성화는 생체인증이나 RSA 이외 안전한 프로토콜과 함께 새로운 알고리즘 조합들이 요구될 것으로 예상한다. 본 논문은 오픈 API 환경에서 하위 수준의 보안 프로토콜 수준의 취약점을 분석하고, 다중 채널 기법을 제안했다. 성능 분석 결과, MITM 공격에 대한 안전성을 확인(기존 잘 알려진 해킹 기법에 노출되었다고 가정)하였다. 또한, 제안한 다중 세션의 실제 서버의 TPS(Transaction per second) 처리 부하 영향도를 분석하여 암호 알고리즘과 키 길이 선택에 대한 효율성 지표를 제공했다.

REFERENCES

[1] Yi, M. (2020). Comparison of MyData Use Among the US, Europe, and the Korean Governments. *Journal of the Korean BIBLIA Society for library and Information Science, 31(2),* 183-201. DOI: 10.14699/kbiblia.2020.31.2.183

[2] J. H. Park. Activation of My Data System and Legal Issues. *Law Research Institute of Ajou University, 14(1),* 96-119. DOI : 10.21589/ajlaw.2020.14.1.96

[3] J. A. Park. (2020). Study on methods for establishing legislation on data protection and

distribution. *The Institute for Legal Studies, Sogang University, 9(2),* 3-41.
DOI : 10.35505/slj.2020.06.9.2.3

[4] M. J. Song & I. S. Kim. (2019). A Study on Privacy Protection in Financial Mydata Policy through Comparison of the EU's PSD2. *Journal of The Korea Institute of Information Security and Cryptology, 29(5),* 1205-1219.

[5] Financial Security Institute. (n.d).. *Convergence Security Department Fintech Security Team.* Guide(Onlnie). http://www.fsec.or.kr/

[6] J. H. Seo. (2018). Innovation strategy of the domestic banking industry through activation of open API. *Korea Institute of Finance*, 1-60.

[7] Feike Hacquebord at el. (n.d.). *When PSD2 Opens More Doors: The Risks of Open Banking,* Trend Micro. Cyber Threats. https://blog.trendmicro.com/

[8] J. H. Na & J. C. Na (2018). Open platform standardization trend for safe fintech service. *Korea Institute Of Information Security And Cryptology, 28(4),* 13-17.
UCI : I410-ECN-0101-2018-004-003408438

[9] I. S. Kim. (2018). Financial security and countermeasures for the financial sector in response to changes in the fintech environment. *Korea Federation of Banks Financial webzine, 732,* 6-13.

[10] Financial Security Institute. (n.d.). *Security check related to open banking (main contents),* Financial Security Institute(Online). https://www.fsc.go.kr/

[11] https://developers.open-platform.or.kr

[12] J. E. Kim, I. S. Kim. (2017). A Study on the Liability of Information Protection for the Third Party Supply of Personal Information/Focus on Fintech Companies Using OPEN APIs. *Journal of Korea Society for e-Business Studies, 22(4),* 21-38.
UCI(KEPA) : I410-ECN-0101-2018-004-001571185

[13] D. H. Choi, I. S. Kim. (2019). A Study on the Policy Proposal and Model B2B2C for Safe Open Banking. *Journal of The Korea Institute of Information Security and Cryptology, 29(6),* 1271-1283.
DOI : 10.13089/JKIISC.2019.29.6.1271

[14] J. K. Jung, Y. M. Kim. (2016). Secure Access Token Model of Open Banking Platform using Hash Chain. *The Korean Society Of Computer And Information Proceedings of the Korean Society of Computer Information Conference, 24(2),* 277-280.

[15] M. S. Son, H. Y. Kim. (2020). A Real Estate Lease Transaction System Using Blockchain and Open Banking API. *Journal of Korean Institute of Information Technology, 18(5),* 109-119.
DOI : 10.14801/jkiit.2020.18.5.109

[16] K. J. Jang. (2017). A Study on Business Application of Payment System using BlockChain Technology. *Global e-Business Association, 18(6),* 113-130.
DOI : 10.20462/TeBS.2018.12.19.6.349

[17] S. M. Yoo et el. (2018). POSCAL : A Protocol of Service Access Control by Authentication Level. *Journal of The Korea Institute of Information Security and Cryptology, 28(6),* 1509-1522.
DOI : 10.13089/JKIISC.2018.28.6.1509

[18] H. B. Kang, H. C. Jang, C. S. Jang. (2019). IUWT Based Token Authentication Technology. *The Journal of Korean Institute of Information Technology, 17(2),* 143-150.
DOI : 10.14801/jkiit.2019.17.2.143

[19] K. W. Jung, H. S. Shin, J. H. Park. (2017). Integrated Authentication Protocol of Financial Sector that Modified OAuth2.0. *Journal of the Korea Institute of Information Security & Cryptology, 27(2),* 373-381.
DOI : 10.13089/JKIISC.2017.27.2.373

[20] B. C. Lee. (2018). Stateless Randomized Token Authentication for Performance Improvement of OAuth 2.0 MAC Token Authentication. *Journal of the Korea Institute of Information Security & Cryptology, 28(6),* 1343-1354.
DOI : 10.13089/JKIISC.2018.28.6.1343

[21] B. D. Göçer and Ş. Bahtiyar, (2019, September). An Authorization Framework with OAuth for FinTech Servers. *In 2019 4th International Conference on Computer Science and Engineering (UBMK)* (pp. 536-541). IEEE.
DOI: 10.1109/UBMK.2019.8907182.

[22] Jakob Nielsen. (n.d.). *10 Usability Heuristics for User Interface Design.* Nielsen Norman Group (Online). https://www.nngroup.com/articles/ten-usability-heuristics/

김 상 근(Sang-Geun Kim) [정회원]

· 1996년 2월 : 중앙대학교 컴퓨터공학과 (공학박사)
· 2003년 - 2004년 : Sydney University Visiting Scholar
· 1996년 3월 ~ 현재 : 성결대학교 컴퓨터공학과 교수

· 관심분야 : 정보보안, 핀테크, 빅데이터, 소프트웨어공학
· E-Mail : sgkim@sungkyul.ac.kr