Journal of the Korea Academia-Industrial cooperation Society
Vol. 20, No. 11 pp. 469-478, 2019

https://doi.org/10.5762/KAIS.2019.20.11.469
ISSN 1975-4701 / eISSN 2288-4688

# 카 쉐어링 클라우드 환경에서 최적화된 바이오 정보 기반 보안 기법 설계

이광형[1], 박상현[2*]
[1]서일대학교 소프트웨어공학과, [2]숭실대학교 컴퓨터공학과

# Design of Secure Scheme based on Bio-information Optimized for Car-sharing Cloud

Kwang-Hyoung Lee[1], Sang-Hyeon Park[2*]
[1]Department of Software Engineering, Seoil University
[2]Computer Engineering, Soongsil University

**요 약** 카 쉐어링 서비스는 경제위기 이후 실용적 소비패턴 의식의 확산과 환경의식의 고취, 스마트폰 확산을 통한 서비스 이용 편의성이 증가됨으로 인해 새로운 대중교통으로 자리매김을 하고 있다. 시장이 발전하고 많은 사람들이 이용하고 있지만 그에 대한 보안은 확실히 이뤄지지 않고 있다. 차를 이용하기 위해선 단지 ID와 PW로 로그인만 하게 되면 차량을 렌트하고 제어할 수 있어 피해가 예상된다. 본 논문에서 제안하는 프로토콜은 지문정보를 이용하여 카 빅데이터가 등록되어 있는 다양한 Service Provider Cloud을 브로커를 통해 사용자에게 최적화된 서비스와 간편하고 강력한 인증을 제공하고자 한다. 제안한 기법을 이용하면 바이오정보의 노출을 줄일 수 있고, 하나의 브로커를 통해 다수의 Service Provider Cloud로부터 서비스를 받을 수 있다. 또한 기존 카쉐어링 플랫폼 대비 모바일 디바이스에서 공개키 연산 및 세션키 저장량을 20% 가량 낮췄고, 간편하고 강력한 인증을 제공하고 보안채널을 구성하기 때문에 안전한 통신을 할 수 있다. 향후 카쉐어링 서비스 클라우드 환경에서 본 논문에서 제안한 기법을 통해 안전한 통신과 사용자의 편의성을 증대 시키기를 기대한다.

**Abstract** Car-sharing services have been settled on as a new type of public transportation owing to their enhanced convenience, expanded awareness of practical consumption patterns, the inspiration for environmental conscientiousness, and the diffusion of smart phones following the economic crisis. With development of the market, many people have started using such services. However, security is still an issue. Damage is expected since IDs and passwords are required for log-in when renting and controlling the vehicles. The protocol suggested in this study uses bio-information, providing an optimized service, and convenient (but strong) authentication with various service-provider clouds registering car big data about users through brokers. If using the techniques suggested here, it is feasible to reduce the exposure of the bio-information, and to receive service from multiple service-provider clouds through one particular broker. In addition, the proposed protocol reduces public key operations and session key storage by 20% on mobile devices, compared to existing car-sharing platforms, and because it provides convenient, but strong, authentication (and therefore constitutes a secure channel), it is possible to proceed with secure communications. It is anticipated that the techniques suggested in this study will enhance secure communications and user convenience in the future car-sharing-service cloud environment.

**Keywords :** Car-Sharing, Authentication, Big Data, Cloud-Service, Broker

본 논문은 2019년도 서일대학교 학술연구비에 의해 연구되었음.
*Corresponding Author : Sang Hyeon Park(Soongsil Univ.)
email: shyeon15@gmail.com
Received September 26, 2019 Revised October 18, 2019
Accepted November 1, 2019 Published November 30, 2019

## 1. 서론

카 쉐어링 서비스는 자동차와 IT 산업간 컨버젼스 서비스이다[1]. 자동차에 무선통신, 결제 서비스 등을 결합함으로써 365일 24시간 무인서비스가 가능해지면서 카쉐어링 서비스가 본격화 되었다. 카 쉐어링 서비스는 카 빅데이터가 등록되어 있는 Service Provider Cloud로부터 회원들이 자동차가 필요할 때마다 시간별로 예약을 통해 차량을 공동으로 이용할 수 있는 초 단기 차량렌트 사업이다. 카쉐어링 서비스는 새로운 대중교통의 수단으로 떠오르고 있으며 이미 많은 사용자들이 이용 중이다. 카 쉐어링 서비스는 친환경 소비심리, 실용적 소비패턴 의식의 확산, 비용절감, 스마트폰 확산을 통한 서비스 이용의 편리함 등의 이유로 시장이 커지고 있고 북미와 유럽을 중심으로 빠르게 성장하고 있다[2]. 이런 시장의 흐름에 반해 보안적인 취약점이 존재한다. 카쉐어링 서비스는 회원 가입 이후 차를 예약 하거나 이용할 때 단순 ID/PW방식을 이용하여 사용자를 인증하는 방식을 사용한다. ID/PW 기법의 취약점은 사용자는 자신의 익숙한 ID/PW를 이용하고 의미를 지닌 단어를 이용하여 설정한다는 점을 통해 유추 혹은 무차별 대입공격에 취약하고, 취약점을 감안하여 Password를 설정한다 해도 지속적인 Password갱신과 관리가 잘 이루어지지 않는 경우가 대부분이다[3]. 또한 사용자의 관리가 철저하더라도 악의적인 공격자에 의해 만들어진 파밍 사이트나 스니핑, 피싱을 통한 비밀번호 유출이 일어날 수 있다는 취약점이 존재한다[3]. 사용자의 계정이 탈취 된다면 악의적인 공격자가 정당한 회원의 계정을 이용해 자동차 서비스를 이용하여 계정 생성시 연결된 카드에서의 요금 결제로 인한 피해부터, 사고가 발생할 시 막대한 금전적 피해가 생길 수 있다. 최근 이러한 문제점을 해결하기 위하여 바이오정보를 이용한 다양한 인증기법이 제안되고 있지만 바이오정보는 변하지 않는 유일한 값이기 때문에 유출이 되어도 변경 될 수 없다는 문제점을 가지고 있다[4]. 따라서 바이오정보를 노출시키지 않고 악의적인 사용자로부터 안전하게 서비스를 제공하고 정당한 사용자에게 정상적인 서비스를 제공할 수 있도록 기존 ID/PW 방식을 대체할 수 있는 보안 기법이 필요하다. 본 논문에서는 카 빅데이터가 등록되어 있는 다양한 Service Provider Cloud로부터 정당한 사용자가 서비스를 제공받을 수 있도록 사용자의 바이오정보를 기반으로 안전한 서비스를 제공할 수 있는 보안기법을 제안한다. 또한 브로커를 기반으로 바이오정보의 안전을 제공하고 사용자의 편의성에 최적화된 카쉐어링 서비스를 제공할 수 있도록 하였다.

본 논문의 구성은 다음과 같다. 2장에서는 본 논문의 주제가 되는 카쉐어링 서비스와 VCC Architecture의 보안 요구사항에 대해 설명하고, 3장에서는 본 논문에서 제안하는 카쉐어링 아키텍쳐 및 상호 인증에 대해 자세히 기술한다. 4장에서는 제안하는 scheme에 대한 보안 평가, 컴퓨팅 자원 분석을 하고, 5장에서는 Discussion을 한다.

## 2. 관련연구

### 2.1 카쉐어링 아키텍쳐

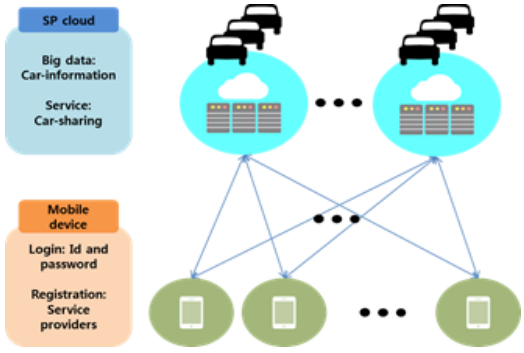


Fig. 1. Car-Sharing Architecture

카쉐어링 서비스는 시간단위로 자동차를 빌리는 서비스로 초 단기 렌터카 서비스로 분류된다. 초기의 카쉐어링 서비스는 커뮤니티 차원의 운동에서 시작되었다. 지역 또는 단체에 속한 구성원들이 매우 간단한 차량 이용을 목적으로 개별 차량을 소유하는 것은 낭비라는 인식에서 출발하여 공동의 차량을 시간단위로 사용하는 방식이 시초이다. 하지만 현실적으로 지역공동체나 단체차원에서 차량을 구매하고 관리하기가 어려워 대부분 카쉐어링 서비스는 전문적으로 관리하는 업체나 렌터카 업체를 통해 이뤄지고 있다[5].

일반적으로 카쉐어링 서비스 인프라는 Fig. 1과 같이 SmartCar, User, Service Provider Cloud로 구성되어 있다[5-7]. SmartCar는 유저가 예약을 통해 사용할 수 있다. SmartCar는 운행정보, 거리정보, 사용내역, 사고정보 등을 Service Provider Cloud에 전송한다. SmartCar는 네트워크를 통해 직접 Service Provider Cloud와 통신하거나 Road Side Unit을 통해 Service Provider Cloud와 통신한다. User는 카쉐어링 업체에 회원가입을

한 뒤 디바이스를 통해 차량을 예약하고 디바이스를 이용해 차량을 제어하게 된다. Service Provider Cloud는 차량의 예약정보, 차량 상태정보, 사용자의 주행정보, 요금정보 등을 저장하고 있다. Service Provider Cloud는 차량에게 예약자 정보를 전송하여 차량이 유저의 디바이스와 통신하여 유저가 스마트카를 제어할 수 있게 한다.

## 2.2 VCC 아키텍쳐

Vehicle Cloud Computing 는 다양한 네트워크의 융합 기술이다. VCC는 총 3개의 계층으로 나뉜다[8]. Inside-Vehicle 계층, Communication 계층 그리고 cloud계층이다. Fig.2번과 같이 첫번째 계층에서는 차량 내에 탑재된 각종 센서와 저장공간, 헤드유닛 등으로 구성된다. 통신계층에서는 V2V통신, V2I통신을 담당하게 되고 Cloud계층에서는 cloud기반구조, cloud서비스, application 계층으로 이루어져있다. VCC의 가장 큰 장점은 데이터를 클라우드 저장공간에 모아 빅데이터화 하는 것과 빅데이터화 된 데이터들을 클라우드에서 massive하고 복잡한 계산을 빠른시간에 해내는 것이다.

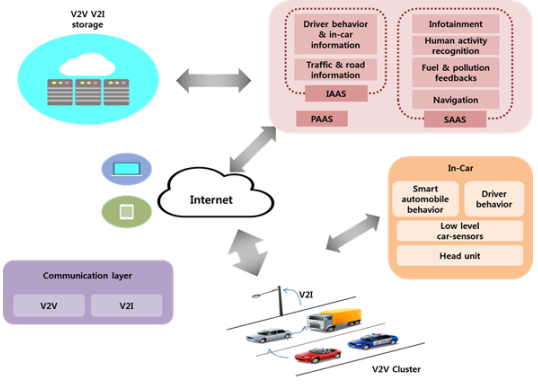


Fig. 2. VCC Architecture

## 2.3 보안 요구사항

VCC를 통한 카쉐어링 환경은 기존의 카쉐어링 환경과 다른 다양한 보안 위협이 존재하고 있다. Security, Integrity와 VCC를 통한 서비스를 제공하기 위하여 서버 가상화에서의 보안이 제공 되어야 한다. 또한 각기 다른 네트워크 환경과 이종 클라우드 플랫폼간의 통합이 중요하다.

### 2.3.1 Multi-Platform

VCC를 통한 카쉐어링 환경에서 각기 다른 VCC가 각자 수집한 데이터포맷, 사용자 정보 등이 다르기 때문에 이종의 플랫폼에서 각자 다른 포맷의 정보들로 인해 보안문제가 발생할 수 있다. 현재 멀티플랫폼 환경을 개선시키기 위한 노력과 연구들이 있지만, 아직까진 미미한 수준이다. 이러한 VCC환경에서 서로 다른 서비스 프로바이더들과 사용자간에 유기적으로 통신할 수 있어야 한다.

### 2.3.2 Security

카쉐어링 환경이 발전함에 따라 이에 비례하여 보안 위협도 증가하였다. 특히 카쉐어링 환경은 스마트카를 여러명이 빌려 쓰는 서비스이기 때문에 다양한 보안 위협을 가지고 있으며, 스마트카가 악의적인 공격자로부터 공격을 받게 된다면, 금전적 손해 뿐만 아니라 운전 능력이 없는 공격자에게 탈취당해 불특정 다수의 생명이 위협받을 수 있다. 악의적인 공격자에게 노출이 되지 않기 위해 반드시 상호 인증이 사전에 이뤄진 통신을 해야 하고 상호 인증 이후에도 replay attack, Man in the Middle attack, forward security 등의 공격으로부터 안전하도록 보안채널을 구성해야 한다.

### 2.3.3 사용자 편의성

카쉐어링 서비스에서 사용자의 편의성은 중요하다. 기존의 서비스는 한 사용자가 차량을 예약할 때 동승자 중에 운전자를 따로 설정할 수 있게 되어있다. 사용자가 목적지가 같고 항상 같이 사용을 할 경우에는 문제가 없다 하지만 목적지가 다르거나 예약한 사용자는 다른곳에 있고 동승자만이 따로 움직일 때에는 차량을 제어할 수 없게 된다. 그렇다고 잠시 떨어져 움직일 때를 위해서 차량 두개를 빌리게 된다면 예약한 사용자에게는 낭비이고, 다른 잠재적인 사용자도 원할 때 서비스를 받지 못할 경우가 생기게 되기 때문에 사용권의 안전한 양도 또한 고려되야 할 부분이다.

## 2.4 Previous Research

이번 장에서는 논문에서 제안한 scheme에서 신뢰관계에 있는 VCC와 smartcar의 인증이 관련된 이전 works와 바이오인증에 대한 works들을 리뷰한다.

He Yuan Huang et al.[9]은 클라우드에서 Single Sign On를 제공하는 브로커를 이용한 아키텍쳐를 제안하였다. Identity Federation 브로커는 Trusted Third Party를 통해 서비스 프로바이더의 클라우드들에게 신뢰

성을 확보 하고 사용자 중심의 방법을 통해 Identity Federation을 쉽게 관리한다. 또한 OpenID를 이용하여 Single Sing on 서비스에 가입된 클라우드를 하나의 클라우드에 가입해 서비스를 받을 수 있다. 제안하는 아키텍에서는 보안에 대한 다른 사항을 다루지 않고 HTTP over TLS/SSL을 이용한다.

Jaekyung Lee et al.[10]은 멀티 클라우드 환경에서 사용자에게 서비스의 투명성을 제공하는 인증기법을 제안하였다. 제안한 프로토콜에서는 사용자에게 투명한 서비스와 편의를 제공하기 위해 서비스제공자가 인증과정을 거치도록 설계 하였다. 사용자는 한번의 인증을 통해 다른 클라우드의 서비스를 제공받게 된다. 제안하는 논문은 멀티클라우드에서 사용자에게 편의성과 상호인증을 전자서명 기반으로 제공하여 위장공격과 부인공격을 방지하고 있으나 기밀성에서 취약점을 가지고 계산 효율이 떨어진다.

Rohitash Kumar Banyal et al.[11]은 클라우드 컴퓨팅 환경에서 멀티팩터 인증 프레임워크를 제안하였다. 그들은 인증단계를 3단계로 나누어 각각에 Secret Key와 Captcha, OTP, IMEI를 이용해 인증을 한다. 제안하는 프레임워크는 정적인 비밀값이 아닌 동적으로 바뀌는 팩터들을 사용하여 상호 인증을 하였다. 단일 클라우드 환경에서 적용되는 인증 방법이기 때문에 다양한 클라우드 서버를 이용해야 좋은 효율을 내는 멀티플랫폼 카쉐어링 서비스에 부적합하다.

## 3. 제안

### 3.1 Proposed Car-Sharing Architecture

Fig3.은 제안하는 카쉐어링 환경에서의 아키텍쳐이다. 카쉐어링 서비스는 하나의 기업에서 단독적으로 제공하는 것이 아닌 서로 다른 SP에 등록되어 여러 회사에서 관리하는 스마트카를 사용하는 서비스이다. 지문정보는 바이오정보로 한명당 하나의 지문만이 존재한다. 따라서 지문정보는 한번 노출이 되면 지속적으로 피해를 입을 가능성이 높다. 가공되어 있지만 재생성이 불가능한 바이오정보는 각각의 회사에 저장되지 않고 각각의 서비스 클라우드를 중계해주는 브로커에만 바이오정보를 등록하여 여러 SP가 제공하는 서비스를 받고자 한다. 브로커와 각각의 SP들은 등록과정을 통해 신뢰관계를 형성하고, 사용자는 브로커를 통해 서비스에 접근한다. 또한 카쉐어링 서비스는 보안상 취약한 단순 ID/PW방식으로 인증을 하기 때문에 보안의 강도를 높이고자 ID/PW방식과 지문인식을 결합한 브로커와의 사용자 인증, 등록phase에서 공개키 기반 상호 인증한 브로커와 SP의 클라우드 간의 통신을 통해 안전한 카쉐어링 서비스를 제공하는 scheme을 제안하고자 한다.

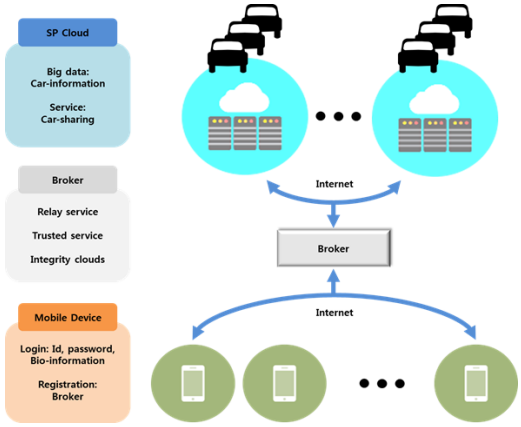


Fig. 3. Proposed Car-Sharing Architecture

### 3.2 제안 프로토콜

제안하는 카쉐어링 환경에서 브로커를 이용한 상호인증 기법은Provisioning 단계와 Authentication 단계, 티켓 양도 phase로 나누어져 있다. Provisioning 단계에서는 각 Service provider와 브로커, 유저와 브로커가 상호 등록한다. Authentication 단계에서는 사용자의 모바일 디바이스와 브로커간과, 브로커와 Service provider Cloud간, 사용자 모바일 디바이스와 Smartcar간의 상호인증을 수행하고 티켓을 발부하여 인가된 사용자만이 Smartcar를 이용 할 수 있게 한다. 티켓 양도 phase에서는 서로 합의 된 유저간의 티켓 양도 프로토콜을 이용해 인가된 사용자를 바꾸는 작업을 진행한다. Broker, SP Cloud와 CA, Smartcar와 SP Cloud, M1과 M2는 신뢰관계라 가정한다. 제안하는 프로토콜의 파라미터 설명은 Table1과 같다.

Table 1. Proposed Protocol Parameters

| Notation | Meaning |
|---|---|
| MobileDevice | User's MobileDevice for collect Bio-information |
| SP cloud | Service Provider's cloud server |
| CA | Certificate Authority |
| A.ID/PW | A's ID, Password |
| b | Bio-information |
| PW | Password |

| | |
|---|---|
| N | Secret value nounce |
| ET | Expire Time |
| $Ticket_A$ | Ticket encrypted with Key A |
| $A_{pub}$ | A's publickey |
| $R_A$ | A's Random Number |
| verify() | Value comparison function with the value in parentheses |
| h() | hash function |
| E() | Encryption |
| D() | Decryption |
| Auth() | Function to certify values in parentheses |

### 3.2.1 Provisioning Phase

Fig4.는 ServiceProvider가 Broker에 등록하는 과정과, 사용자가 MobileDevice를 이용해 Broker에 등록하고 세션키를 만들 바이오정보로 만들어진 비밀값을 나눠가지는 과정이다. 1번부터 7번까지는 SP가 브로커에 등록하는 과정과 서로 필요한 정보를 전송하는 과정이다. 8번부터 14번은 사용자가 Broker에 등록하고 바이오정보를 이용한 비밀값을 이용해 세션키를 만들어 인증하는 과정이다. CA는 사용자가 사전에 지문정보를 등록하고 SPcloud의 PublicKey를 인증해주는 기관이다.

Step1. SP cloud 서버는 SP의 ID와 SP가 생성한 랜덤값 $R_{sp}$을 Broker에게 보냄

Step2. Broker는 신뢰하는 인증기관 CA에 SP의 인증서를 요청한다.

Step3. CA는 Broker로부터 요청받은 SP의 인증서를 보낸다.

Step4. 브로커는 인증서를 확인하고 SP의 PK를 이용해 자신의 인증서, 자신이 생성한 랜덤값 $R_B$, SP의 ID와 랜덤값 $R_{sp}$, 자신이 생성한 랜덤값 $R_B$를 서명하여 보내고 $R_B$를 hash해 세션키를 만든다..

Step5. 브로커로부터 받은 인증서를 CA에게 보내 브로커의 인증서인지 확인한다.

Step6. 브로커로부터 받은 인증서가 브로커의 인증서가 맞다면 인증서와 메세지를 보내주고 아니라면 폐기한다.

Step7. 인증서를 확인 받은 후 브로커가 보낸 메시지를 받고 메시지를 SP의 개인키로 복호화하여 얻은 $R_B$을이용해세션키를생성하고인증서와브로커의 ID와 랜덤값 $R_B$, 자신이 생성한 랜덤값 $R_{SP}$ 에 서명한 뒤 세션

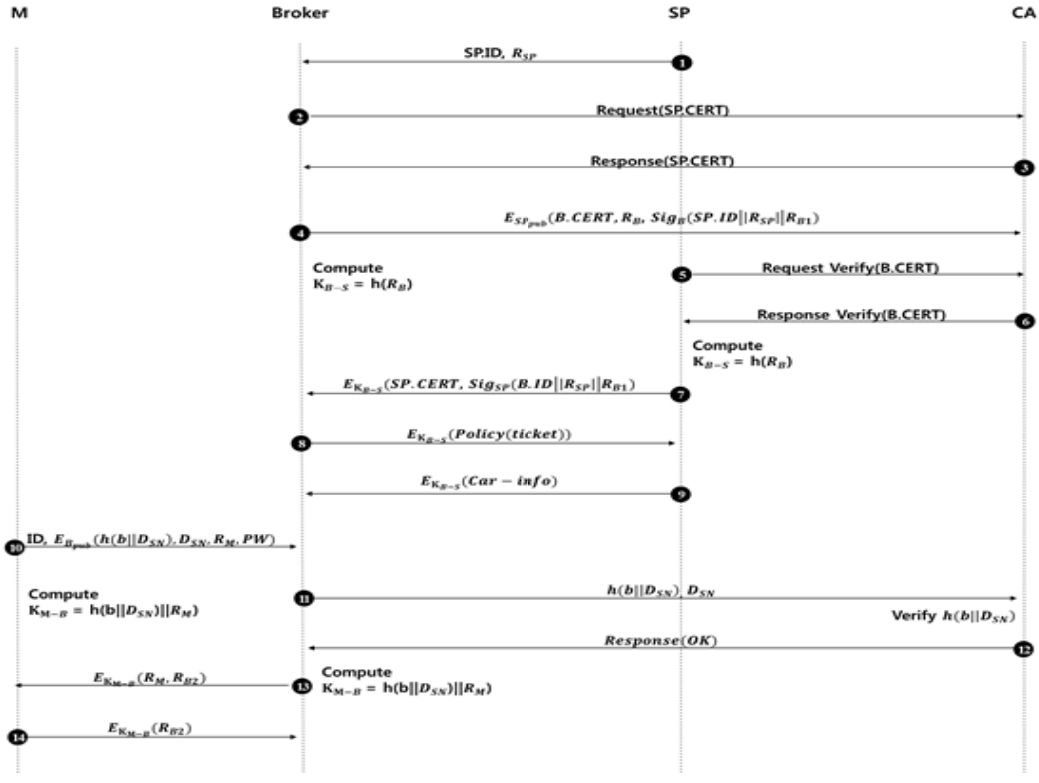


Fig. 4. Proposed authentication protocol

키로 암호화하여 보낸다.

Step8. 인증서와 서명으로 서로를 인증한 후 브로커는 이후 서비스에서 제공될 Ticket에 관한Policy를 보낸다.

Step9. SP도 이후 서비스제공에 필요한 정보인 쉐어링카에 대한 정보를 세션키로 암호화하여 브로커에게 보낸다.

Step10. 유저는 가입요청을 하고 ID는 평문으로 보내고 브로커의 공개키를 이용해 비밀값인 바이오정보와 디바이스 시리얼넘버를 연접한 뒤 hash한 값과 디바이스 시리얼넘버, 유저가 생성한 랜덤값 $R_M$, 패스워드를 암호화해서 보낸다.

Step11. 브로커는 유저로부터 받은 과비밀값 h(b||$D_{SN}$) 값을 유저가 사전에 바이오정보를 등록한 신뢰기관인 CA에 보내 검증을 요청한다.

Step12. CA에서는 기존의 지문정보와 브로커로부터 받은 값을이용해 h(b||$D_{SN}$)을 계산하고 브로커로부터 받은 h(b||$D_{SN}$)을 검증 후 확인 메시지를 보낸다.

Step13. 확인 메시지를 받은 브로커는 유저의 정보가 맞다면 세션키 K = h(h(b||$D_{SN}$)||$R_M$)를 계산하고 유저로부터 받은 랜덤값 $R_M$과 자신이 생성한 랜덤값 $R_{B1}$을 K로 암호화해서 보낸다.

Step14. 사용자측에서 세션키 K를 같은 방법으로 계산하여 만들고 보내온 메시지 $R_{B1}$를 확인한 후 다시 K를 이용해 $R_{B1}$를 암호화해서 보냄으로 등록 phase를 종료한다.

### 3.2.2 인증 Phase

Fig5.는 사용자가 모바일 디바이스를 이용해 SP에서 제공하는 SmartCar를 사용할 수 있게 하는 티켓을 얻는 방법이다. 1번부터 7번까지는 사용자를 인증하는 과정이다. 8번부터 12번까지는 사용자에게 티켓을 발부하고 티켓을 이용해 스마트카의 제어권을 넘겨받는 과정이다.

Step1. 사용자는 Smartcar의 제어권을 받기 위해 Smartcar에 ID를 포함한 사용요청 메세지를 보낸다.

Step2. Smartcar측에서는 SP cloud에 브로커의 ID를 가지고 접근하는 사용자에게 제어권을 넘겨줘도 되는지 SP cloud에게 확인 메시지를 보낸다.

Step3. SP cloud는 브로커에게 ID를 확인하고 사용자 인증을 요청한다.

Step4. 브로커측은 Provisioning 단계에서 등록한 바이오정보를 이용한 세션키를 통해 사용자가 맞는지 확인을 요청한다.

Step5. 사용자는 기존 ID/PW인증에 필요한 세션키 생성을 위해 랜덤값을 보내게 되는데 ID는 평문으로 전송하고 $B_{pub}$로 PW와 세션키를 생성할 때 필요한 랜덤값 $R_M$을암호화하여보낸다.

Step6. 브로커측은 자신의 개인키로 메시지를 복호화하고 획득한 $R_M$과 사전에 provisioning 단계에서 등록한 바이오정보를 이용해 세션키 K를 계산한 후 $R_M$과 자신이 생성한 $R_B$를 전송한다.

Step7. 사용자는 나눠가진 바이오정보값과 보낸 $R_M$ 값을 이용해 세션키를 생성하고 $R_B$를보낸다.

Step8. 브로커는 ID에 대한 정당한 사용자가 맞고 인증 완료를 알린다.

Step9. SPcloud는 브로커에게 사용자에게 줄 티켓을 Car.ID, m.ID, N, ET, TS를 차량과의 비밀키를 이용해 암호화한 후 전송한다 N은 차량이 SPcloud가 만든 티켓임을 확인할 값, ET는 티켓의 수명, TS는 타임스탬프로 재사용 공격에 대비한 값이다.

Step10. 브로커는 SPcloud로부터 받은 $Ticket_{CK}$을 사용자에게 전달한다.

Step11. SmartCar는 사용 요청한 사용자에게 $Ticket_{CK}$을 요청한다.

Step12. 사용자는 $Ticket_{CK}$을 보여주고 SmartCar의 제어권을 넘겨 받는다.

### 3.2.3 티켓 양도 Phase

Fig6.은 사용자1이 동승자나 혹은 다른 지인에게 티켓을 양도하는 과정이다. 사용자1이 티켓 양도를 요청하고 이후 사용자2의 인증 그리고 티켓을 양도하는 순서로 진행된다. 사용자1의 모바일 디바이스와 사용자2의 모바일 디바이스는 서로 신뢰관계에 있다고 생각한다.

Step1. 사용자1은 브로커에게 사용자2에게 티켓을 양도 하겠다고 요청 메세지를 전송한다.

Step2. 브로커는 SPcloud에 ID1과 ID2가 티켓 양도를 요청했다고 알리는 메시지를 보낸다.

Step3. SPcloud는 브로커에게 ID2의 인증을 요청하는 메시지를 보낸다.

Step4. 브로커측은 Provisioning 단계에서 등록한 바이오정보를 이용한 세션키를 통해 사용자가 맞는지 확인을 요청한다.

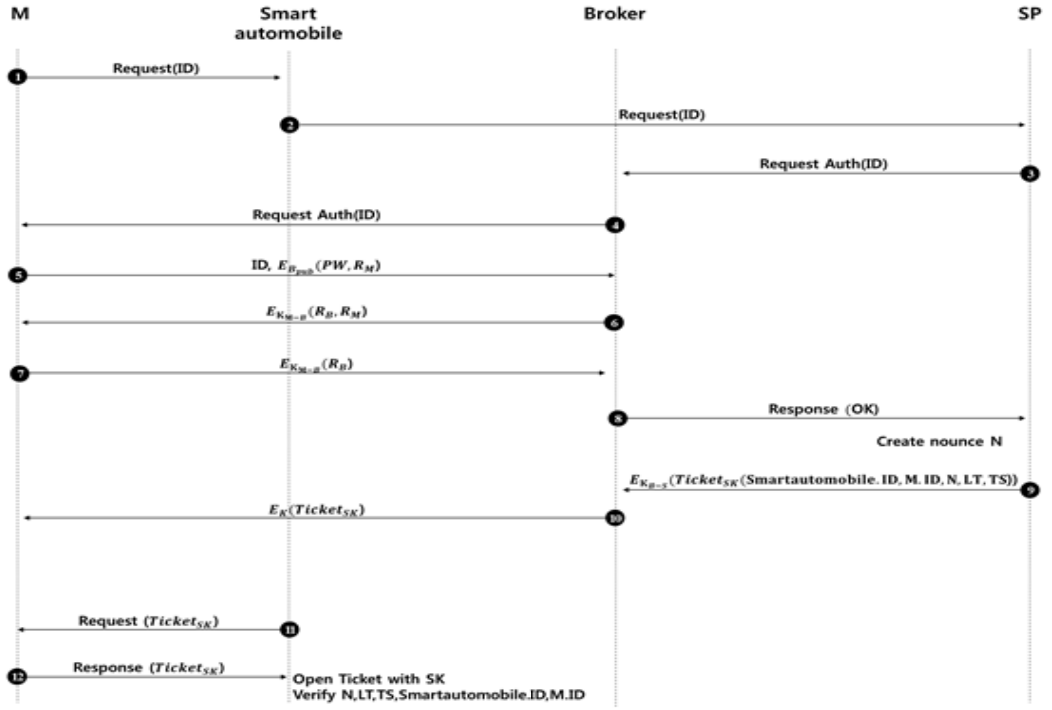


Fig. 5. Proposed authentication protocol

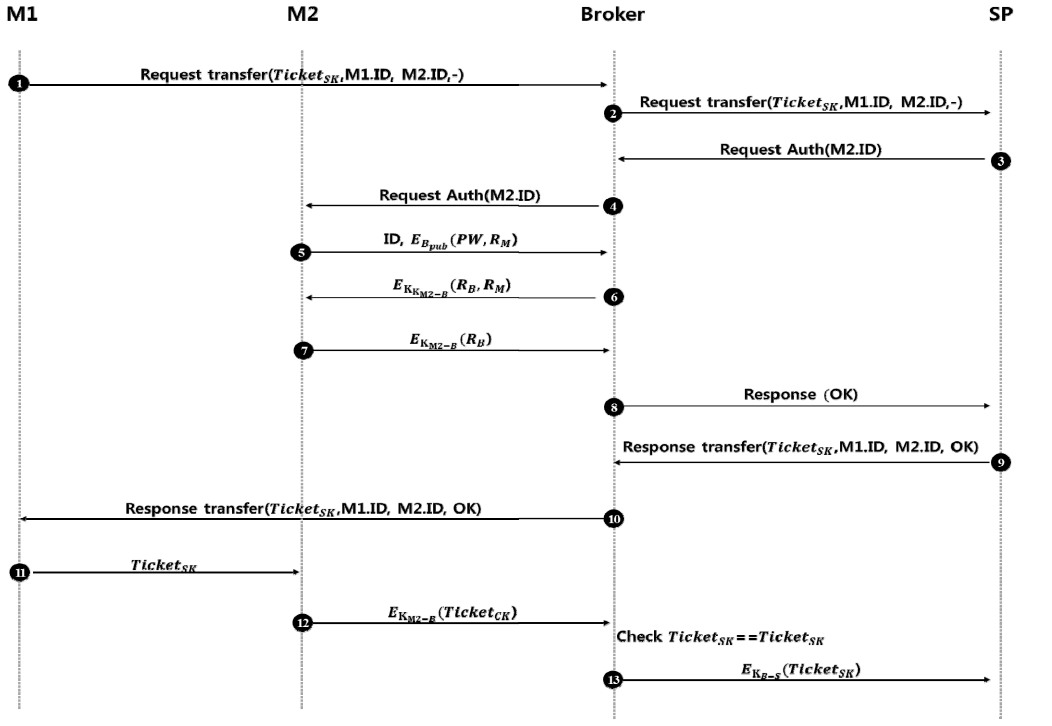


Fig. 6. Proposed ticket transfer protocol

Step5. 사용자2는 기존 ID/PW인증에 필요한 메세지와 세션키 생성을 위한 랜덤값 $R_M$ 을 보낸다. ID는 평문으로 전송하고 $B_{pub}$로 PW와 세션키를 생성 시 필요한 랜덤값 $R_M$을 암호화 하여 보낸다.

Step6. 브로커측은 자신의 개인키로 메시지를 복호화하고 획득한 $R_M$과 사전에 provisioning 단계에서 등록한 바이오정보를 이용해 대칭키 K를 계산한 후 $R_M$과 자신이 생성한 $R_B$를전송한다.

Step7. 사용자는 나눠가진 바이오정보값과 보낸 $R_M$ 값을 이용해 세션키를 생성하고 $R_B$를보낸다.

Step8. 브로커는 SPcloud에게 ID2가 정당한 사용자이며 인증 완료를 알린다.

Step9. SPcloud는 사용자에게 티켓 양도를 허락하는 메시지를 브로커에게 보낸다.

Step10. 브로커는 SPcloud로부터 받은 티켓 양도 허락 메시지를 사용자1에게 보낸다.

Step11. 사용자1은 사용자2에게 티켓을 보내준다.

Step12. 사용자2는 티켓 양도가 완료되었다는 메시지를 브로커에게 전송한다.

Step13. 브로커는 티켓 양도가 완료되었다는 메시지를 SPcloud에게 전송한다.

## 4. 보안 및 성능 분석

### 4.1 보안 분석

제안하는 프로토콜은 유저와 브로커간, 브로커와 ServiceProvider간 각각에 상호인증을 제공하고 보안체널을 설립한다. 그리고 제안하는 프로토콜은 Replay attack, forward security, phishing, impersonation attack, relay attack에 안전하며 바이오정보의 노출을 최소화 하였다. 유저와 브로커가 안전하게 인증하고 세션키를 공유할 수 있도록 CA가 유저의 바이오정보를 인증하는 역할을 한다. Table2는 2장에서 살펴본 보안요구사항에 대한 보안평가이다.

Table 2. Security Comparison Analysis

| | He Yuan Huang et al. | JaeKyung et al. | Rohitash Kumar et al. | Proposed scheme |
|---|---|---|---|---|
| Multiplatform | O | X | X | O |
| Forward Security | X | X | O | O |
| Phishing | X | X | X | O |
| Impersonation Attack | X | O | △ | O |
| Replay Attack | X | O | O | O |

#### 4.1.1 멀티-플랫폼

제안하는 브로커 기반 카쉐어링 서비스 인프라는 다양한 서비스 프로바이더 클라우드를 상호인증을 통한 신뢰하는 브로커를 기반으로 두어 하나의 브로커를 이용해 다수의 서비스 프로바이더를 유기적으로 이용할 수 있도록 하였다. 브로커는 서비스 프로바이더에게 policy를 제공하여 서비스 프로바이더들에게 데이터의 표준을 제시하고 사용자는 브로커만을 이용해 서비스 프로바이더에게 접근하기 때문에 이종데이터 형식의 문제를 해결하였다. 또한 사용자의 정보는 가공된 형태로 브로커에게만 저장되기 때문에 정보의 노출을 줄이는 효과를 제공하였다.

#### 4.1.2 전방향 보안

악의적인 공격자는 사용자와 브로커간의 현재 세션키를 탈취하여 과거의 메시지를 복구하려고 시도 할 수 있다. 만약 사용자와 브로커가 주기를 가지고 세션키를 재사용하거나 또는 세션키를 생성하는데 과거와 현재에 영향을 주는 parameters를 사용한다면 과거의 세션키를 복구 할 수 있다. 그러나 제안하는 프로토콜에서는 세션키를 생성하는데 이용하는 가공된 바이오정보 값은 과거와 현재에 영향을 주지 않고 다른 parameter는 세션키를 생성할 때 마다 사용자측에서 랜덤하게 생성되는 을 사용하고 이전에 사용된 세션키는 제거되므로Forward security에 안전하다.

#### 4.1.3 피싱

악의적인 공격자는 사용자와 브로커간의 인증에서 사용되는 ID/PW기법에서 중요한 PW를 phishing을 통하여 탈취할 수 있다. 탈취한 ID와 PW를 이용해 공격자가 생성한 을보내인증을시도할수있다. 하지만 제안하는 논문에서는 그 이후 통신을 세션키를 이용한 암호화를 통해 하게 된다. 브로커측에서는 공격자가 보낸 과등록 phase에서 사전에 나눠가진 가공된 바이오 정보를 이용해 세션키를 생성하고 세션키를 이용해 암호화를 하게 된다. 하지만 공격자는 가공된 바이오정보를 가지고 있지 않기 때문에 세션키를 생성할 수 없기 때문에 PW가 phishing을 통해 탈취되더라도 인증이 불가능하다.

#### 4.1.4 가장 공격

악의적인 공격자는 정당한 디바이스로 위장하여 상대방을 속여 비밀을 알아내거나 세션키를 설립하는 공격이다. 제안한 프로토콜에서는 디바이스에서 브로커에게 ID, PW, $R_M$을 보내고 세션키를 계산하게 된다. 공격자는 세션키에 들어가는 $R_M$을 자신이 생성해서 보낼 수 있다. 하지만 본 논문에서 제안한 프로토콜에 핵심적인 부분은 사전에 나눠가진 가공된 바이오정보에 있다. 인증단계에서 사용자는 바이오정보를 가공해서 브로커에게 보내면 브로커는 CA를 통해 그 값을 확인 받고 저장하게 된다. 브로커측에서는 사전에 저장된 바이오정보값과 $R_M$을 이용해 세션키를 만들고 사용자측은 인증요청이 들어올 때 바로 바이오정보를 수집,계산하여 만들기 때문에 공격자는 인증이 불가하고, 세션키를 만들지 못하기 때문에 가장공격으로부터 저항성을 가진다.

#### 4.1.5 재사용 공격

악의적인 공격자는 통신과정 중의 메시지를 재사용하여 인증에 이용할 수 있다. 그러나 노출되어 있는 값은 ID뿐이고 매번 통신할 때마다 바뀌는 Challenge Response값인 Rm을 사용한다. 또한 Rm은 세션키를 만드는 비밀값이고 다른 비밀값 h(b||Dsn) 값과 같이 브로커의 공개키로 암호화 되어 전송되기 때문에 이후 통신과정에서 세션키 K를 만들어 내지 못한다. 또한 세션키 K는 재사용되지 않고 만들지도 못해 이후 인증과정에서 어떠한 정보도 알아낼 수 없고 인증도 못하게 된다.

### 4.2 컴퓨팅 자원 분석

Table3은 제안하는 프로토콜에서 provision phase와 authentication phase, 티켓 양도 phase에서 mobile deivce가 X개 Broker M개, SPcloud가 Y개, SmartCar Z대 일 때 연산되는 컴퓨팅 자원을 보여준다. 사용자 중심이기 때문에 사용자는 1번의 프로토콜에서 1대의 차량만 빌리게 되므로 스마트카는 1로 설정하였고, SPcloud는 사용자와 관련 없이 브로커와의 등록관계이기 때문에 X번 그러나 이후 authentication phase에선 한곳의 서비스 프로바이더의 스마트카를 이용하기 때문에 SP 한 곳의 연산만을 추가하였다. 브로커를 이용한 카쉐어링 클라우드 서비스이기 때문에 브로커에 대부분의 연산이 치중 되어있다. 가장 복잡한 연산인 공개키 이용한 암,복호화는 등록단계에서 각 SPcloud와 4번에 각 모바일디바이스로부터 복호화 1번을 추가, 인증 초기 단계에서 각 모바일 디바이스와 1번이 전부이고 그 이후엔 세션키 K를 이용해 지속적인 정보를 주고받기 때문에 연산자원에 있어 부족함이 없고, 최소한의 연산으로 상호인증 및 보안채널을 구축하도록 하였다.

Table 3. Computing Resource Analysis

| | Mobile | Broker | SPcloud | SmartCar |
|---|---|---|---|---|
| Verify | 2 | 2X+Y | 1 | 1 |
| Hash() | 1 | X | - | - |
| Encryption(pub) | 1 | Y | - | - |
| Decryption(pub) | - | X | 1 | - |
| Encryption(K) | 2 | 2X+Y | 1 | - |
| Decryption(K) | 2 | 2X+Y | 1 | 1 |
| Generate Random Value | 2 | 2X+Y | 1 | - |
| Compute Session Key K | 2 | 2X+Y | 1 | - |

## 5. 결론

본 논문에서는 세계적으로 커지는 카쉐어링 시장에 비해 취약한 인증 문제에 대하여 다루었다. 카쉐어링에서의 인증은 권한이 없는 사용자에 의해 정당한 사용자가 금전적인 손해를 볼 수 있을뿐더러 악의적인 공격자가 운전 능력이 없는 사람일 때 차를 빌리게 된다면 인명피해도 불가피할 수 있기 때문에 중요한 사항이다. 제안하는 바이오정보 기반의 인증은 최소한의 자원으로 카 빅데이터가 등록되어있는 카쉐어링 SP로부터 서비스를 중개해주는 브로커와 스마트 디바이스간에 인증과 보안채널을 설립하는 scheme이다. 바이오정보는 개인마다 갖고있는 고유한 정보이고 다시 바꿀 수 없기 때문에 바이오정보의 노출을 최소화하고 사용자와 서비스 프로바이더의 편의를 위해 브로커를 설치 하였다. 다양한 보안 위협에 대하여 보안 분석을 통해 안전하다는 것을 평가하였으며, 각각 통신 주체에 따라 필요한 컴퓨팅 자원을 분석함으로써 제안하는 scheme을 평가하였다. 본 논문에서 제안하는 scheme을 통해 앞으로 카쉐어링 환경에서 안전한 인증과 통신, 양질의 서비스를 제공받을 수 있기를 기대한다.

## References

[1] Boyacı, Burak, Konstantinos G. Zografos, and Nikolas

Geroliminis. An optimization framework for the development of efficient one-way car-sharing systems. European Journal of Operational Research 240.3 (2015) 718-733.
DOI: https://doi.org/10.1016/j.ejor.2014.07.020

[2] Shaheen, Susan A., and Nelson D. Chan. "Evolution of e-mobility in carsharing business models." Electric Vehicle Business Models. Springer, Cham, 2015. 169-178.
DOI: https://doi.org/10.1007/978-3-319-12244-1_10

[3] Kim, Seoyeon, et al. "A Resource Allocation Strategy for Cloud Computing in Vehicular Datacenter" The Journal of The Institute of Internet, Broadcasting and Communication 18.4 (2018): 183-189.
DOI: https://doi.org/10.7236/JIIBC.2018.18.4.183

[4] Son, Byung-Chang and Ryu, Taebeum, "Evaluation of Perceived Exertion and Satisfaction in Opening and Closing Tailgates of Sport Utility Vehicles", Journal of the Society of Korea Industrial and Systems Engineering 40.1 (2017): 1-10.
DOI: https://doi.org/10.11627/jkise.2017.40.1.001

[5] Lamport, Leslie. "Password authentication with insecure communication. "Communications of the ACM 24.11 (1981): 770-772.
DOI: https://doi.org/10.1145/358790.358797

[6] Shin, Kwang-Cheul. "A robust biometric-based user authentication protocol in wireless sensor network environment." The Journal of Society for e-Business Studies 18.3 (2013): 107-123.
DOI: https://doi.org/10.7838/jsebs.2013.18.3.107

[7] Lane, C., et al. "Car Sharing, A vehicle for sustainable mobility in emerging markets." World Resources institute Centre for Sustainable Cities (2015).
DOI:
https://www.itdp.org/2015/12/14/carsharing-a-vehicle-for-sustainable-mobility-in-emerging-markets/

[8] Katzev, Richard. "Car-sharing: A new approach to urban transportation problems." Analyses of Social Issues and Public Policy 3.1 (2003): 65-86.
DOI: https://doi.org/10.1111/j.1530-2415.2003.00015.x

[9] Lee, Joo-Kwan, et al. "Design of V2I Based Vehicle Identification number In a VANET Environment." Journal of the Korea Academia-Industrial cooperation Society 15.12 (2014): 7292-7301.
DOI: https://doi.org/10.5762/KAIS.2014.15.12.7292

[10] Hussain, Rasheed, and Heekuck Oh. "Cooperation-Aware VANET Clouds: Providing Secure Cloud Services to Vehicular Ad Hoc Networks." JIPS 10.1 (2014): 103-118.
DOI: https://doi.org/10.3745/JIPS.2014.10.1.103

[11] Huang, He Yuan, et al. "Identity federation broker for service cloud." 2010 International Conference on Service Sciences. IEEE, 2010.
DOI: https://doi.org/10.1109/ICSS.2010.46

[12] Jaekyung Lee, Junggab Son, Hunmin Kim, Heekuck Oh. "An Authentication Scheme for Providing to User Service Transparency in Multicloud Environment." Journal of the Korea Institute of Information Security & Cryptology, 23.6 (2013.12): 1131-1141
DOI: https://doi.org/10.13089/JKIISC.2013.23.6.1131

[13] Banyal, Rohitash Kumar, Pragya Jain, and Vijendra Kumar Jain. "Multi-factor authentication framework for cloud computing." 2013 Fifth International Conference on Computational Intelligence, Modelling and Simulation. IEEE, 2013.
DOI: https://doi.org/10.1109/CIMSim.2013.25

**이 광 형(Kwang-Hyoung Lee)** [종신회원]

- 1998년 2월 : 광주대학교 컴퓨터공학과 졸업(공학사)
- 2002년 2월 : 숭실대학교 컴퓨터공학과 (공학석사)
- 2005년 2월 : 숭실대학교 컴퓨터공학과 (공학박사)
- 2005년 3월 ~ 현재 : 서일대학 소프트웨어공학과 부교수

〈관심분야〉
멀티미디어 데이터 검색, 영상처리, 멀티미디어 보안, 학습콘텐츠, AI

**박 상 현(Sang Hyeon Park)** [정회원]

- 2015년 7월 : 평생교육원 (컴퓨터공학 학사)
- 2017년 8월 : 숭실대학교 일반대학원 컴퓨터공학부 (공학석사)
- 2018년 3월 ~ 현재 : 숭실대학교 일반대학원 컴퓨터공학부 (박사과정)

〈관심분야〉
정보보안, 인증