Cyber-Physical-Social 시스템과 OAuth를 이용한 IoT 인증 기법

조정우*, 이국영, 이기영

* 국립인천대학교

An Authentication Scheme Using OAuth and Cyber Physical Social System

Jeong-woo Cho*, Kuk-young Lee, Ki Young Lee

*Incheon National University

E-mail : jungwooseven@naver.com

요 약

최근 IoT 환경에서 많은 디바이스들의 등장과 더불어 특정 사용자만 접근 가능한 보호된 네트워크가 필요하게 되었다. IoT 네트워크 환경에 OAuth 프로토콜을 적용하여 인증하게 되면 보다 쉽게 네트워크 인증 체계를 구축할 수 있으나 OAuth는 공격자가 Token을 가로채게 되면 쉽게 공격에 노출되는 단점이 있어 2차 인증이 필수적이라고 할 수 있다. 궁극적인 IoT는 Fog Computing이 필수적이다. Fog Computing은 Cloud를 확장시켜 network Core에서 뿐만 아니라 Edge에서도 Computing이 가능하게 하고, node간의 통신을 가능하게 하는 플랫폼이다. Fog Computing의 장점으로는 Location awareness나 Support for mobility 등을 들 수 있다. Fog Computing내에서 사용할 인증이 이런 Fog Computing의 장점을 살린다면, 더 IoT에 걸맞은 인증을 할 수 있을 것이다. 이에 2차 인증에 기존의 인증서나 id/password, 또는 group key같은 번거로운 것을 이용하지 않고 Cyber-Physical-Social System을 사용한다면 user의 편리성을 더 증가시킬 수 있을 것이다. 본 연구에서는 Cyber-Physical-Social System기반의 인증에 관한 연구를 진행하려 한다.

ABSTRACT

Recently on IoT environment, there is necessary of protected network, which is only specific user can access it. Applying OAuth protocol on IoT, it can be easier to construct network authentication system, but it is hard to construct protected network authentication system. And there is weakness of OAuth protocol, which is easily attacked by sniffing Token by attacker. So, it is necessary to secondary authentication for OAuth. In ultimate IoT, the fog computing is essential. Fog computing is extension of cloud that enables networking not only in core system but also in edge system and communication node to node. Strength of fog computing is location awareness, support for mobility, and so on. If authentication in fog computing uses this strength, it can be more specialized in Fog Computing. So, in secondary Authentication, using Cyber-Physical-Social System will increase convenience of user than using existing authentication system, such as authentication certificate, id/password and group key, which is inconvenient for user. This study is about authentication based Cyber-Physical-Social System.

키워드

IoT, Authentication, OAuth, Fog Computing, Cyber-Physical-Social System

Ⅰ. 서 론

최근 IoT망의 증가에 따라 Fog Computing이라는 용어가 등장하기 시작했다. Fog Computing은 Cloud를 확장시켜 network Core에서 뿐만 아니라 Edge에서도 Computing이 가능하게 하고, node간의 통신을 가능하게 하는 플랫폼이다. 궁극적인 IoT는 이 Fog Computing을 이용하는 방안으로 흘러갈 것이다. Fog Computing의 장점으로는 Location awareness나 Support for mobility 등을 들 수 있다.

또한 IoT 환경에서 많은 디바이스들의 등장과 더불어 특정 사용자만 접근 가능한 보호된 네트워크가 필요하게 되었다. 이를 위해 IoT내에서 인증은 인증의 초점을 서비스에 맞추어야 한다. 즉, 인증하는 것으로 끝나는 것이 아니라 서비스에 따라 사용할 수 있는 사용자가 따로 있고, 그 서비스에 접근할 때에 자신이 적합한 사용자임을 인증하게 되는 것 이다.

그러나 IoT의 특성상 무거운 인증은 사용하기 어렵기 때문에 IoT상에서의 보안은 가벼운 것을 선호하는 경향이 크다. 이에 상대적으로 Cost가 적고 Overhead가 크지 않은 OAuth는 이러한 인증에 알맞은 인증방법중 하나라고 할 수 있다. 그러나, OAuth의 특성상 Access Token을 갈취 당하게 되면 너무 쉽게 공격당하게 되므로 2차적인 인증을 하는 연구가 많다.[1][2][3]

이에 본 논문은 Fog Computing의 Location awareness라는 특징을 이용해 CPSS(Cyber Physi-cal Social System)을 이용한 Location기반의 2차 인증을 진행하고자 한다.

Ⅱ. 기존 인증방법의 문제점

서비스기반의 인증에 Group key를 사용하는 인증방법도 있다[4]. 그러나 Group key의 특성상 user가 어떠한 Group에 참여했다가 다시 나갔을 때 약방향성의 안정성을 위해 Group key를 지속적으로 바꿔 주어야 하는 불편함이 있었다. 또한 Group key를 사용할 경우, Device나 Service의 종류가 많아지면 수많은 키를 관리하고, 수정해야 하는 단점이 있다. 서버입장에서 Group key를 도난당하게 되면 아무것도 하지 못하므로 Group key를 관리하는데 많은 시간과 비용을 들이게 된다.

OAuth를 이용한다면 이러한 불편함은 생기지 않는다. OAuth를 이용하면 상대적으로 보안 시스템을 설계하기가 편하고, ID/Password를 Service Provider측에서 관리하므로 해당 관리에 들어가는 비용역시 없어져 상대적으로 비용을 낮출 수 있다. 그러나 이러한 OAuth의 단점은 꽤나 명확하다. OAuth는 Service Provider와의 인증을 통해 Access Token을 얻게 되는데, 공격자가 이것을 갈취할 경우 공격에 쉽게 노출된다.

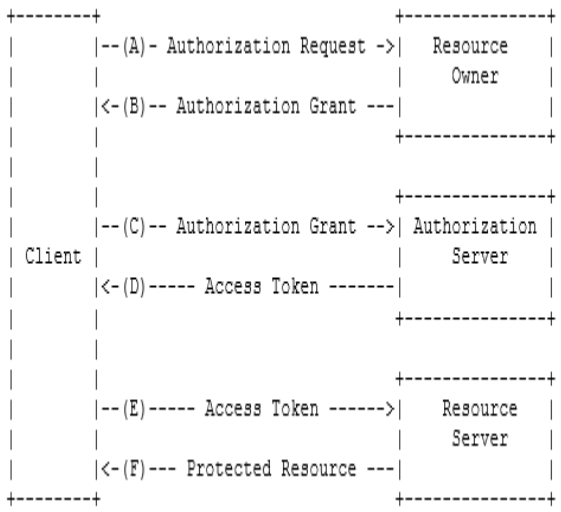

그림 1. Flow of OAuth2.0 Protocol[5]

그림1은 OAuth의 기본적인 Flow이다. Client가 Resource Owner에게 Authorization Request를 하면, Resource Owner는 Authorization Server로의 Redirection을 시켜준다. Client 는 Authoriza-tion Server와 Authentication을 하게 되고, 그를 통해 Access Token을 얻게 된다. 얻은 Access Token으로 Resource Server에게 Protected Resou-rce를 요청하면, Resource Server는 해당토큰이 정당한지를 확인하고 User에게 response를 준다.

이 Flow를 바로 IoT에 적용하기에는 크게 2가지 단점이 있다. 첫째, 앞에서도 제기했듯, Token을 갈취당하면 공격당하기 너무 쉽다. 둘째, 정당한 Token만 있으면 되기 때문에 해당 사용자가 해당 서비스에 적합한 사용자인지를 판단할 수 없다. OAuth를 사용하기 위해서는 이 두 가지 단점을 극복할 필요성이 있다.

Ⅲ. 제안하는 방식

표 1. 제안하는 방식의 Database구조

Service	Device	Valid user	Acess Control
A	ABC	User A	F
B	DEF	User B	P

표1과 같이 제안하는 방식에서 Service는 Valid user와 Access Control의 두 가지를 가진다. Valid user는 Device에 접근할 수 있는 user의 OAuth ID가 들어간다. Access Control에는 해당 Service에 대한 접근 권한을 Valid user의 OAuth ID(SNS계정)에 등록된 친구에게도 줄 경우에는 F, 그렇지 않고 Valid한 user만이 사용할 수 있게 하게끔 하려면 P를 넣는다. 이는 그림3의 Flow를 통해 Device의 Valid user의 ID로 로그인했을 경우 바꿀 수 있다.

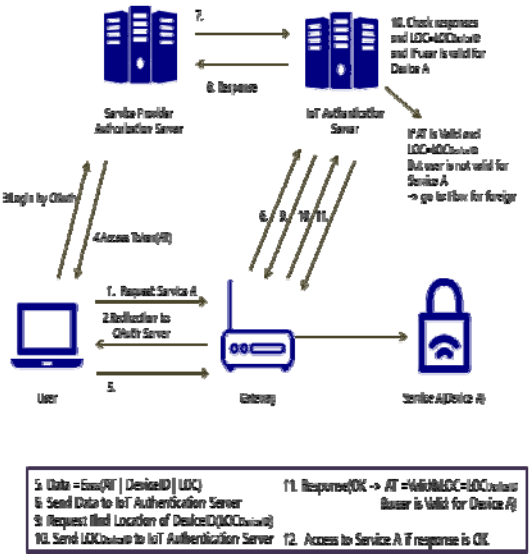

그림 2. 제안하는 Flow-사용자

그림2는 본 논문에서 제안하는 사용자의 Flow이다. 사용자가 Service에 접근하게 되면 이를 Service Provider의 Server로 Redirection을 시켜주고, 사용자는 인증과정을 거쳐 Token을 얻는다. 사용자는 이 Token과 Data를 같이 IoT Authenti-cation Server에 전송하고, IoT Authentication Server는 해당 Token과 Data가 Valid한지, 해당 사용자가 해당 Service의 Device에 Valid한지를 확인하고, 모두 Valid함이 확인 되면, Service에 접근할 수 있게 한다.

Data는 Location과 Access Token, 그리고 DeviceID 의 3가지이다. Device ID는 현재 서비스에 접근한 Device의 ID이다. 이 세 가지 Data는 Session Key를 이용, DES로 암호화되어 전송한다. DES가 사용되지 않는 이유는 Key가 짧아 Brute-Force Attack에 쉽게 뚫려 버린다는 것인데, Access Token은 유효기간이 30분 정도로 짧아 공격하는데 의미가 없다.

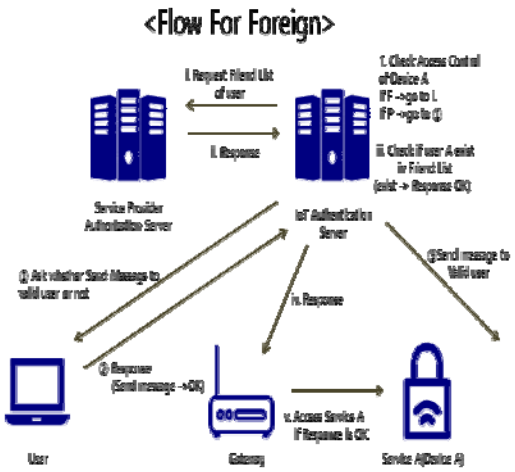

그림 3. 제안하는 Flow-제3자

그림3의 Flow는 제3자를 위한 Flow이다. 그림2의 flow에서 사용자는 미리 IoT Authentication Server에 등록되어 있어야 한다. 그러나 이러한 방식은 가스점검을 나왔다거나 친구에게 문을 열어주는 등 제3자가 잠깐 이용하고 빠져야하는 경우에는 적합하지 않다. 이에 제3자의 Flow는 필수적이다. Access Token과 Location은 Valid하지만, 해당 Service에 접근권한이 없을 경우, 해당 Service의 Device의 Access Control를 확인한다. F일 경우, 해당user의 친구목록을 Service Provider를 통해 받아 Valid user가 친구목록에 있는 지를 확인하고 있으면, 접근할 수 있게 한다. Access Control이 P이거나, 친구목록에 Valid user가 없으면, Valid user에게 메시지를 보낼 것인지를 확인한다. Valid user는 메시지를 받고 Flow3를 통해 로그인하여 Service에 접근해 준다.

Ⅳ. 보안성검토

보안적 측면에서 기존의 OAuth를 IoT 네트워크에 적용할 경우 Service Provider의 인증정보만 가지고 있으면 IoT 네트워크가 제공하는 서비스에 누구나 접근 가능하지만 제안하는 Flow에서는 Valid user와 해당 user의 친구(Access Control = F일 경우)만이 해당하는 Service에 접근할 수 있고, 아닌 경우에는 Valid user에게 Message를 보낼 수 있는 기능만 있을 뿐, 해당 Service에 접근할 수 없다.

또한 그림3의 Flow에서 Valid user가 아닌 접근하는 user의 친구목록을 확인하는 것은 SNS의 특성을 이용했다. SNS에서 친구를 맺으려면 상호간의 확인을 해야 하기 때문에, 친구목록에 Valid user를 추가하려면 Valid user의 승인이 필요하다. SNS의 계정의 친구인 경우 해당 승인을 거쳤기 때문에 Access Control이 F일 경우 승인과정을 생략하였다. 이는 Physical뿐만 아니라 Social적 요소도 인증의 한 요소를 차지하게 된다.

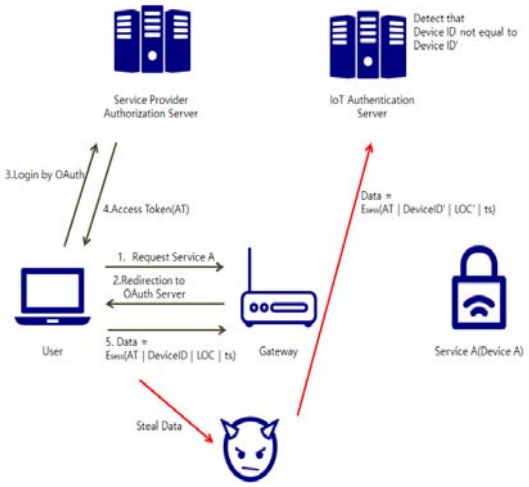

그림 4. 위장공격

OAuth에서 가장 많이 쓰이는 공격은 Access Token을 갈취하고 이를 이용해 공격하는 위장 공격이다. 그러나 본 논문의 flow라면 위장공격을 방지할 수 있다. 1차적으로 DES를 이용하여 보안을 하여 공격을 방지한다. 또한 Token이 갈취되어도 공격자의 Device ID'은 처음에 인증과정을 거쳤던 Device ID와 다르므로, 서버는 이를 인식하고 Device ID'의 위치를 확인하지 않고도 이를 공격으로 인식하여 해당 접근을 불가하게 할 것이다. 또한, 공격자가 Data의 Device ID부분을 바꾼다 해도, Access Token의 Expiration Time안에 바뀔 수 있는 위치는 한정적이므로, 공격자의 위치가 인증과정을 거쳤던 사용자의 위치와 많이 떨어져 있지 않아야 한다. 이 역시 공격을 많이 줄일 수 있을 것이다.

Ⅴ. 결 론

본 논문이 제시한 Flow는 Fog Computing기반에서 돌아간다는 특징이 있다. 이는 아직까지 Fog Computing은 상용화 단계에 접근하고 있다는 관점에서는 단점이라고 할 수 있다. 물론 Fog Computing의 Location Awareness가 아니라 GPS를 이용한다면 이 단점은 극복할 수 있다. 그러나 이것은 장점이라고도 볼 수 있다. 궁극적인 IoT는 Fog Computing을 당연히 지원할 것이고, 이 Fog Computing내에서의 인증이 Fog Computing의 장점을 활용할 수 있다면 더 IoT친화적인 인증이라고 말할 수 있을 것이다.

또한 ID/Password나 인증서는 ID나 Password를 기억해 둬야하고, Password를 Service에 접근할 때 마다 입력해야 한다는 것은 보안을 떠나 솔직한 사용자의 입장에서는 불편하다. 본 연구는 Location을 기반으로 하고 Physical과 Social이라는 기존의 방법과 비교했을 때 상대적으로 덜 번거롭지만 보안의 강도에 많은 차이가 없는 System을 이용한 인증이기 때문에 이런 번거로움을 줄일 수 있다.

IoT내에서 인증은 필수적이다. 거의 모든 것이 인터넷에 연결되어 있고, 인터넷을 통해서 Control 및 Access가 가능하기 때문이다. 그래서 만약 인증이 제대로 이뤄지지 않는다면 공격자에 의해 치명적인 피해를 얻을 수 있다. 공격자는 IoT서버 내에 접속해서 개인정보를 빼내어 온다던가, 현관문제어를 못하게 하여 집에 들어올 수 없도록 할 수도 있다. 그러므로 검증된 사용자가 IoT환경 내에 접속하는 인증은 필수적인 요소 중 하나라고 볼 수 있다. 이에 본 논문은 이러한 인증에서 보안성도 높지만 더욱 사용자에게 친화적이며 서비스 중심의 인증의 한 방향을 제시하고자 했다.

참고문헌

[1] 최영규, 김선정, 김강석 ,김기형, "OAuth기반 IoT Network 인증기법", 한국정보과학회 학술발표논문집, vol.2015, No.6, pp.1069-1071, 2015.

[2] 최영규, "사물인터넷(IoT) 네트워크 환경에서 OAuth기반 사용자 인증기법", 아주대학교 학위논문(석사), 2015.

[3] Simone Cirani, Marco Picone, Pietro Gonizzi, Luca Veltri and Gianluigi Ferrari, "IoT-OAS : An OAuth-Based Authorization Service Architecture for Secure Services in IoT Scenarios", In IEEE Sensors Journal, vol 15, no.2 , February 2015.

[4] Zhong Liu, Dong-sheng Yang, Ding Wen, Wei-ming Zhang, Wenji Mao, "Cyber-Physical-Social Systems for Command and Control", IEEE Intelligent Systems, vol.26, no. 4, pp. 92-96, 2012.

[5] D.Hardt, "The OAuth2.0 Authorization Framework", RFC6749, 2012.