IoT 환경에 적합한 경량 EAP 메소드 설계

유요셉* · 김기천*

*건국대학교 컴퓨터·정보통신공학과

A Design of Lightweight-EAP Method for IoT Environment

Joseph Yoo* · Keecheon Kim**

*Dept. of Computer Science & Engineering, Konkuk University

E-mail : josephyoo94@konkuk.ac.kr, kckim@konkuk.ac.kr

요 약

EAP는 확장 가능한 인증 프로토콜로써 다양한 인증 메커니즘을 가진 EAP 메소드를 지원하는 프로토콜이다. EAP 자체는 IP를 통한 전송 환경에 대한 설계가 아닌 인증만을 위한 프로토콜로 디자인되어 있기 때문에, peer와 authenticator 사이의 인증을 마친 이후에 일반적인 데이터 전송에는 사용되지 않는다. EAP의 간결한 통신 구조로 볼 때 EAP 자체는 가볍게 동작할 수 있는 프로토콜이나, 어떠한 EAP 메소드를 선정하여 사용하느냐에 따라 그 절차가 좀 더 복잡해질 수 있다. 특히, 최근에는 IoT 시장이 정착되며 무선환경에서의 데이터 손실이나 변조, 연결의 반복 등으로 인해 잦은 인증 환경이 나타난다. 이 때 일부 보안성이 뛰어난 EAP 메소드의 경우에는 복잡한 절차를 이용하는 만큼 가볍고 빠른 통신을 요구하는 일부 IoT 환경에서는 적합하지 않다. 이에 본 논문에서는 기존 EAP의 틀은 건드리지 않으며 잦은 인증 및 빠른 통신을 요구하는 IoT 환경에 적합한 경량 인증 EAP 메소드를 설계한다.

ABSTRACT

EAP is an extensible authentication protocol that supports EAP methods with various authentication mechanisms. Since EAP itself is designed as a protocol for authentication only, it is not used for general data transmission after authentication between peer and authenticator. EAP itself is a protocol that can operate lightly in terms of the simple communication structure of EAP, but the procedure may become more complicated depending on which EAP method is selected and used. In particular, the IoT market has recently become established, and frequent authentication environments arise due to data loss, modulation, and repeated connections in a wireless environment. In this case, some highly secure EAP methods are not suitable for some IoT environments that require lighter and faster communications than complex procedures. In this paper, we design a lightweight authentication EAP method that is suitable for IoT environment that does not touch the existing EAP framework and requires frequent authentication and fast communication.

키워드

EAP, EAP메소드, 인증, 경량화, IoT

Ⅰ. 서 론

최근 통신을 하는 기기들이 사물 전반으로 확대가 되는 사물인터넷(IoT) 산업이 급격한 성장을 보이고 있다. 글로벌 시장 전문 기관인 IHS에 따르면 IoT 기기들의 수는 2015년 약 154개에서 2025년에는 약 754억개로 늘어날 전망이다. [1]

통신을 하는 장치의 유형과 크기가 다양해짐에 따라 소형 단말이 사물인터넷의 주요 요소로써 자리 잡게 되었고, 이로 인해 낮은 단가의 저사양 통신 모듈의 사용이 증가하고 있다. 사양이 낮은 통신 모듈은 일반적인 기기들에 비해 처리 능력이 낮기 때문에 데이터의 전송 및 처리 시, 경량화된 프로세스가 요구되고 있다. 때문에 현재의

추세는 전송 단계의 경량화 및 데이터 처리의 효율성에 대한 연구가 활발하다.

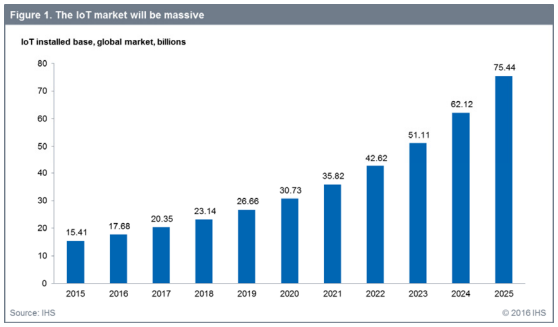

그림 1. IoT 기반 장치들의 증가 예측(IHS 2016)

상당수의 IoT 장치들은 무선 네트워크로 연결된 경우가 많으며, 이에 따른 데이터의 손실 및 변조 가능성, 재연결 시의 인증이 다시 이루어져야 하는 등 다양한 문제를 내포하고 있다.

본 논문에서는 인증 프로토콜의 데이터를 담아서 전송하는 프로토콜인 EAP 및 보안이 갖춰진 인증 메소드에 대해 알아보고 잦은 인증 상황과 저사양의 컴퓨팅 환경에 적합한 EAP 메소드를 제안한다.

II. EAP

2.1 EAP(Extensible Authentication Protocol) 개요

EAP는 인증 프로토콜의 일종으로써, 보안 프로토콜이란 네트워크의 사용자 인증 및 단말 인증과 관련되는 보안서비스를 제공하는 프로토콜이다. 특히 EAP는 복수의 인증 프로토콜을 캡슐화하여 다양한 방식의 인증을 허용하는 범용 인증 프레임워크다. EAP는 data-link 계층의 PPP환경에서의 동작을 우선으로 디자인되었으며, 이후 IEEE 802 환경에서 동작할 수 있도록 수정되었다. 동작하는 계층이 낮은 만큼 통신 시 IP가 요구되지는 않는다는 특징이 있다. 즉, 아직 네트워크에 접속이 이루어지지 않아서 IP의 할당이 이루어지지 않은 상황에서의 네트워크 접속 인증을 전제로 디자인되어 있어 일반적인 데이터 전송의 용도로 사용하기에 적합한 프로토콜은 아니다.

EAP의 인증 절차는 크게 나눠서 authenticator가 peer에게 인증 요청(request)을 전송, peer의 응답(response) 패킷 전송, EAP 메소드에 따른 추가적인 request와 response의 과정을 반복, authenticator가 Failure/Success를 전송하는 단계로 나눠진다.

EAP는 총 4개의 계층으로 이루어져 있으며, 계층의 종류에는 하위 계층(Lower Layer), EAP 계층(EAP Layer), EAP Peer/Authenticator Layer, EAP 메소드 계층(EAP method layer)이 있다. 하위 계층은 EAP 프레임을 peer와 authenticator간의 전송을 하며, PPP 및 유/무선 LAN등이 포함된다. EAP계층은 EAP 패킷을 주고 받는 역할을 하며, 수신 시 패킷의 Code 필드에 따라 EAP peer 계층이나 EAP authenticator 계층으로 전달하게 된다. EAP 메소드 계층은 실제적인 인증 알고리즘이 구현되는 계층으로, 패킷의 Type 필드에 따라 올바른 인증 메소드를 이용하여 처리하게 된다.

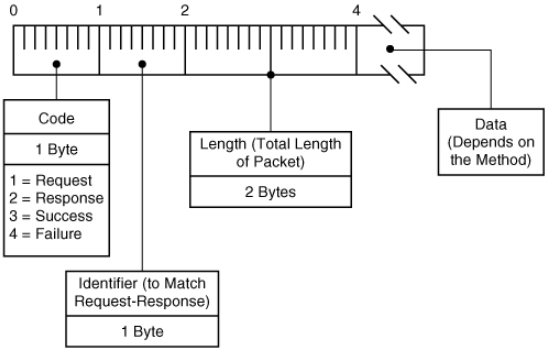

그림 2. EAP Packet 구조

EAP 패킷의 Code 필드에는 1 ~ 4의 값이 들어올 수 있으며 각각 request, response, success, failure를 나타낸다. 이 중 peer는 response만 보낼 수 있다는 특징이 있다. ID는 패킷의 식별번호이며, response 패킷의 ID는 request와 일치해야 한다. 새로운 request시에는 새로운 번호를 사용하며, response를 기다리는 동안 발생한 타임아웃으로 인한 재전송 시에는 동일한 번호를 사용한다. 이러한 경우 수신 측에서는 중복된 ID값을 가진 패킷에 대해 하나만 응답을 하고 나머지는 버리게 된다. Code 필드가 success나 failure시에는 패킷의 Data 필드가 없이 4바이트만 전송하게 되며, request와 response시에는 Data필드의 첫 1byte를 type필드로 사용하게 된다. Type 필드의 1부터 3까지의 값은 실제 EAP 메소드의 type이 아닌 EAP 전용 메시지로 사용된다. 이 EAP 전용 메시지가 아닌 나머지 type 필드 값은 특정 메소드를 사용하도록 지정한다. [2]

2.2 EAP 메소드

여기서는 EAP의 메소드들에 대해 정리한다. 다양한 EAP 메소드들이 존재하지만, RFC 표준으로 지정된 대표적인 메소드들을 제시한다.

EAP-MD5(Message Digest 5)는 EAP type 4번으로 사용자의 이름과 패스워드 기반의 단순한 인증 프로토콜이다. 사용자 별 인증이 가능하고 구현이 쉬우나, 단방향 인증을 동해 사용자만 인증이 가능하며 서버의 인증은 불가능하다. 기존의 PPP-CHAP을 유사한 방식으로 EAP에 적용하였으나 현재는 취약성이 보고되어 더 이상 권고하지 않는다. [3]

EAP-TLS는 EAP type 13번으로 취약한 무선 환경에서 신뢰성 있는 보안 채널을 형성하기 위해 사용한다. 인증서 기반 프로토콜이므로 서버 인증서와 클라이언트 인증서 모두 필요하다. 따라서 서버와 클라이언트의 상호 인증이 가능하다는 특징이 있다. [4]

EAP-TTLS(Tunneled TLS)는 EAP type 21번으로 내부 망은 TLS 터널링을 하고 외부 망만 공인 PKI를 사용하는 방식이다. 이 방식은 인증서가 설치된 서버와 개인 사용자 사이의 터널을 이용해서 암호화 된 인증 정보를 주고받는다. [5]

EAP-FAST(Flexible Authentication via Secure Tunneling)은 EAP type 43번으로 실제 내부적인 인증은 TLS 터널 내에서 이루어지고, TLS 터널화를 위한 인증은 비용이 많이 소요되는 PKI 표준 인증서 대신 인증서버에서 생성된 공유키(PAC)를 사용한다. PAC를 이용하여 단말과 서버 간의 상호인증을 하고 TLS 터널을 위한 협상을 한다. TLS 설정 이후에 실제적인 사용자 인증은 TLS 터널을 통해 수행된다. [6]

Ⅲ. 경량 암호화

3.1 HIGHT(HIGh security and lightweigHT)

HIGHT 암호 알고리즘은 RFID나 USN과 같은 저전력 및 경량화를 요구하는 컴퓨팅 환경에서 기밀성을 제공하기 위해 2005년 국내에서 개발된 64bit 블록 암호 알고리즘이다. HIGHT는 2006년에 정보통신단체표준(TTA)으로 제정, 2010년에는 ISO/IEC 국제 블록 암호 표준으로 제정되었다.

HIGHT의 전체 구조는 일반적인 Feistel 변형 구조로 이루어져 있으며, 64bit의 평문과 128bit Master key로부터 생성된 8개의 8bit 화이트닝 key와 128개의 8bit Sub Key를 입력으로 사용하여 총 32 라운드를 거쳐 64bit 암호문을 출력한다. HIGHT는 제한된 자원을 갖는 환경에서 사용할 수 있도록 8bit 단위의 기본적인 연산들인 XOR, 더하기, 순환이동만으로 설계되었다. [7]

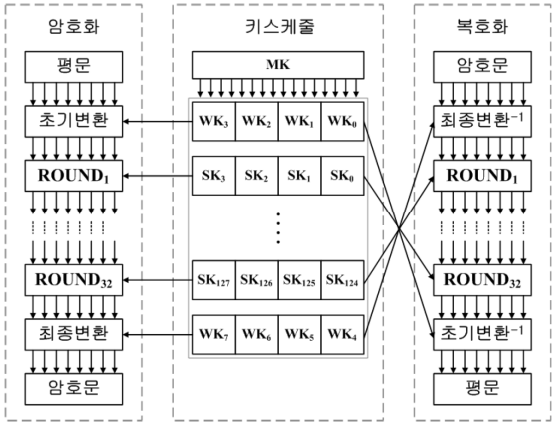

그림 3. HIGHT 전체 구조

HIGHT 알고리즘은 기본적인 AES 블록 암호 알고리즘에 비해 동일한 컴퓨팅 환경에서 약 3배 이상의 빠른 속도를 가지고 있다고 알려져 있으며, 이를 비교한 표는 다음과 같다. [8]

알고리즘	CPU/OS/Compiler	
	P3/WinXP/VC	P4/WinXP/VC
HIGHT	88.882	72.413
AES1	329.452	234.765

*P3: Pentium Ⅲ, 801 MHz
*WinXP: Microsoft Windows XP
*P4: Pentium Ⅳ, 2GHz
*VC: Microsoft Visual C++ Version 6.0

표 1. HIGHT와 AES의 S/W 구현 성능 비교

Ⅳ. HIGHT를 사용한 EAP-HIGHT 제안

본 장에서는 저전력 및 경량화가 요구되는 컴퓨팅 환경인 IoT 환경에서 적합한 EAP 메소드를 제안하고자 한다. EAP의 안전한 통신을 위해서는 인증서를 사용하는가의 여부도 중요하지만, 통신이 TLS 터널링을 통해 안전하게 보안이 이루어지는지가 더욱 중요하고 볼 수 있다. IoT 환경에서는 최악의 경우 장치간의 연결이 매번 끊겼다가 다시 접속되는 상황을 고려해야 하며, 상위 전송계층에서 전송되는 데이터는 개발자가 쉽게 경량화할 수 있으나, 매 인증 시 발생하는 오버헤드에 대해서는 인증 프로토콜 자체에서의 경량화 된 디자인이 필요하다. 기존의 TLS방식들은 초기 인증 시에는 공개키 방식으로 이루어지며, 이후 인증서가 전달되면 데이터를 Triple-DES와 같은 블록 암호 방식을 사용해 암호화하여 전달하게 된다. 이 블록 암호화 방식을 국제 표준 저전력 및 경량 암호화 방식인 64bit HIGHT를 사용하는 것으로 바꿀 시, 저전력/저사양 환경인 IoT 환경에 적용하기 적합한 알고리즘이라 볼 수 있다. 따라서 EAP 메소드에서의 TLS 터널링 형식은 유지하되, 이 터널링 이후 과정에서 사용되는 암호화 알고리즘을 ISO/IEC 국제 블록 암호 표준으로 제정된 HIGHT를 사용하는 EAP-HIGHT방식의 구현을 제안한다.

본 논문에서 제안하는 EAP-HIGHT의 구조는 [그림 4]와 같다.

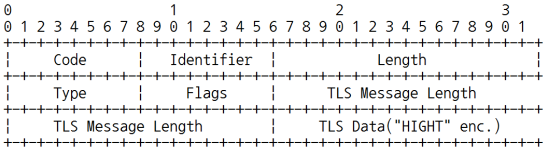

그림 4. EAP-TLS 기반의 EAP-HIGHT 구조

[그림 4]에서 나타낸 것과 같이, EAP-HIGHT는 기존의 EAP-TLS의 구조를 따르지만 TLS 터널링

이후에 암호화가 진행된 “TLS Data”의 부분에 사용되는 암호화 알고리즘에 HIGHT를 사용한다. 이 EAP-HIGHT 메소드는 EAP-TLS 메소드와는 별개의 메소드로 존재하게 되며, EAP의 “IANA Consideration”에 명시된 “EAP Type”에 대한 type 번호 할당이 필요하다. 본 논문에서는 우선 EAP의 “Method Types” 생성 규칙에 따라 테스트 용도인 255번을 사용한다.

Ⅴ. 결 론

본 논문에서는 인증 프로토콜의 데이터를 담아서 전송하는 프로토콜인 EAP에 대해 알아보고 잦은 인증 상황 및 저전력 환경에 적합한 EAP 메소드에 대하여 제안했다. IoT 시장의 성장으로 인해 경량화 된 통신 환경이 요구되고 있으며, 프로그래머가 사용하는 프로토콜들에 대해서는 다양한 경량화 및 안전한 통신에 대한 연구가 활발히 이루어지고 있다. 그러나 단말의 인증 절차에 대해서는 기존의 인터넷 환경에서 사용되는 프로토콜을 그대로 사용하는 경우가 많으며, 이는 상황에 따라 IoT환경에서 적합하지 않을 수 있다. 이러한 점들 때문에 경량 암호 및 인증에 대한 관심이 높아지고 있는 추세인 만큼, EAP에서 사용되는 인증 방식에 사용되는 데이터의 암호화 알고리즘으로 HIGHT 경량 블록 암호화를 적용하는 제안된 EAP-HIGHT를 사용함으로써 보안기능을 갖춘 경량화된 인증 방식이 이루어지리라 기대된다.

향후 연구에서는 본 논문에서 제안한 EAP-HIGHT를 별도의 EAP 메소드로써 구현을 하여 실제 임베디드 기기 환경에서의 성능 테스트를 진행할 예정이다.

ACKNOWLEDGMENT

이 논문은 2016 년도 정부(미래창조과학부)의 재원으로 정보통신 기술진흥센터의 지원(No.B0511-16-0001, 글로벌 딜리버리 클라우드 플랫폼의 대규모 OTT 서비스 적용을 위한 방송•통신 사업자 공동의 시범 사업)으로 수행된 연구임

참고문헌

[1] IHS, “IoT platforms: enabling the Internet of Things”, 2016,
(https://www.ihs.com/Info/0416/internet-of-things.html).

[2] IETF, “RFC 3748 - Extensible Authentication Protocol (EAP)”, (https://tools.ietf.org/html/rfc3748).

[3] IETF, “RFC 2284 - PPP extensible Authentication Protocol (EAP)”, (https://tools.ietf.org/html/rfc2284).

[4] IETF, “RFC 5216 - The EAP-TLS Authentication Protocol”, (https://tools.ietf.org/html/rfc5216).

[5] IETF, “RFC 5281 - Extensible Authentication Protocol Tunneled Transport Layer Security Authenticated Protocol Version 0 (EAP-TTLSv0)”, (https://tools.ietf.org/html/rfc5281).

[6] IETF, “RFC 4851 - The Flexible Authentication via Secure Tunneling Extensible Authentication Protocol Method (EAP-FAST)”, (https://tools.ietf.org/html/rfc4851).

[7] KISA, “HIGHT 알고리즘 사양 및 세부 명세”, 2009.

[8] KISA, “HIGHT - KISA”, (https://seed.kisa.or.kr/iwt/ko/sup/EgovHightInfo.do)