IoT 종단간 보안을 위한 ECQV 인증서 기반의 보안 메커니즘★

연 한 별*, 박 창 섭**

요 약

IoT 기술은 점차 발전하고 있으며 관련 서비스와 기술들이 생활 곳곳에 스며들고 있다. 이러한 IoT 기술은 사용자의 삶을 편하게 해주지만 양날의 검처럼 큰 위협 또한 가지고 있다. 때문에 보안의 중요성이 떠오르며 관련 연구들이 활발하게 진행되고 있다. 기존에 진행되는 연구들은 네트워크 아키텍처 관점에서 종단간 보안을 위해 DTLS를 사용하며 특히 성능이 제약된 기기에 생기는 부하를 줄이는 데 초점이 맞춰져 있다.

본 논문에서는 역시 네트워크 관점에서 DTLS 프로토콜의 부하를 줄이기 위해 기존의 X.509 인증서가 아닌 경량화된 인증서인 ECQV 인증서를 사용하는 DTLS 프로토콜을 제안한다. 또한 제안기법을 실제로 구현하고 기존의 보안 모드인 PSK, RPK 모드와 비교 및 분석한다.

ECQV Certificate Based Security Mechanism for End-to-End Security in IoT

Han-Beol Yeon*, Chang seop Park**

ABSTRACT

IoT technology is evolving and related services and technologies are spreading throughout the life. These IoT technologies make life easier for users, but they also have big threats like double-edged swords.Therefore, the importance of security is emerging and related researches are actively proceeding. Existing researches have focused on reducing the computational load on the constrained devices, performing the DTLS for the end-to-end security from a network architecture perspective. In this paper, we propose a DTLS protocol that uses ECQV certificate instead of existing X.509 certificate to reduce the load of DTLS protocol from the network perspective. In addition, the proposed scheme is implemented and compared with PSK and RPK modes.

Key words : 사물인터넷, 종단간 보안, DTLS, ECQV 인증서

접수일(2017년 3월 4일), 게재확정일(2017년 3월 27일)

★ 본 연구는 미래창조과학부 및 한국인터넷진흥원의 “고용계약형 정보보호 석사과정 지원사업”의 연구 결과로 수행되었음. (과제번호 H2101-16-1001)

★ 이 논문은 2016년도 정부 (미래창조과학부)의 재원으로 한국연구재단의 지원을 받아 수행된 과학기술인력교류활성화지원사업성과임. (NRF-2016H1D2A2916919)

* 단국대학교/컴퓨터학과

** 단국대학교/컴퓨터학과

1. 서 론

IoT(사물인터넷, Internet of Things) 기술은 최근 스마트 기기의 보급과 더불어 큰 주목을 받고 있으며 여러 기업에서 관련된 서비스와 제품을 개발 및 상용화하고 있다. IoT 환경은 성능이 제약된 기기(Constrained Device)로 이루어진 LoWPAN(Low-power Wireless Personal Area Networks)이다. 때문에 일반 데스크톱 환경의 보안을 적용하기 힘들어 상용화된 제품과 서비스에는 대부분 보안이 적용되어 있지 않다. 하지만 IoT 서비스는 스마트 홈, 자율주행 자동차, 의료기기 등과 같이 점차 인간의 생명과 일상생활에 밀접해지기 때문에 보안은 꼭 고려되어야 한다. 이에 국제 표준화 단체인 IETF의 여러 워킹 그룹과 국내외 대학 및 기업연구소에서 네트워크 관점에서의 보안이 연구되고 있다. 네트워크 관점의 보안은 기기와 환경에 크게 구애받지 않고 적용할 수 있다는 특징이 있으며 기존의 WSN(Wireless Sensor Network) 보안과 달리 종단간 보안을 요구한다는 차이점이 있다[1][2].

본 논문에서는 종단간 보안을 위해 ECQV(Elliptic Curve Qu-Vanstone) 인증서를 사용하는 DTLS(Datagram Transport Layer Security)를 실제 환경에서 구현하고 PSK(Pre-Shared Key), RPK(Raw Public Key)모드와 비교 및 분석을 진행한다. 본 논문의 구성은 다음과 같다. 2장에서는 IoT 네트워크 환경과 관련된 프로토콜들에 대해 소개하고 3장에선 Delegation Architecture를 이용해 DTLS 연결을 구현한 관련 연구들에 대해 소개한다. 4장에서는 ECQV 인증서를 사용하는 DTLS 기반의 프로토콜을 제안하고 5장에서는 실제 구현한 환경에서 진행한 성능분석에 대해 논하며 마지막으로 6장에서는 결론을 내린다.

2. IoT 네트워크 관련 프로토콜

2.1 IEEE 802.15.4

IEEE 802.15.4는 LR-WPANs(Low Rate Wireless Personal Area Networks)의 PHY계층과 MAC계층을 정의하는 표준이다[3]. 이 표준에서는 MTU(Maximum Transmission Unit)을 127바이트로 제한하고 있는데, 그 이상의 데이터를 전송할 시 단편화 및 재조립이 필요하다.

또한 IPv6 인프라를 사용할 경우 UDP(User Datagram Protocol)와 IPv6 헤더가 추가되는데 이 추가되는 헤더로 인해 사용 가능한 페이로드의 사이즈가 54바이트에 불과하게 된다. 이런 문제들을 해결하기 위해 6LoWPAN(IPv6 over Low power Wireless Personal Area Network)을 사용한다.

2.2 6LoWPAN

IETF의 6LoWPAN 워킹 그룹에서 제정된 표준으로, 데이터 전송 속도가 느린 IEEE 802.15.4를 기반으로 하는 LoWPAN에서 헤더 사이즈가 큰 IPv6 패킷을 효율적으로 전달할 수 있도록 해주는 적응 계층(Adaptation Layer)이다[4]. IEEE 802.15.4를 기반으로 한 환경에서 IPv6와 UDP를 사용하게 된다면 큰 헤더의 사이즈 때문에 헤더 압축 기술이 필요하다.

6LoWPAN은 헤더 압축을 지원하는 데 이를 이용해 헤더를 최대로 압축한다면 사용가능한 페이로드가 최대 97바이트까지 늘어나게 된다. 추가적으로 6LoWPAN은 단편화와 재조립 및 Mesh-routing 등의 기술을 지원한다.

2.3 RPL

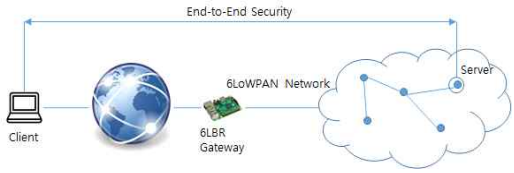
RPL(IPv6 Routing Protocol for Low-power Lossy Networks)은 IETF의 ROLL(Routing Over Low power and Lossy networks) 워킹 그룹에서 제정된 표준이다[5]. RPL은 성능이 제약된 노드들로 이루어진 LLNs 환경에서 루트를 중심으로 DODAG(Destination Oriented Directed Acyclic Graph)를 형성하여 최적의 라우팅 경로를 설정하는 프로토콜이다.

이 프로토콜에서는 루트와 노드의 거리를 나타내는 값인 RANK를 사용하는 데, 이 값을 포함한 DIO(DODAG Information Object)와 DAO(DODAG Advertisement Object) 메시지를 이용해 자신의 위치를 계산하고 부모와 자식 관계를 형성해 최적의 경로를 파악한다.

2.4 CoAP

CoAP(Constrained Application Protocol)은 IETF

의 CoRE(Constrained RESTful Environment) 워킹 그룹에서 제정된 표준으로 UDP를 기반으로 한 IoT 환경을 애플리케이션 단에서 제어하기 위한 목적을 가진다[6]. HTTP를 기반으로 한 이 표준에서는 데이터의 기밀성과 무결성 등의 보안을 적용하기 위해 DTLS의 사용을 권장한다. CoAP은 DTLS를 사용해 세 가지 보안모드인 PSK, RPK, 인증서 모드를 지원한다.



(그림 1) End-to-End Security

3. 관련연구

3.1 위임 아키텍처에서의 DTLS 연결과 재개

IoT 환경은 주로 성능이 제약된 기기 C들로 구성되어 있다. 이 기기들은 종단간 보안을 위해 통신 주체간의 DTLS 연결을 진행하며 세션키를 생성한 뒤 해당 세션키를 사용해 안전한 통신을 할 수 있다. 하지만 DTLS연결은 C들에게 많은 부하를 발생시키기 때문에 성능이 저하될 수 있다. 만약 수많은 기기가 서비스를 제공하는 상태라면 작은 부하라도 서비스에 큰 문제를 야기할 수 있다. 때문에 이 기법에서는 부하를 대신할 위임 아키텍처(Delegation Architecture)를 사용해 부하를 줄인다[7].

이 기법에서 제안된 방식은 다음과 같이 동작한다. 위임 서버(Delegation Server)는 먼저 C의 역할을 대신하기 위해 사전에 Cipher Suite를 전달받고 안전한 Security Context 교환을 보호하기 위해 대칭키를 공유한다. 이후 위임 서버가 클라이언트와 DTLS Handshake를 진행한 후 세션에 관련된 정보를 Session Ticket이라는 형태로 저장한다. Session Ticket은 C에게 전해지며 이것을 사용해 기존의 세션을 재개할 수 있다. 이 기법은 C에게 DTLS Handshake로 인한 오버헤드를 줄일 수 있다. 또한 원할 때 DTLS 연결을 재설정하고 끊을 수 있다는 특징이 있다.

3.2 선택적 위임 아키텍처를 이용한 DTLS

이 기법 역시 성능이 제약된 기기 C에서 DTLS Handshake를 진행 할 때 부하를 줄이기 위해 제안된 기법이다. 하지만 3.1의 제안 기법처럼 모든 Handshake가 대신해서 진행되는 것이 아니고 많은 연산이 필요한 일부만 선택적으로 대신해 진행된다[8].

이 논문에서 제안된 기법은 다음과 같이 동작한다. 대리 서버는 DTLS Handshake가 진행되기 이전에 먼저 성능이 제약된 기기에게 인증서, Cipher Suite, 키 재료를 받는다. 여기서 대리 서버는 DTLS Handshake 일부를 대신 진행하는 기기로 1) C의 암호화 키와 인증서를 관리하고, 2) 클라이언트의 인증서를 분석하며, 3) 클라이언트와 C간의 상호 인증, 4) 클라이언트와 C간의 세션 설정, 5) C와 Transfer of Session 작업을 수행하는 역할을 한다.

이후 DTLS Handshake가 진행될 때 C는 대리 서버와 대칭키 모드로 Handshake를 진행하며 동시에 대리 서버는 클라이언트와 인증서 모드로 Handshake를 진행하고 세션키를 생성한다. 이후 클라이언트와 Handshake가 종료되기 전에 해당 세션의 정보를 C에게 전달하는 Transfer of Session이 진행된다. 이 단계에서 세션의 정보를 Session Ticket 형태로 C에게 전달하며 이후 Handshake가 정상적으로 종료되면 그대로 성능이 제약된 기기와 클라이언트간에 DTLS 프로토콜이 속개된다. 이 기법은 C에서 인증서 기반의 DTLS를 사용할 수 있게 한다.

3.3 기존 연구들에 대한 평가

앞에서 살펴본 기존 연구들은 모두 성능이 제약된 기기 C들에게 오버헤드를 최대한 줄인 DTLS를 사용할 수 있는 기법을 제안하였다. 제3의 기기를 대리 서버로 사용하여 대신 DTLS Handshake를 진행하게 하였고 그 때 만들어지는 세션 정보를 Session Ticket에 담아 C에게 전달해 DTLS 세션을 속개하는 방식으로 오버헤드를 줄였다.

하지만 기존 연구들은 대리 서버에 대한 의존도가 높아 만약 대리 서버가 어떠한 공격을 당하거나 물리적으로 사용할 수 없는 상태가 된다면 C와 클라이언트 간의 DTLS 연결이 불가능해진다는 문제점을 가

지고 있다. 또한 대리 서버와 C간의 사전 단계로 키를 공유하는 데, 이러한 경우 PSK 기법과 크게 다른 모습을 보이지 않는다. 때문에 본 논문에서 제안하는 기법은 대리 서버를 사용하지 않고 C와 클라이언트 양자 간에 직접 DTLS를 수행할 수 있도록 한다. 하지만 기존의 DTLS가 기본으로 지원하는 보안 모드 중 PSK 모드의 경우 가장 간편하고 빠르지만 사전에 모든 기기들과 PSK를 공유해야한다는 조건이 요구되는데 이것은 많은 기기를 배치하는 IoT 환경에 적합하지 않은 선행 작업이며 RPK 모드의 경우 인증서의 out-of-band 유효성 검증이 필요해 오버헤드가 발생할 수 있다는 단점이 있고 인증서 모드의 경우 X.509 인증서가 굉장히 큰 크기를 가지고 있기 때문에 단편화가 발생하여 이 또한 오버헤드를 발생할 수 있다는 단점들이 존재한다.

때문에 상호 인증과 키 생성을 위해 X.509 인증서 대신 ECQV 인증서를 사용하도록 한다. ECQV 인증서는 X.509 인증서에 비해 가벼우며 빠른 연산 속도를 가지고 있고 인증서의 out-of-band 유효성 검증이 필요하지 않다는 장점을 가지고 있다.

4. ECQV 인증서를 사용하는 DTLS

4.1 제안 기법과 관련된 기술

4.1.1 표기법

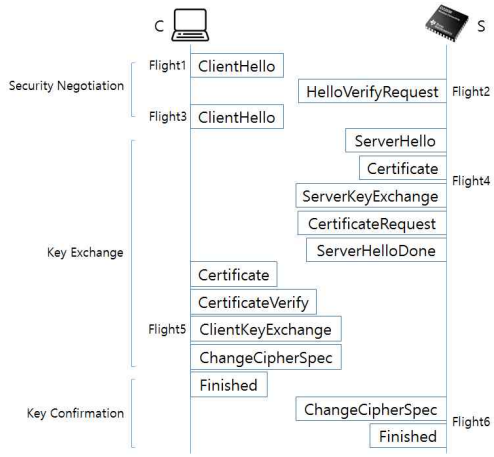
<표 1> 표기법

기호	표기법
rt	A의 인증서
H	해시 함수
Q_A	A의 공개키
q_A	A의 개인키
A_d	A의 식별자

4.1.2 DTLS

DTLS는 통신 주체간의 종단간 보안을 위해 TCP (Transmission Control Protocol)에서 사용되는 프로토콜인 TLS를 UDP에서 사용할 수 있도록 만든 프로토콜이다. DTLS는 UDP에서만 발생할 수 있는 데이터 손실, 단편화, 리오더링, 리플레이 같은 통신상의 문제들을 처리하기 위한 추가적인 기능을 가지고 있다[9].

이 프로토콜은 안전한 통신을 위해 사용되는 암호화 키 생성과 상호 인증을 위한 Handshake 작업이 진행된다. DTLS Handshake에서는 새로운 연결을 세팅하고 보안협상, 키 교환 및 교환된 키를 기반으로 세션키를 생성하는 키 확인의 세 단계로 이루어진다. 이 프로토콜은 메시지들의 묶음인 6개의 Flight 단위로 전송된다.



(그림 2) DTLS Handshake 전체 메시지

(1) Flight 1, 2, 3

가장 먼저 클라이언트가 서버에게 ClientHello 메시지를 보냄으로써 Handshake 프로토콜이 기동된다. 이 메시지에는 클라이언트가 지원하는 Cipher Suite(해시 알고리즘, 압축 알고리즘)와 랜덤 넘버 등이 포함된다. 만약 이 메시지가 지속적으로 보내지는 DoS 공격이 발생할 경우를 대비해 HelloVerifyRequest와 두 번째 ClientHello 메시지에 Cookie라는 값을 사용한다. 이 Cookie 값은 서버측에서 HelloVerifyRequest 메시지에 넣어 보내며 이 값을 클라이언트가 다시 보냄으로 정상적인 연결이 수립되는 것을 확인한다. 이후 ServerHello 메시지에 Cipher Suite의 리스트 중 선택된

한 가지와 랜덤 넘버를 ServerHello 메시지에 포함하여 보낸다. Cipher Suite는 Handshake 과정에서 사용될 보안 모드, 암호화 기법, MAC 알고리즘 등의 집합이다. 또한 랜덤 넘버는 이후에 세션키를 만드는 데 사용된다.

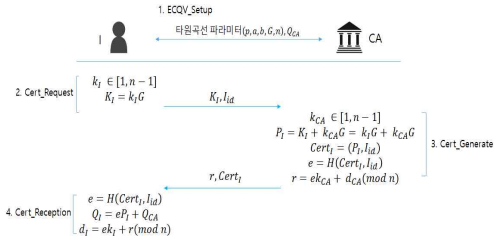
(2) Flight 4, 5, 6

Flight 4에서 서버를 인증하기 위한 인증서와 키를 만드는 데 사용되는 추가적인 데이터들이 보내지며 Flight 5에서 클라이언트 측의 인증서와 다른 재료들을 보내며 상호 인증이 이루어진다. 또한 이 단계에서 상대방의 키를 사용해 세션키를 생성하며 만들어진 세션키를 사용해 모든 Handshake 메시지를 MAC 함수를 사용한 후 Finished 메시지에 포함해 보내게 된다. 서버 또한 같은 방법으로 세션키를 만들고 모든 Handshake 메시지를 MAC 함수를 사용하여 받은 Finished 메시지와 같은지 검증한 후 세션이 종료되게 된다.

4.1.3 ECQV 인증서

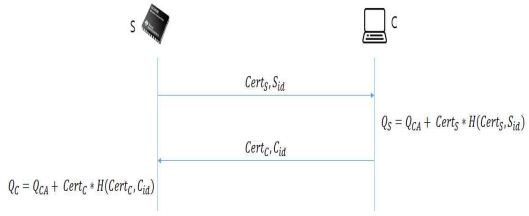
기존에 널리 사용되는 X.509 인증서는 PKI(Public Key Infrastructure)에서의 표준 인증서 형식이다. 이 인증서는 CA(Certificate Authority)가 공개키, 서명, 식별자 등의 데이터를 사용해 생성한다. X.509 인증서 내부에는 공개키가 명시되어 있으며 CA와의 통신을 통해 해당 인증서에서 공개키를 추출할 수 있다. 이러한 기존의 인증서는 IoT 환경에 적합하지 않은데 이유는 사이즈가 수백 바이트에 이르기 때문에 단편화와 재조립이 발생할 수 있으며 out-of-band 유효성 검증이 필요해 부하가 생기기 때문이다. 때문에 이에 비해 굉장히 작고 연산이 빠른 ECQV가 주목받고 있다.

ECQV 인증서는 Implicit Certificate의 종류 중 하나로 공개키와 전자서명을 별도로 명시하는 기존의 X.509 인증서와 달리 공개키와 전자서명을 하나의 Reconstruction 데이터로 융합해 인증서로 사용한다. CA의 공개키를 가지고 있다면 특정 연산을 통해 이 데이터에서 소유자의 공개키를 쉽게 추출할 수 있으며 사이즈 또한 EC(Elliptic Curve)포인트 1개의 크기밖에 되지 않고 인증서의 유효 기간을 짧게 하여 CA와의 통신을 통한 인증서 취소 여부 확인의 부담을 줄여 추가적인 부하가 발생하지 않기 때문에 IoT 환경에 적합하다. ECQV 인증서는 (그림 3)과 같은 발급 과정을 거쳐 발급된다.



(그림 3) ECQV 인증서 발급과정

먼저 발급자는 CA와 타원곡선 파라미터를 합의하고 CA의 공개키를 받는다. 그 후 랜덤한 값을 생성하고 자신의 식별자와 함께 발급 요청을 하면 CA는 받은 데이터와 추가적인 재료들로 발급자의 ECQV 인증서를 발급해준다. 해당 ECQV 인증서를 통해 발급자는 자신의 공개키와 개인키를 계산해낼 수 있으며 (그림 4)와 같이 CA의 공개키만 있다면 인증서 소유자의 공개키도 계산해낼 수 있다.



(그림 4) ECQV 인증서 소유자의 공개키 추출

4.2 제안기법 및 환경

본 논문에서는 종단간 보안을 위해 기존의 DTLS 프로토콜에 새로운 보안모드인 ECQV 인증서를 사용하는 모드를 제안한다.

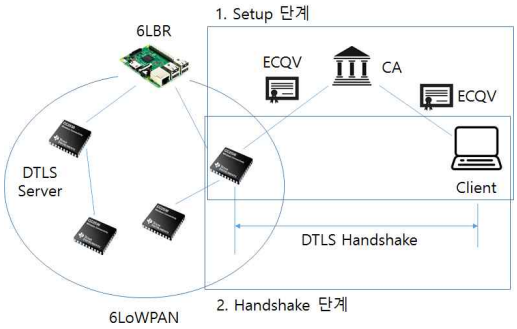
제안기법은 서버와 클라이언트가 CA로부터 ECQV 인증서를 발급받는 Setup 단계와 발급받은 ECQV 인증서를 사용해 DTLS Handshake를 진행하는 Handshake 단계의 두 단계로 진행된다. 전체적은 제안기법의 흐름은 (그림 6)과 같다.

(1) Setup 단계

DTLS Handshake에서 ECQV 인증서를 사용하기 위해서는 사전에 ECQV 인증서를 가지고 있어야 한다. 때문에 서버와 클라이언트는 실제 환경에 배치되기 전에 CA와의 통신을 통해 자신의 ECQV 인증서를 발급받는다.

먼저 각 주체는 CA와 통신하여 타원곡선 파라미터 $(,a,b, ,n)$를 합의하고 CA의 공개키인 $\boldsymbol{Q}_{CA}$를 받는다. 타원곡선 $E(F)$가 $y^2 \equiv x^3 + ax + b (\mathrm{mod} p)$일 때 p는 prime이고 G는 generator이며 n은 타원곡선의 위수(order)이다. 여기서 합의한 타원곡선 파라미터는 ECQV 인증서 발급과정에서 사용되는 값들이다.

이후 ECQV 인증서 발급과정에 따라 CA에게 자신의 식별자와 랜덤 값 등의 데이터와 함께 발급 요청을 하며 ECQV 인증서를 발급받는다. 제안에서 서버는 실제 IoT 환경을 관리하는 관리자를 통해 ECQV 인증서를 발급받고 물리적으로 ECQV 인증서를 저장한다.

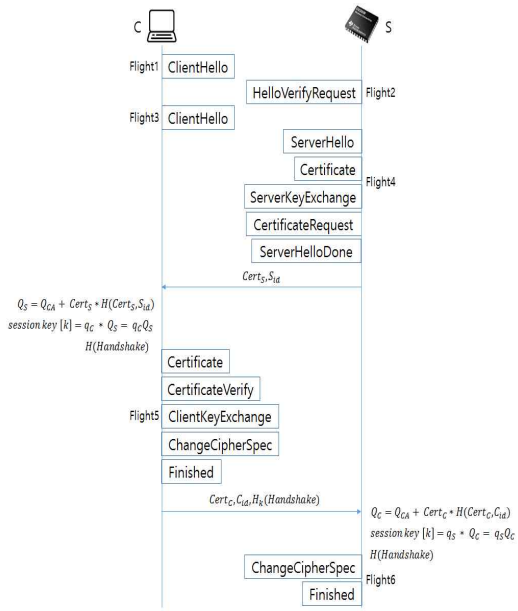


(그림 5) 제안기법의 두 단계

(2) Handshake 단계

클라이언트가 LoWPAN의 어떠한 노드와 안전한 통신을 원할 때 해당 노드에게 DTLS 연결을 요청한다. 이 때 클라이언트는 LoWPAN 내의 모든 노드들의 주소를 담은 리스트를 가지고 있다고 가정한다.

이 때 Handshake는 ECQV 인증서를 사용해 아래의 (그림 6)에 보이는 것처럼 DTLS 인증서 모드와 동일하게 진행된다. 이 모드에서 사용되는 Cipher Suite는 TLS_ECDHE_ECQV_WITH_AES_128_CCM_8이다.



(그림 6) 제안 기법

5. 구현 및 성능평가

5.1 구현 환경 및 장비소개

5.1.1 경량화된 DTLS

TinyDTLS는 IoT 환경에서 사용될 수 있도록 개발된 경량화된 오픈소스 DTLS이다. 기본 보안모드로 PSK, RPK 모드를 지원하며 큰 크기의 X.509 인증서를 사용하는 인증서 모드는 성능상의 이유로 지원하지 않는다.

추가적으로 TinyDTLS의 RPK 모드에서 사용하는 ECC 모듈은 최적화가 되어 있지 않기 때문에 micro ECC 모듈을 별도로 적용해 사용하였다. micro ECC 모듈은 C로 구현된 모듈로 코드의 사이즈를 줄이고 ECDH와 ECDSA(Elliptic Curve Digital Signature Algorithm)연산을 더욱 빠르게 수행할 수 있게 만들어졌다.

5.1.2 장비 소개

SmarfRF06 Evaluation Board	CC2538EM
	
IP Time 505	Raspberry PI 2
	

(그림 7) 사용 장비

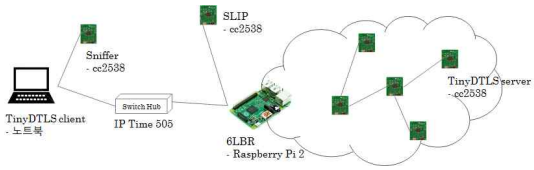
Raspberry PI 2는 기초 컴퓨터 과학의 교육을 위해 개발된 작은 싱글 보드 컴퓨터로 실험환경에서는 6LBR(6LoWPAN Border Router)를 올려 라우터로 사용하였고 LoWPAN을 구성하는 센서노드는 CC2538EM을 사용하였다. CC2538EM은 ARM Coretex-M3 기반의 MCU, 최대 32KB on-chip RAM과 최대 512KB on-chip flash와 IEEE 802.15.4 라디오를 가지고 있다. IP Time 505는 Switch Hub로 이더넷을 통해 클라이언트와 6LBR을 연결해 유선 측 IPv6 전용망을 구성하는 데 사용하였다.

마지막으로 SmartRF06 Evaluation Board는 CC2538EM을 장착해 애플리케이션을 다운받거나 디버깅하는 데 사용하였다.

5.1.3 실험 방법

본 논문의 실험 환경은 (그림 8)과 같다. 노트북과 Raspberry PI 2에 각각 Contiki OS를 설치한 후 Switch Hub를 이용해 유선 전용망을 구성하였다. 이후 여러 개의 CC2538EM에 TinyDTLS Server를 포팅하여 LoWPAN을 구성한 후 노트북에서 TinyDTLS Client를 기동시켜 LoWPAN안의 노드와 TinyDTLS 연결을 진행하였다.

Handshake에서 사용되는 ECQV 인증서는 CA에게 발급받았다는 가정으로 사전에 임의로 생성하여 물리적으로 저장하였다. 실험에서는 TinyDTLS에서 기본적으로 지원하는 PSK, RPK 모드와 직접 구현한 ECQV 인증서 모드로 DTLS 세션을 실제로 기동시켜 성능 분석을 하였다.

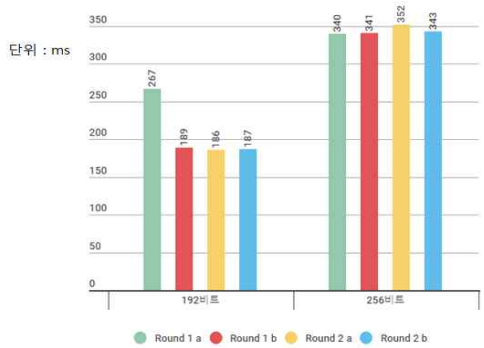


(그림 8) 구현 환경

5.2 성능평가

5.2.1 ECDH 시간 측정

이 절에서는 RPK 모드와 ECQV 모드에서 세션키를 만드는 데 사용되는 ECDH 알고리즘의 시간 소모량을 측정하였다. (그림 9)에 보이는 결과처럼 각각 키의 길이를 192비트, 256비트로 설정해 실험하였으며 Round 1은 자신의 공개키 쌍을 생성하는 부분, Round 2는 ECDH 연산을 통해 세션키를 생성하는 부분이다. 또한 a, b는 각 통신주체를 의미한다.

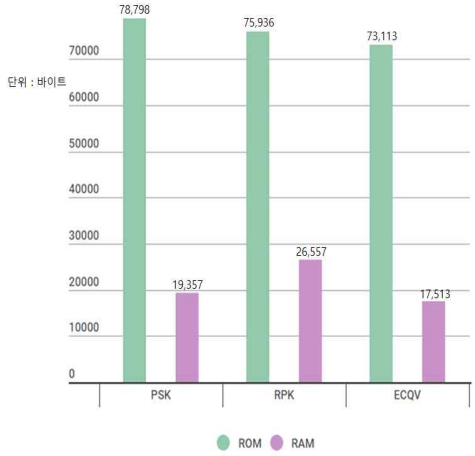


(그림 9) ECDH 모듈 연산 시간 측정 그래프

5.2.2 메모리 사용 측정

이 절에서는 각 모드의 RAM과 ROM 소모량을 측정하였다. TinyDTLS 소스를 컴파일하면 *.elf 파일이 생성되는데 이 파일을 arm-none-eabi-size 유틸리티를 사용해 측정한 값을 공식에 따라 계산하였다. 계산 결과 (그림 10)과 같이 세 가지 보안모드는 거의 차이가 나지 않았지만 미세하게 ECQV 모드가 더 적은 메모리 소모량을 나타내었다. 특히 RAM 소모량의 경우 RPK 모드의 RAM 소모량에 비해 약 65%에 불과한

모습을 나타내어 메모리 소모량 측면에서 봤을 때 기존의 모드들에 비해 IoT 환경에 더욱 적합한 것으로 판단된다.



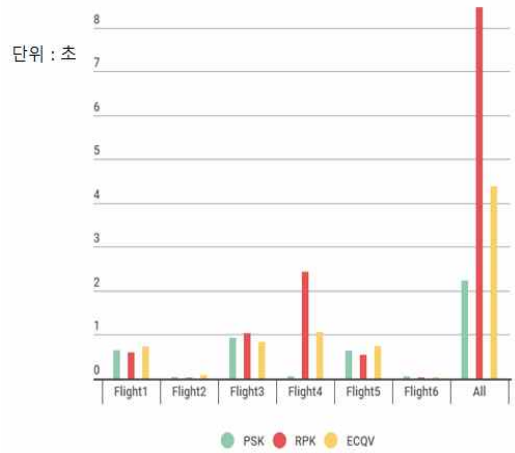
(그림 10) 보안모드 별 RAM/ROM 소모량

5.2.3 Handshake Flight 별 연산 시간

이 절에서는 DTLS Handshake를 각 모드별로 기동시켜 걸리는 시간을 (그림 11)에서 보이는 것과 같이 Flight 별로 측정하였다. Flight의 시작 부분과 끝 부분에 시간 측정 함수를 삽입해 측정하였으며 서버의 경우 RTIMER()함수를 클라이언트의 경우 gettimeofday()함수를 사용하였다.

(그림 11)에서 볼 수 있듯이 먼저 Flight 1과 3은 ClientHello 메시지만 전송하는 부분이다. 하지만 Flight 3에서만 시간이 증가한 것이 눈에 띄었는데 이는 DoS 공격 방지를 위한 Cookie 값의 검증 부분이 들어가 있기 때문이다.

또한 Flight 4에서 RPK 모드가 ECQV 모드에 비해 많은 시간이 소모되는 것을 확인할 수 있었다. 이는 RPK 모드에서 ECDSA와 관련된 연산이 추가되어 있기 때문이다. 전체적인 시간 소모량을 봤을 때 ECQV 모드가 RPK 모드에 비해 약 50% 정도의 시간 감소를 보였다. PSK 모드의 경우 가장 빠른 시간을 보였지만 대량의 기기와의 PSK를 저장해야 한다는 점이 현실적으로 어렵다는 문제가 존재한다.



(그림 11) 각 보안 모드의 Flight 별 소요 시간

6. 결 론

일반적으로 종단간 보안을 위해서 DTLS 프로토콜이 사용되는데 IoT 환경에서는 성능의 한계로 인해 사용되기가 어렵다. 때문에 기존 연구들에서 소개한 기법들처럼 성능이 제약된 기기의 부하를 줄이기 위한 연구들이 진행되고 있다.

소개한 기법들 같은 경우 대리 서버를 사용해 부하를 줄이는데 대리 서버와 성능이 제약된 기기 간에 사전에 공유되는 키가 필요하며 대리 서버가 사용할 수 없는 상태가 된다면 DTLS 연결이 불가능하다는 한계가 존재한다. 만약 대리 서버를 사용하지 않고 직접적인 DTLS 연결을 진행한다면 각 모드 별로 문제점이 존재한다. PSK 모드의 경우 대량의 기기들과의 PSK를 저장해야 한다는 점, RPK 모드의 경우 인증서 검증을 위해 out-of-band 검증이 이뤄지기 때문에 생기는 부하가 존재한다는 점, X.509 인증서 모드의 경우 큰 사이즈로 인한 단편화 및 out-of-band 검증으로 인한 부하가 발생한다는 문제점들이다.

때문에 본 논문에서는 직접적인 DTLS 연결을 하기 위해 경량화된 인증서인 ECQV 인증서를 사용하였다. ECQV 메커니즘은 가벼운 ECQV 인증서를 사용해 상호 인증은 물론 키 추출도 간편하다는 장점들이 존재한다. 실제로 구현 및 성능평가를 진행하였고

메모리 소모량, 시간 소모량의 측면에서 기존의 보안 모드들에 비해 월등한 모습을 보였다.

참고문헌

[1] Granjal, Jorge, Edmundo Monteiro, and Jorge Sá Silva, "Security in the integration of low-power wireless sensor networks with the internet: A survey," Ad Hoc Networks 24, pp. 264-297, 2015.

[2] Roman, Rodrigo, Pablo Najera, and Javier Lopez, "Securing the internet of things," Computer 44.9, pp. 51-59, 2011.

[3] IEEE Computer Society. IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs), 2006.

[4] Kushalnagar, Nandakishore, Gabriel Montenegro, and Christian Schumacher, "IPv6 over low-power wireless personal area networks (6LoWPANs): overview, assumptions, problem statement, and goals," No. RFC 4919, 2007.

[5] Winter, Tim, "RPL: IPv6 routing protocol for low-power and lossy networks," 2012.

[6] Shelby, Zach, Klaus Hartke, and Carsten Bormann, "The constrained application protocol (CoAP)," No. RFC 7252, 2014.

[7] Hummen, R., Shafagh, H., Raza, S., Voig, T., & Wehrle, K, "Delegation-based Authentication and Authorization for the IP-based Internet of Things," 2014 Eleventh Annual IEEE International Conference on Sensing, Communication, and Networking (SECON), IEEE, pp. 284-292, 2014.

[8] dos Santos, G. L., da Cunha Rodrigues, G., Granville, L. Z., & Tarouco, L. M. R, "A DTLS-based security architecture for the Internet of Things," 2015 IEEE Symposium on Computers and Communication (ISCC), IEEE, pp. 809-815, 2015.

[9] 권혁진, and 강남희, "사물인터넷에서 경량화 장치 간 DTLS 세션 설정 시 에너지 소비량 분석," 한국통신학회논문지 40.8, pp. 1588-1596, 2015.

[저자소개]

연 한 별 (Han-Beol Yeon)
2014년 8월 단국대학교
컴퓨터과학과 학사
2017년 2월 단국대학교
컴퓨터학과 석사

email : hihihi89@gmail.com

박 창 섭 (Chang-seop Park)
1983년 2월 연세대학교
경제학과 학사
1987년 2월 Leigh University
컴퓨터과학과 석사
1990년 2월 Leigh University
컴퓨터과학과 박사
1990년 3월 ~ 현재 단국대학교
컴퓨터학과 교수

email : csp0@dankook.ac.kr