Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.6, Dec. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.6.1307

DTLS 기반의 안전한 CoAP 응용을 위한 접근제어 메커니즘*

정 연 성,† 박 창 섭‡
단국대학교

Access Control Mechanism for Secure CoAP Applications Based on DTLS*

Yeon-seong Jeong,† Chang-seop Park‡
Dankook University

요 약

DTLS의 PSK 모드는 DTLS 핸드쉐이크의 성능 측면에서 가장 효율적이지만 센서 디바이스의 개수가 증가함에 따라 대칭키 쌍을 미리 배포하는 것과 관리하는 것이 용이하지 않다. 반면에 RPK 모드와 인증서 모드는 키 관리가 용이하지만 계산상의 효율성은 매우 떨어진다. 본 논문에서는 자원 제약적인 센서 디바이스에 적합한 인증서인 ECQV를 통한 인증서 모드와 PSK 모드를 사용하여 그룹 환경에서의 종단 간 보안을 달성하기 위한 프로토콜을 제안한다. Initial DTLS 핸드쉐이크는 ECQV 인증서 모드를 사용하고, 이후 동일 그룹에 속한 CoAP 서버와의 Subsequent DTLS 핸드쉐이크는 PSK 모드를 사용하기 때문에 그룹내 CoAP 서버들과의 전체적인 계산 부담을 감소시킬 수 있다. 제안 프로토콜에서는 특정 CoAP 클라이언트가 그룹의 제한된 CoAP 서버에 접근 할 수 있게끔 세밀한 접근제어가 행해진다.

ABSTRACT

While the PSK mode of the DTLS is the most efficient in terms of the performance, it is not easy to pre-distribute and manage the symmetric key pairs as the number of sensor devices increases. On the other hand, both the RPK and certificate modes offer a convenient key management tool, but they do not guarantee a good computational performance. In this paper, the end-to-end security protocol suitable for the constrained devices is proposed, based on both the ECQV certificate and the PSK mode. Namely, the initial DTLS handshake is performed using the ECQV certificate, and the subsequent DTLS handshakes with the other CoAP servers in the same group are performed using the PSK mode for the purpose of reducing the overall computational load. Furthermore, a fine-grained access control for the CoAP client can be enforced to allow access to the limited number of CoAP servers.

Keywords: DTLS, CoAP, IoT

I. 서 론

IoT(Internet of Things)의 관심이 증가함에 따라 보급이 확산되고 기반 제품과 서비스가 빠르게 증가되고 있다. 특히 IoT 운영환경이 기존 폐쇄적 환경에서 개방적 환경으로 확장되어짐에 따라 개인

Received(10. 16. 2017), Modified(11. 27. 2017), Accepted(11. 27. 2017)
* 본 연구는 미래창조과학부 및 한국인터넷진흥원의 "고용계약형 정보보호 석사과정 지원사업"의 연구결과로 수행되었음 (과제번호 H2101-17-1001)
* 이 논문은 2017년도 정부(교육부)의 재원으로 한국 연구재단의 지원을 받아 수행된 기본 연구지원사업 성과임. (NRF-2017R1D1A1B03027862)
† 주저자, wjddustjd112@naver.com
‡ 교신저자, csp0@dankook.ac.kr(Corresponding author)

및 조직의 민감한 정보까지 다루게 되어 IoT에 대한 보안의 중요성은 높아지고 있다. 전통적인 WSN (Wireless Sensor Network)에서는 여러 센서 디바이스들을 관리하는 프록시가 존재하고, 프록시가 센서 디바이스의 센싱정보를 수집하면 원격 클라이언트가 프록시에게 정보를 요구하는 방식이기 때문에 직접 센서 디바이스에의 접근은 허용되지 않았다. 센서 디바이스와 프록시 사이에는 IEEE 802.15.4 보안 [1]이나 ZigBee 보안과 같은 링크 계층의 보안이 필요하며 프록시와 클라이언트 사이에는 IPsec이나 TLS와 같은 네트워크 또는 전송 계층 보안이 요구된다. 따라서 클라이언트와 센서 디바이스 사이에 이음새 없는 보안(Seamless Security)이 적용되지는 않는다. 반면에, WSN이 연동되는 IoT 네트워크 아키텍처 기반의 응용에서는 종단 간(end-to-end) 보안이 필수적이다. IPsec 및 TLS는 계산, 메모리 및 전력 면에서 자원 제약적인 센서 디바이스에는 적합하지 않기 때문에 종단 간 보안에는 사용될 수 없다. 따라서 IoT 네트워크 아키텍처 기반의 종단 간 보안을 위해서는 센서 디바이스와 클라이언트 사이의 비대칭 성능을 고려한 보안 프로토콜이 요구된다. DTLS (Datagram TLS) [2]는 종단 간 비대칭 성능에 대한 문제를 보완할 수 있는 IoT 보안을 위한 사실상의 표준이다. CoAP (Constrained Application Protocol) [3]은 DTLS의 보안모드로 PSK(Pre-Shared Key), RPK(Raw Public Key) 및 인증서 모드를 정의 하고 있다. 각 보안모드에는 장단점이 존재한다. PSK 모드는 성능 면에서 가장 적합한 선택이지만 CoAP 서버와 클라이언트가 많아질수록 고유한 대칭키를 미리 배포 하는 것이 쉽지 않다. 반면에, RPK와 인증서 모드는 ECDSA(EC-Digital Signature Algorithm) 연산이 필요하기 때문에 센서 디바이스의 계산부하가 PSK보다 상대적으로 높다. 하지만 PKI(Public Key Infrastructure)로 인해서 키 관리가 상대적으로 용이한 측면이 있다. 인증서 모드에서 사용되는 인증서는 X.509를 준용하기 때문에 호환성과 확장성이 높지만, 인증서의 크기가 크고 계산부하가 상대적으로 높기 때문에 자원 제약적인 센서 디바이스가 활용되는 IoT 네트워크에서는 적합하지 않다. 결국, 전통적인 인증서보다 상대적으로 더 작은 암시적 인증서인 타원곡선 기반의 ECQV (EC-Qu-Vanstone) [4]를 활용하는 것이 유리하다.

본 논문에서는 CoAP 클라이언트가 미리 할당 된 그룹내의 CoAP 서버들에 접근 할 수 있는 그룹 지향적 종단 간 보안을 고려한다. 본 연구의 핵심적인 내용은 다음과 같다. 첫째, CoAP 클라이언트와 CoAP 서버 그룹간의 보안연결을 설정한다. 이를 위해 CoAP 서버 및 클라이언트는 각각 ECQV 인증서 및 인증된 RPK를 사용한다. 둘째, DTLS 핸드쉐이크로 인한 총 계산량을 감소시키기 위해 "Initial DTLS 핸드쉐이크"를 마치면서 PSK 토큰을 발급받고, 동일 그룹내의 다른 서버와의 "Subsequent DTLS 핸드쉐이크"는 PSK 모드로 진행된다. 셋째, 동일 그룹내의 CoAP 서버들에 대한 세분화된 접근제어 메커니즘(Fine-Grained Access Control)을 내재시키기 위해 CoAP 클라이언트에게 발급하는 "Capability ID"의 개념을 도입한다.

본 논문의 구성은 다음과 같다. 2장에서는 DTLS와 관련된 기존연구에 대해서 설명하고 3장에서는 그룹 지향적 종단 간 보안 연결을 위한 DTLS 핸드쉐이크 프로토콜을 제안한다. Capability List에 따른 CoAP 서버 접근제어에 대해서도 논의된다. 4장에서는 안전성 분석을 하고 제안된 프로토콜과 다른 프로토콜과의 성능을 비교 분석한다. 마지막으로 5장에서는 결론을 내린다.

II. DTLS 기반 CoAP 보안 및 관련연구

2.1 DTLS 핸드쉐이크

DTLS 핸드쉐이크 과정은 보안협상(Security Negotiation), 키교환(Key Exchange) 그리고 키확인(Key Confirmation) 단계로 구성된다. 먼저 보안협상 단계에서는 보안모드(Security Mode)와 Cipher Suit가 결정되고, 추가적으로 Nonce가 교환된다. 보안모드에 따라 "Pre-Master Secret"이라는 인증키 AK가 도출되고 키교환 단계에서 AK와 Nonce를 이용해 "Master Secret"이라는 세션키 SK가 도출 된다. 마지막으로 CoAP 서버와 클라이언트는 키확인 단계를 통해 SK를 기반으로 한 상호인증을 수행한다.

DTLS의 PSK 모드에서 CoAP 서버는 자신과 통신할 CoAP 클라이언트들과 공유하는 PSK가 탑재된 접근제어 목록을 유지한다. DTLS 핸드쉐이크를 시작하기 위해서는 별도의 방식을 통해 CoAP 서버와 해당 인가된 클라이언트에게 대칭키가 사전

제공되어야 한다. RPK 모드의 CoAP 서버와 클라이언트는 각각 ECDSA 공개키/개인키 쌍을 유지하며 PSK 모드와 같이 별도의 방식을 통해서 서로의 공개키를 획득해야 한다. 이 방식에서도 CoAP 서버는 인가된 클라이언트와 해당 공개키의 접근제어 목록을 구성 할 수 있다. 키확인 단계에서의 상호인증은 ECDSA를 기반으로 수행된다. 마지막으로 인증서 모드에서는 ECDSA를 기반으로 하는 X.509 인증서가 사용되며 CoAP 서버와 클라이언트는 ECDSA 공개키/개인키 쌍과 함께 X.509 인증서를 활용하게 된다.

2.2 기존 연구

IoT 네트워크상의 종단 간 보안을 위해서는 센서 디바이스와 클라이언트 사이의 비대칭 성능을 고려한 보안 프로토콜이 요구된다. PSK 모드는 센서 디바이스에 가장 적합한 선택이지만 키 관리 측면에서 바람직하지 않고, 반면에 인증서 모드는 CoAP 서버의 계산 부담을 야기한다. 따라서 관련 기존연구의 대부분은 CoAP 서버와 클라이언트의 비대칭 성능을 보완하기 위해 위임서버(Delegation Server), 접근제어 서버 또는 6LBR (6LoWPAN Border Router) 등으로 명명된 SP(Security Proxy)를 활용하고 있다. [5, 6]에서는 SP가 위임서버의 역할을 수행한다. 즉, 클라이언트는 SP와 DTLS 핸드쉐이크를 진행하고 해당 세션이 생성이 되면 세션 ID, Cihper Suit 등의 상태정보를 CoAP 서버에 전달하여 세션을 재개한다. [7, 8]에서는 SP가 접근 제어 서버의 역할을 수행한다. [7]에서는 SP가 서버와 사전에 대칭키를 공유하며 클라이언트는 접근하려는 서버에 대한 대칭키를 SP에게 요구한다. SP로부터 전달받은 대칭키를 통해서 DTLS 핸드쉐이크를 진행하게 된다. [8]은 클라이언트가 SP를 통해서 티켓을 발급 받는 과정을 기술하고 있지만 티켓을 활용한 서버와의 DTLS 핸드쉐이크에 대한 구체적인 설명은 나타나 있지 않다. [9]에서는 SP가 자원 제한적인 센서 디바이스가 수행하기에 부담이 되는 DH(Diffie-Hellman) 연산을 대리 수행한다. 클라이언트가 자신의 DH 공개키를 분할해서 다수의 SP에서 보내면 SP는 DH 연산을 통해서 암호화된 키 정보를 전송한다. 서버는 SP로부터 받은 키 정보들을 복호화해서 클라이언트의 공개키를 도출하게 된다. [10]에서는 6LBR이 SP의 역할을 수행하며 자원 제한적인 센서 디바이스와 클라이언트 간에 DTLS 핸드쉐이크를 중재한다. 즉, ECC 공개키 암호의 계산 부하는 자원 제약적이지 않은 6LBR로 전가된다. 기존의 연구들은 SP를 활용하여서 자원이 제약적인 센서 디바이스의 부담을 줄이는 방법을 제안하였다. 하지만 DTLS 핸드쉐이크를 진행하면서 SP에 키 관련 정보가 노출된다. [5]에서는 생성된 세션에 대한 정보를 관리하기도 한다. 종단 간 보안 측면에서 키 정보나 세션의 정보가 보안 연결을 필요로 하는 해당 객체 이외의 다른 객체도 알고 있다는 것은 문제가 될 수 있다. SP가 공격당한다면 네트워크에 참여하는 객체에 대한 민감한 정보들이 모두 노출되게 되고 SP를 사용할 수 없게 된다면 어떤 객체도 보안 연결을 할 수 없게 된다. 본 논문에서는 이러한 문제를 해결하기 위해서 SP는 클라이언트에 대한 접근 제어 정보와 같은 최소한의 정보만 가지고 있고 키 정보 및 세션 정보 같은 보안에 민감한 정보가 노출되거나 직접적으로 DTLS 핸드쉐이크에 참여하지 않는다.

이외에도, DTLS 패킷 크기를 줄여서 통신비용을 낮추기 위한 몇 가지 방법들이 제안되었다. DTLS 헤더 압축[11], IP 및 UDP 헤더 압축[12]과 같은 방법들이 제안되었다.

III. 제안 프로토콜

3.1 설계원리

본 제안에서의 네트워크 모델은 Fig.1.에서와 같이 다수의 센서 디바이스들 (CoAP 서버 $S_1, S_2, \ldots$)로 구성된 LoWPAN(low-power wireless personal area network)이 6LBR을 통해 IPv6 인터넷에 연결된다. 센서 디바이스들의 집합(S)은 여러 서브 집합($Sub_j,\ j=1,2,\ldots,k$)으로 그룹화 되고, 특정 센서 디바이스는 하나의 서브 집합에만 속하게 된다. 특정 서브 집합에 속하는 센서 디바이스들은 동일한 그룹키 GK_j를 공유하고 있으며 SP는 LoWPAN 내의 센서 디바이스(S_i)를 관리하고 센서 디바이스들과의 대칭키(K_i)를 공유하고 있다고 가정한다. 또한 센서 디바이스들은 CoAP 클라이언트와 ECDH 키 교환을 위해 ECQV 인증서를 발급받게 되는데, SP는 ECQV 인증서를 발행하는 CA의 역할을 수행한다.

Table 1. Table of notations

Notation	Description
N_i	nonce (random number)
AK_A, SK_A	authentication key and session key shared between C_A and S_x
$Cert_x$	ECQV certificate of S_x
$[.]GK$	symmetric encryption of [.] by a group key GK
$H(.)$	second-preimage resistant hash function
$kdf(.)$	key derivation function
$MIC(K)$	message-integrity code based on a symmetric key K covering all preceding messages exchanged

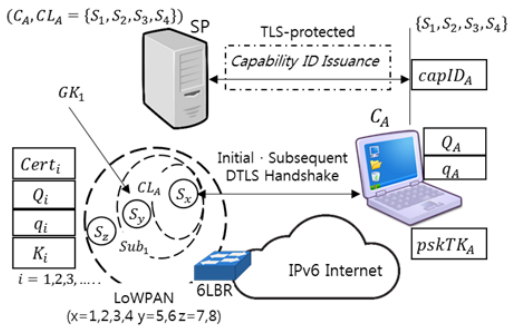

Fig. 1. Proposed network model and DTLS handshakes

타원 곡선 $E(F_p)$는 유한체 E_p(p는 소수)상에 존재하는 $y^2 \equiv x^3 + ax + b$를 만족하는 포인트들의 집합이다. 타원곡선 매개변수는 (p,a,b,G,n)으로 표기하며, G는 위수가 $n = |E(F_p)|$인 베이스 포인트이다. 센서 디바이스에 대한 ECQV 인증서 발급은 다음과 같이 진행된다. S_x는 SP로부터 인증서를 요청하기 위해 $r_x \in [1, n-1]$을 생성하여, $R_x = r_x \,.G$를 보낸다. SP도 $r_{sp} \in [1, n-1]$를 생성하여 S_x의 암시적 인증서 및 암시적 서명을 다음과 같이 생성하다.

$$Cert_x = R_x + r_{SP} \,.G = R_x + R_{SP} \tag{1}$$

$$s = q_{SP} + r_{SP} \,.H(Cert_x, S_x) \tag{2}$$

SP로부터 $Cert_x$와 s를 받게 되면, S_x는 자신의 ECDH 개인키 $q_x = s + r_x.H(Cert_x, S_x)$와 ECDH 공개키 $Q_x = q_x.G$를 도출한다. S_x의 공개키는 ECQV 인증서와 SP의 공개키 Q_{SP}만 주어진다면 다음과 같이 도출이 가능하다.

$$Q_x = q_x.G = Q_{SP} + Cert_x.H(Cert_x, S_x). \tag{3}$$

SP는 C_A에 대한 접근제어 서버 역할도 수행한다. 인가된 CoAP 클라이언트(C_A)는 접근이 허용되는 특정 서브 집합이 할당되고, 거기에 속하는 센서 디바이스들에 대한 접근만이 허용된다. 특히, CoAP 클라이언트의 역할에 따라서 할당된 서브 집합 내에서 접근이 허용되는 센서 디바이스들을 다음과 같이 추가적으로 제한 할 수도 있다. (Fig.1.에서는 $l = 4$)

$$CL_A = \{S_1, S_2, \ldots, S_l\} \subseteq Sub_1 \;,\; l \geq 1. \tag{4}$$

이때 CL_A는 클라이언트(C_A)의 Capability List로 정의되며, SP는 CL_A에 해당하는 $capID_A$ (Capability ID)를 발행해 준다. C_A는 자체적으로 ECDH 개인키 및 공개키 쌍 $(q_A, Q_A = q_A.G)$을 가지고 있다고 가정한다. C_A는 자신의 CL_A에 포함된 센서 디바이스와의 "Initial DTLS 핸드쉐이크"를 진행할 때 ECQV 인증서 및 인증된 RPK 모드를 사용한다. 해당 핸드쉐이크가 종료되면 센서 디바이스는 PSK토큰을 생성해서 C_A에게 발행하고 "Subsequent DTLS 핸드쉐이크"에서는 PSK 토큰을 사용해서 센서 디바이스와 C_A간의 핸드쉐이크 계산 복잡도를 줄일 수 있다.

3.2 Capability List에 따른 접근 제어

인가된 CoAP 클라이언트는 Capability List에 따라 해당 센서 디바이스에 접근할 수 있다. Capability List $\{S_1, S_2, \ldots, S_l\}$에 따른 접근제어를 내재시키기 위해 $Auth_A$가 루트 해시노드가 되는 Merkle 트리를 정의한다. Fig.2.에서와 같이 리프 노드들을 Capability List로 구성하여 포화이진트리가 정의된다($n \geq 0, l = 2^n$).

〔**알고리즘 1**〕 Capability List 구성

- 주어진 $S_1,S_2,...S_l$를 $h_{(\log_2 l)0}$, $h_{(\log_2 l)1},...,h_{(\log_2 l)(l-1)}$로 설정.
- $Auth_A \leftarrow h_{00} = H(h_{10},h_{11})$,

$x = 0,1,2,\ldots,(\log_2 l)-1$와 $y = 0,1,2,...,l-1$에 대해서 $h_{xy} = H(h_{(x+1)(2y)},h_{(x+1)(2y+1)})$를 계산.

리프 노드에서 루트 해시 노드까지의 경로에 있는 노드의 하위 해시 노드들을 "*Subsidiary* 노드"로 정의한다 (Fig.2.에서 h_{20}의 *Subsidiary*는 h_{21}과 h_{11}이 된다). $Auth_A$는 리프 노드와 *Subsidiary* 노드를 통해 계산이 가능하다 (h_{20}이 주어지면 모든 리프 노드에서부터 시작해서 계산하지 않고 h_{21}과 h_{11}를 통해서 계산 할 수 있다). 이를 위해 다음의 함수를 정의한다. 리프 노드 S_x와 그에 따른 $Subsidiary_x$로부터 $Auth_A = Auth_A_Comp(S_x,Subsidiary_x)$를 통해서 루트 해시 노드가 계산 된다 ($x \in [1,l]$). $Auth_A$는 C_A가 CL_A에 있는 센서 디바이스에 접근할 수 있는 $capID_A$를 구성하는데 사용된다.

〔**프로토콜 2**〕는 SP가 $capID_A$를 생성해서 C_A에게 전달해 주는 Capability ID 발급 프로토콜이다. 해당 프로토콜은 TLS로 보호되어진다고 가정하지만 C_A와 SP간의 보안연결이 어떻게 설정되는지는 본 논문의 범위를 벗어난다.

〔**프로토콜 2**〕 Capability ID 발급

- $SP \leftarrow C_A : C_A,\ S_x,\ Q_A$
- $SP \rightarrow C_A : capID_A,\ CL_A,\ Q_{SP}$

C_A는 ECDH 공개키/개인키 쌍 (Q_A, q_A)을 생성하고 SP에게 (C_A,S_x,Q_A)를 전송하여 Capability ID 발급을 요청한다. S_x는 C_A가 최초로 접근하고자 하는 센서 디바이스이다. SP는 C_A에게 허용되는 센서 디바이스들의 목록인 Capability List를 결정하고, S_x가 CL_A에 속하는 지를 검사한 이후에 $capID_A = H(K_x,Q_A,Auth_A)$를 생성해서 C_A에게 전송한다. K_x는 SP와 S_x사이에 미리 공유된 대칭키 이다.

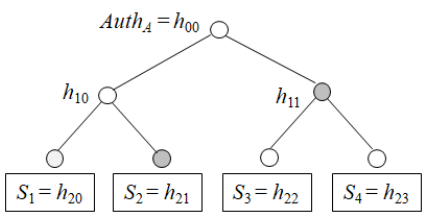

Fig. 2. Merkle tree with CL_A

3.3 제안 프로토콜

C_A가 $S_x (\in CL_A)$에 접근한다고 가정하고, 양자간에는 "Initial DTLS 핸드쉐이크" 프로토콜이 기동된다.

〔**프로토콜 3**〕 Initial DTLS 핸드쉐이크

- $S_x \leftarrow C_A : N_A$
- $S_x \rightarrow C_A : N_x,\ Cert_x$
- $S_x \leftarrow C_A : Q_A,\ capID_A,\ Subsidiary_x,\ MIC(SK_A)$
- $S_x \rightarrow C_A : pskTK_A,\ MIC(SK_A)$

C_A가 생성한 난수 N_A를 받은 S_x는 N_A와 동일한 방법으로 자체 생성한 N_x와 자신의 ECQV 인증서 $Cert_x$를 C_A에게 전송한다. C_A는 S_x의 ECDH 공개키를 다음과 같이 구한 이후에 인증키와 세션키를 도출한다.

$$Q_x (= q_x \cdot G) = Q_{sp} + Cert_x \cdot H(Cert_x,S_x) \quad (5)$$

$$AK_A (= q_A \cdot Q_x),\ SK_A = kdf(AK_A,N_x,N_A) \quad (6)$$

S_x에게 C_A의 공개키 Q_A와 $capID_A, Subsidiary_x$를 보내면서 $MIC(SK_A)$을 통해 전송 메시지의 무결성을 유지한다. S_x도 동일하게 $AK_A (= q_x \cdot Q_A)$와 $SK_A = kdf(AK_A,N_x,N_A)$를 계산, $MIC(SK_A)$의 유효성을 확인한다.

$Auth_A = Auth_A_Comp(S_x,Subsidiary_x)$의 도출을 통해 $capID_A = H(K_x,Q_A,Auth_A)$를 계산하고 C_A에게 받은 $capID_A$와 동일한지 확인한다. 동일하다면 C_A가 S_x에 접근할 수 있는 권한이 있다는 것이 확인된다. 마지막으로 "Subsequent DTLS 핸드쉐이크"에서 사용할 $pskTK_A = [C_A,AK_A,Auth_A]GK_1$를 C_A에게 전달한다.

〔프로토콜 4〕 Subsequent DTLS 핸드쉐이크

- $S_{x'} \leftarrow C_A : N_A$
- $S_{x'} \rightarrow C_A : N_{x'}$
- $S_{x'} \leftarrow C_A : pskTK_A,\ Subsidiary_{x'},\ MIC(SK_A)$
- $S_{x'} \rightarrow C_A : MIC(SK_A)$

"Initial DTLS 핸드쉐이크"를 마친 C_A가 CL_A에 속해있는 다른 센서 디바이스 $S_{x'}(\in CL_A)$에 접근할 경우에는 "Subsequent DTLS 핸드쉐이크"가 진행된다. 상호 교환된 난수 N_A와 $N_{x'}$ 그리고 인증키 AK_A를 기반으로 $SK_A = kdf(AK_A, N_{x'}, N_A)$를 도출하고 $pskTK_A$와 $Subsidiary_{x'}$를 $S_{x'}$에게 전달한다. $S_{x'}$는 S_x와 동일한 서브 집합 Sub_1에 속해 있기 때문에 GK_1을 이미 알고 있다. 따라서 $S_{x'}$는 $Auth_A$를 계산하고 C_A가 전송했던 $pskTK_A$의 $Auth_A$와 동일한지 확인하며, 동일하다면 C_A는 $S_{x'}$에 접근 할 수 있는 권한이 있다는 것이 입증된다. $S_{x'}$ 역시 $SK_A = kdf(AK_A, N_{x'}, N_A)$를 계산하고 마지막 메시지의 전송을 통해 자신에 대한 인증을 수행하게 된다.

IV. 안전성 분석 및 성능 평가

4.1 Capability ID 위조 가능성

센서 디바이스들은 각각 다른 서브 집합 (Sub_j, $j = 1, 2, ..., k$) 중 하나의 집합에만 속하기 때문에 CoAP 클라이언트는 정해진 Capability List에 따라서 하나의 서브 집합에만 속하는 CoAP 서버에 접근 할 수 있게 된다. 특히, C_A의 $capID_A$가 SP로부터 발행될 때, $H(K_x, Q_A, Auth_A)$를 통해서 생성되기 때문에 K_x의 안전한 유지관리는 필수적이다.

본 논문에서 제안하는 DTLS 핸드쉐이크의 안전성은 Capability ID를 위조할 수 없다는 사실에 기반을 두고 있다. 공격자(Adversary)가 $capID_A$와 동일한 Capability List에 접근이 가능한 $capID_{Adv}$를 생성하기 위해서는 $H(K_x, Q_{Adv}, Auth_A)$를 통해서 생성이 되어야 한다. $X = K_x \| Q_A \| Auth_A$이고 $X' = K_x \| Q_{Adv} \| Auth_A$일 때, 공격자는 $X = X'$를 만족하는 ECDH 개인키/공개키 (q_{Adv}, Q_{Adv}) 쌍을 찾아내야 하는데, $H(.)$는 $H(X) = H(X')$를 만족하는 X'를 찾을 수 없는 제2역상 저항성을 가지는 함수이기 때문에 불가능하다. 결국 Capability ID를 위조하는 것 역시 불가능하게 된다.

4.2 센서 디바이스에 대한 Security Bootstrapping

제안된 프로토콜에서는 센서 디바이스와 SP사이에 대칭키(K_i)가 설정된다고 가정하였다. 이 가정의 현실성은 다음과 같이 설명 가능하다. 각 센서 디바이스들은 LoWPAN에 배치되면서 IPv6 주소 설정을 위해서 6LBR과 6LoWPAN ND(Neighbor Discovery) 프로토콜 그리고 IEEE 802.15.4의 Pan Coordinator와 Join 프로토콜을 수행하게 된다. 센서 디바이스가 LoWPAN에 안전하게 배치되도록 하기 위해서는 결국 센서 디바이스와 PC 또는 6LBR 사이에 초기 보안설정을 위한 Security Booststrapping 과정이 필요한데 이 과정을 통해서 센서 디바이스와 SP간의 대칭키 설정이 가능하게 된다.

4.3 성능 평가 및 비교

〔13, 14〕에서 정의된 DTLS 메시지를 기반으로 제안된 보안 프로토콜을 위한 DTLS 메시지의 흐름은 Fig.3.과 같다. 총 6개의 메시지 Flight들로 구성되며 Certificate 메시지 필드를 수정하여 S_1 (x=1)의 인증서 $Cert_1$과 $capID_A$가 탑재되며 ClientKey Exchange 메시지 필드에 $Subsidiary_1$을 탑재한다. 이 후에 동일 서브 집합의 S_2와 Subsequent DTLS 핸드쉐이크를 수행 할 때에는 $Subsidiary_2$와 $pskTK_A$가 탑재된다.

실험 환경은 Contiki 3.0을 사용하였고 센서 디바이스는 TI CC2538em 〔15〕으로 ARM Cortex-M3 Processor (32 MHz)와 512KB Flash, 32KB RAM 그리고 4KM ROM을 지원한다. 6LBR은 Raspberry Pie 2에 CETIC 6LBR 〔16〕을 포팅해서 구성하였고 TinyDTLS 〔17〕를 CC2538em과 Ubuntu PC에 포팅하였다. CC2538em에서는 난수를 생성할 때 16-bit의 LFSR(Linear Feedback Shift Register)를 사용하는 PRNG(Pseudo Random Number Generator)를 제공한다. 타원곡선 매개변수는 secp256r1 〔18〕로 설정하고, 암호 해시함수 $H(.)$

는 SHA-256, 메시지 무결성 코드 $MIC(.)$은 AES-CBC-MAC-128 그리고, 키 유도 함수 $kdf(.)$는 HMAC-SHA-256을 사용한다. Table 2.는 다양한 보안모드에서 DTLS 핸드쉐이크에 사용된 cryptographic primitive들의 평균 실행시간을 보여준다. $ECDSA$, $EnC, H(.)$, $kdf(.)$ 및 $MIC(.)$ 실행에 사용된 메시지 길이는 32바이트, 키 길이는 16바이트를 기준으로 하였다. ECDH는 단일 스칼라 곱셈을 이용한 실행시간, $ECDSA_G$는 ECDSA 생성시간, $ECDSA_V$는 ECDSA 검증시간 그리고 EnC는 대칭키 암호화 계산시간이다.

Fig.4.에서는 제안된 프로토콜과 함께 PSK, RPK(ECDSA),인증서 (ECDSA 탑재 X.509 인증서 및 ECQV 인증서)모드로 DTLS 핸드쉐이크 지연시간을 측정했다. Fig.3.에서의 각 단계(①, ②, ③)마다 클라이언트로부터 패킷을 수신, 처리하고 다시 클라이언트에게 패킷을 전송하는 실행시간을 서버 측에서 측정되었다(각 단계는 보안협상, 키교환 및 키확인 단계를 의미). Proposed 1, 2는 각각 Initial 및 Subsequent DTLS 핸드쉐이크를 나타낸다. PSK, ECQV, Proposed 1 및 Proposed 2는 ①과 ②에서 큰 차이가 없다. RPK와 X.509의 경우, ② 단계에서의 실행시간은 ECDSA 서명생성 작업으로 인해서 2,282 ms와 2.366 ms 로 측정되며 ③ 단계에서는 각각 3.391 ms 와 3.987 ms 으로 측정된다. ECQV 인증서 방식에서는 2개의 EC 스칼라 곱셈을 서버와 클라이언트가 각각 수행하지만 제안된 프로토콜에서는 클라이언트에서 1번만 수행되어 소요되는 시간을 단축 시켰다. 그에 따라서 자원 제약적인 센서 디바이스가 가지는 부담도 줄일 수 있기 때문에 ECQV 인증서 방식 보다 Proposed 1이 더 효율적이다. ③에서 Proposed 2는 PSK와 견줄만한 실행시간을 보여주고 있다.

Fig.5.에서는 6LoWPAN 네트워크상 (1-hop 및 2-hop 환경)에서 각각의 Security Mode에 대한 DTLS 핸드쉐이크 왕복지연(Round Trip Delay)을 보여주고 있다. X.509 및 ECQV의 경우 CA와의 인증서 유효성 검사에 대한 지연은 포함되어 있지 않다.

Table 2. Primitive execution time

Primitive	Parameter	Execution Time
$ECDH$	secp256r1	1100.83ms
$ECDSA_G$	secp256r1	1150.97ms
$ECDSA_V$	secp256r1	1249.09ms
EnC	AES-CTR-128	0.38ms
$H(.)$	SHA-256	0.39ms
$kdf(.)$	AES-CBC-MAC-128	0.76ms
$MIC(.)$	HMAC-SHA-256	6.10ms
$Nonce$	16-bit LFSR with radio ADC	0.06ms

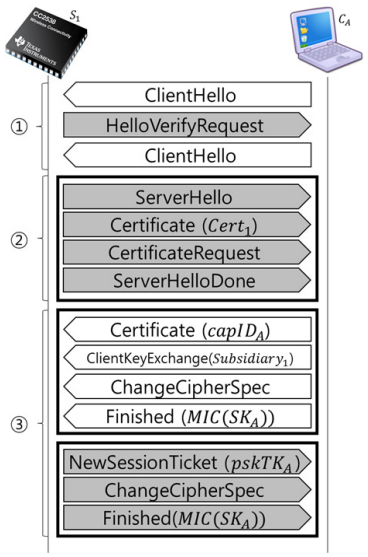

Fig. 3. Initial DTLS handshake message

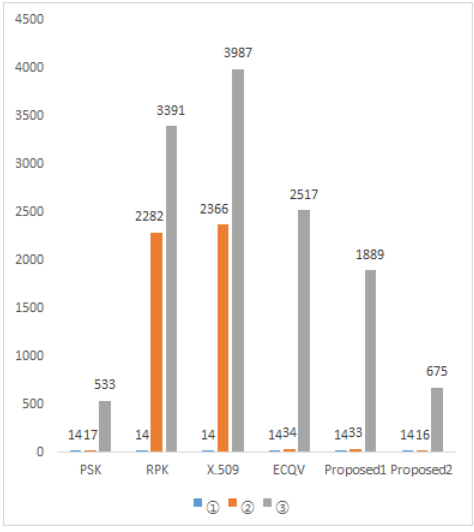

Fig. 4. Comparison of the execution times for each phase(ms)

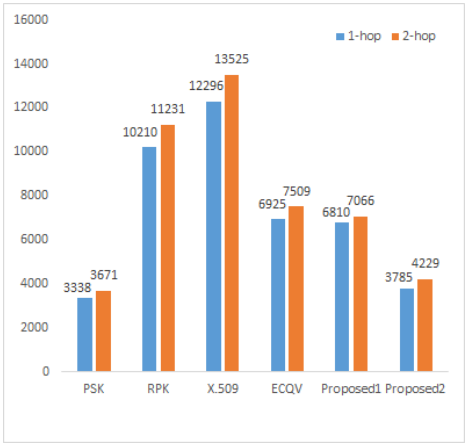

Fig. 5. Round trip delay(ms)

V. 결 론

본 논문에서는 안전한 CoAP 응용을 위한 DTLS 기반의 그룹 지향적 종단 간 보안 메커니즘을 제안하였다. CoAP 서버와 클라이언트 사이의 종단 간 보안 연결의 과정에서 DTLS 핸드쉐이크에 의한 계산량을 줄이기 위해 "Initial DTLS 핸드쉐이크"를 마치면서 PSK 토큰을 발급하고 "Subsequent DTLS 핸드쉐이크"에서는 PSK 모드를 사용하기 때문에 CoAP 서버 그룹과의 DTLS 핸드쉐이크에 의해 유발되는 계산량을 감소시켰다. 그에 따른 센서 디바이스의 에너지 소비량도 감소하기 때문에 배터리의 수명을 연장시킬 수 있다. Capability ID라는 개념을 기반으로 사전에 CoAP 서버를 그룹화하여 세밀한 접근 제어가 가능하게 하였다. 따라서 다양하고 안전한 IoT 애플리케이션이 그룹 지향적 종단 간 보안의 장점을 확보할 수 있게 된다. 전통적인 WSN에서 프록시를 통한 센서 디바이스에의 간접 접근방식에서는 서버와 클라이언트만 키 정보를 공유하는 것과 같은 완벽한 종단 간 보안을 충족 할 수 없었지만 제안된 프로토콜을 통해서 프록시는 센서 디바이스에 대한 접근제어 정보만을 알고 있고 키 정보 같이 민감한 정보는 오로지 서버와 클라이언트 사이에만 공유되는 종단 간 보안을 충족하게 된다.

References

[1] IEEE std. 802.15.4-2011, Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (LR-WPANs), Standard for Information Technology Std. June 2011.

[2] E. Rescorla and N. Modadugu, "Datagram Transport Layer Security," IETF RFC 6347, Jan. 2012.

[3] Z. Shelby, K. Hartke, and C. Bormann, "The Constrained Application Protocol (CoAP)," IETF RFC 7252, Jun. 2014.

[4] D. R. L. Brown, R. Gallant, and S. A. Vanstone, "Provably Secure Implicit Certificate Schemes," Financial Cryptography, LNCS 2339, Springer-Verlag, pp. 156-165, Feb. 2001.

[5] R. Hummen, J. Ziegeldorf, H. Shafagh, S. Raza, and K. Wehrle, "Towards Viable Certificate-based Authentication for the Internet of Things," in Proc. of the 2nd ACM Workshop on Hot Topics on Wireless Security and Privacy, pp. 37-42, Apr. 2013.

[6] N. Kang, J. Park, H. Kwon and S. Jung, "ESSE: Efficient Secure Session Establishment for Internet-Integrated Wireless Sensor Networks," International Journal of Distributed Sensor Networks, vol. Jan. 2015, Article ID 393754.

[7] S. Raza, L. Seitz, D. Sitenkov, and G. Selander, "S3K: Scalable Security With Symmetric Keys—DTLS Key Establishment for the Internet of Things," IEEE Transactions on Automation Science and Engineering, vol. 13, no. 3, pp. 1270-1280, July. 2016.

[8] T. Kothmayr, C. Schmitt, W. Hu, M. Brünig, and G. Carle, "DTLS based Security and Two-way Authentication for the Internet of Things," Ad Hoc Networks, vol. 11, Issue 8, pp. 2710-2723, Nov. 2013.

[9] Y. B. Saied, A. Olivereau, D. Zeghlache, and M. Laurent, "Lightweight Collaborative Key Establishment Scheme for the

Internet of Things," Computer Networks, vol. 64, pp. 273-295, May. 2014.
〔10〕 J. Granjal, E. Monteiro, and J. Sa Silva, "End-to-End Transport-Layer Security for Internet-Integrated Sensing Applications with Mutual and Delegated ECC Public-Key Authentication," in Proc. of IFIP Networking Conference and Workshop, pp. 1 - 9, New York, U.S.A, May 17-19, 2013.
〔11〕 S. Raza, H. Shafagh, K. Hewage, R. Hummen, and T. Voigt, "Lithe: Lightweight Secure CoAP for the Internet of Things," IEEE Sensors Journal, vol. 13, no. 10, pp. 3711-3720, Oct. 2013.
〔12〕 J. Hui and P. Thubert, "Compression Format for IPv6 Datagrams over IEEE 802.15.4-based Networks," IETF RFC 6282, Sep. 2011.
〔13〕 J. Salowey, H. Zhou, P. Eronen, and H. Tschofenig, "Transport Layer Security (TLS) Session Resumption without Server-Side State," IETF RFC 4507, Jan. 2008.
〔14〕 P. Wouters, H. Tschofenig, J. Gilmore, S. Weiler, and T. Kivinen, "Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS)," IETF RFC 7250, Jun. 2014.
〔15〕 Texas Instrument, CC2538 Powerful Wireless Microcontroller System-On-Chip Datasheet, Apr. 2015.
〔16〕 CETIC 6lbr, https://github.com/cetic/6lbr/wiki
〔17〕 TinyDTLS, https://projects.eclipse.org/projects /iot.tinydtls
〔18〕 Certicom Research, "SEC 2: Recommended Elliptic Curve Domain Parameters," Standards for Efficient Cryptography, ver. 2.0, Jan. 27 2010.

〈저자소개〉

정 연 성 (Yeon-seong Jeong) 학생회원
2014년 2월: 백석문화대학교 인터넷보안 졸업
2016년 2월: 공주대학교 컴퓨터공학부 졸업
2016년 3월~현재: 단국대학교 소프트웨어 보안 석사과정
〈관심분야〉 정보보호, 네트워크 보안

박 창 섭 (Chang-seop Park) 종신회원
1983년 2월: 연세대학교 경제학과 졸업
1987년 2월: Lehigh University 컴퓨터과학과 석사
1990년 2월: Lehigh University 컴퓨터과학과 박사
1990년 3월~현재: 단국대학교 소프트웨어학과 교수
〈관심분야〉 정보보호, 네트워크 보안, 무선인터넷 및 모바일 컴퓨팅 보안