情報保護學會論文誌
第 22 卷 第 4 號, 2012. 8

ZigBee 무선 센서 네트워크에서의 안전한 키 분배 프로토콜*

오 수 민,† 최 수 경, 권 예 진, 박 창 섭‡
단국대학교

Secure Key Distribution Protocol for ZigBee Wireless Sensor Network *

Su-min Oh,† Soo-kyeong Choi, Ye-jin Kwon, Chang-seop Park‡
Dankook University

요 약

본 논문에서는 ZigBee-2007 표준 명세서에 포함된 키 분배 프로토콜의 문제점을 지적하고 이에 대한 개선안을 제시한 Yuksel-Nielson 기법이 키 비동기화 공격에 취약함을 보인다. 또한, Yuksel-Nielson 기법을 개선한 새로운 키 분배 프로토콜을 제안하고 이에 대한 안전성 및 성능분석을 통해서 본 논문에서 제안하는 키 분배 프로토콜의 유효성을 검증한다.

ABSTRACT

It is shown in this paper that Yuksel-Nielson's key distribution scheme is not secure against key de-synchronization attack even though their scheme supplement ZigBee-2007 specification's security problems. Furthermore, a new key distribution scheme is proposed, which is the one to fix the security weakness of Yuksel-Nielson's scheme, as well as its security and performance analysis to verify its effectiveness.

Keywords: ZigBee, Key Distribution Protocol, Security, Performance analysis

I. 서 론

저 전력, 저 비용 무선 네트워크 프로토콜의 가장 대표적인 표준인 ZigBee는 헬스케어, 스마트그리드, 환경 및 보안 분야에서 그 활용성이 높아지고 있다. ZigBee [1]와 IEEE 802.15.4 [2]는 다양한 유형의 무선 센서 네트워크 활용을 위한 네트워크 인프라를 제공하는 표준 프로토콜이다. IEEE 802.15.4는 물리적 계층과 MAC 계층을 정의하고, ZigBee에서는 네트워크 및 응용계층을 정의하고 있다. ZigBee의 첫번째 스택 버전인 ZigBee-2004가 발표된 이후, ZigBee-2006을 거쳐 현재는 ZigBee-2007이 최종 발표된 상태이다. ZigBee 네트워크 계층은 Star 형태의 네트워크와 Tree 형태의 네트워크, 그리고 일반적인 Mesh 네트워크 구성을 지원한다. 모든 네트워크는 콘트럴 타워의 역할을 하는 ZigBee Coordinator(ZC) 장치를 통해서 유지, 관리된다. ZigBee 노드(Node)는 소형의 마이크로 컨트롤러 기반에서 실행되므로 메모리와 전력 사용에 있어 상당히 제한적이다. 따라서 ZigBee 노드들은 전력 소모를 줄이기 위하여 대부분 대기상태 (Sleep Mode)에 있으면서, 주기적으로 깨어나 다른 노드와의 작업을 수행한다.

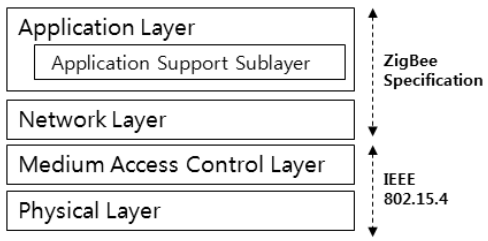
접수일(2012년 4월 16일), 수정일(2012년 7월 6일), 게재확정일(2012년 7월 30일)
* 이 논문은 2012년도 정부(교육과학기술부)의 재원으로 한국연구재단의 기초연구사업 지원을 받아 수행된 것임 (2012R1A1A2000677)
† 주저자, kstori3924@gmail.com
‡ 교신저자, csp0@dankook.ac.kr

일반적인 무선 네트워크 환경은 유선환경보다 보안 공격에 취약함은 이미 널리 알려진 사실이다. ZigBee의 경우도 보안공격에 대응하기 위한 보안기능들을 기본적으로 탑재하고 있다. IEEE 802.15.4 표준에서 정의된 보안기능을 사용하고 있으며, 추가적으로 네트워크 및 응용계층을 지원하기 위한 보안기능들이 포함되어 있다. 특히, IEEE 802.15.4에서 결여되어 있는 키 분배 및 갱신 등의 키 관리기능을 통해서 계층적인 보안 관리의 틀을 마련하고 있다. ZigBee 노드는 보안이 설정되는 계층에서 보안모드(Security Mode)를 선택하여 보안수준(Security Level)을 달리 설정할 수 있으며, 어떤 보안수준을 정하느냐에 따라 보안 서비스 및 복잡도 또한 달라진다. ZigBee 프로토콜에서는 각각의 노드들 간에 메시지 전달 시 보안을 위해 키를 사용하도록 표준 문서에 정의되어 있으며, 이러한 키 관리나 저장을 위한 오버헤드는 ZigBee 노드의 능력에 따라 적절히 사용되어야 한다.

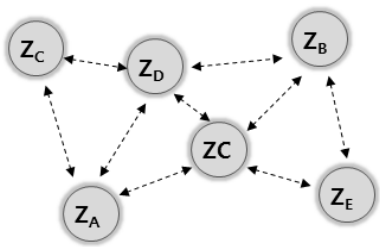
ZigBee 프로토콜의 일반적인 보안과 관련된 논문[7, 8, 9]들이 부분적으로 발표되어 있으나 본 논문에서는 ZigBee-2007 표준 명세서에 나와 있는 키 분배 프로토콜에 초점을 맞춘다. 최근, Yuksel-Nielson [3]은 ZigBee-2007에 명시된 키 분배 프로토콜의 문제점을 지적하고 수정된 키 분배 프로토콜을 제안한 바 있다. 하지만, 이들의 제안 역시 보안상의 문제점이 존재하고 있고, 따라서 이를 개선한 새로운 키 분배 프로토콜을 본 논문에서 제안한다. 2장에서는 ZigBee에서 제공되는 기본적인 보안 아키텍처와 키 분배 프로토콜을 소개하고, 3장에서는 Yuksel-Nielson이 지적한 ZigBee 키 분배 프로토콜의 문제점 및 이들이 제시한 수정된 키 분배 프로토콜을 소개한다. 4장에서는 Yuksel-Nielson 키 분배 프로토콜이 "키 비동기화 공격"에 취약함을 보이고, 새롭게 개선된 키 분배 프로토콜을 제안한다. 5장에서는 제안된 키 분배 프로토콜에 대한 안전성 분석 및 에너지 소모율 기반의 효율성을 분석하고 6장에서 결론을 맺는다.

II. ZigBee 보안 아키텍처

[그림 1]에서와 같이 IEEE 802.15.4 기반의 ZigBee 프로토콜에서는 네트워크 계층(Network Layer) 그리고 응용 계층(Application Layer) 내의 응용지원 부계층(Application Support Sublayer)에 포함된 보안 메커니즘을 통해서 ZigBee 프레임에 대한 기밀성 및 무결성 서비스가 제공된다. ZigBee 네트워크를 구성하는 노드들의 유형으로는 ZigBee Coordinator(ZC), ZigBee Router 및 ZigBee End Device가 있다. 하지만, 본 논문에서는 [그림 2]에서와 같이 ZigBee Router와 ZigBee End Device는 ZC와 구별되는 ZigBee 디바이스로 통칭하며 ZX (X= A, B, C, ...)로 표기한다.



[그림 1] ZigBee 프로토콜 스택



[그림 2] ZigBee 네트워크

ZigBee 프로토콜에서의 기밀성 및 무결성 서비스는 128 비트 블록암호 AES를 기반으로 한 CCM* 암호 알고리즘을 통해서 제공된다. CCM*는 초기 설정된 보안수준에 따라서 기밀성 및 무결성 서비스를 통합 또는 선별적으로 적용하는 것을 가능하게 한다.

2.1 Trust Center 및 암호키 유형

ZigBee Coordinator(ZC) 내에 설치되는 Trust Center(TC)는 ZigBee 네트워크에 속해 있는 모든 ZigBee 노드들이 다양한 보안 프로토콜을 수행하는 데에 있어서 가장 중추적인 역할을 담당한다. TC는 다양한 암호키를 생성하여 ZigBee 노드들에게 분배해 주며, 주기적으로 또는 ZigBee 노드들의 요청에 따라서 암호키를 갱신해 주는 작업을 수행한다. 또한, 새로운 ZigBee 노드가 네트워크에 가입하거나 탈퇴

하는 과정에도 관여한다.

ZigBee 보안 메커니즘에서 사용되는 암호키는 "마스터 키(Master Key)", "링크키(Link Key)" 및 " 네트워크 키(Network Key)"가 있다. "네트워크 키"는 ZigBee 네트워크에 속해 있는 모든 노드들이 TC와 더불어 공유하는 일종의 그룹키(Group Key)로써 네트워크 계층에서의 ZigBee 프레임을 보호하기 위해 사용된다.

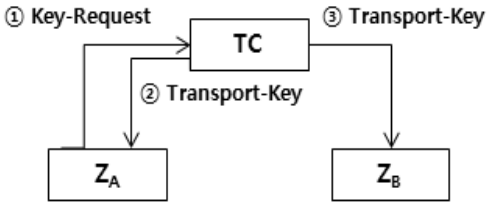
"링크 키"는 "마스터 키"를 기반으로 도출된다. "링크키"는 응용 계층에서의 ZigBee 프레임을 보호하기 위해 사용된다. "마스터 키"와 "링크키"는 TC와 특정 ZigBee 노드 간에 공유되는 "TC-마스터 키" (TC-MK)와 "TC-링크키" (TC-LK), 그리고 임의의 두 ZigBee 노드 간에 공유되는 "AP-마스터 키" (AP-MK)와 "AP-링크키" (AP-LK)로 구분된다. "네트워크 키"와 "TC-마스터 키" 그리고 "TC-링크키"는 ZigBee 노드의 초기화 과정에서 별도의 채널을 통해 설치될 수도 있고, ZigBee 네트워크 가입 후에 TC를 통해 전송될 수 있다. 그러나 후자의 경우는 제3자에 의한 도청의 위협이 존재한다.

2.2 ZigBee 보안모드와 키 분배 프로토콜

ZigBee의 보안모드(Security Mode)에는 Standard Security 모드와 High Security 모드가 있다. Standard Security 모드는 가정 내의 어플리케이션을 위한 낮은 수준의 보안을 위해 설계되었으며 기본적으로 "네트워크 키"만을 사용한다. 반면에 High Security 모드는 상업적인 용도의 어플리케이션 등을 위한 높은 수준의 보안을 위해 설계되었으며 TC에는 ZigBee 노드 리스트, "TC-마스터 키", "TC-링크키", "네트워크 키"가 유지, 관리된다.

ZigBee 네트워크에 가입된 임의의 두 ZigBee 노드 간에 공유되는 "AP-마스터 키"와 "AP-링크키"는 "ZigBee 키 분배 프로토콜"을 통해서 공유된다. 특히, "AP-마스터 키"가 임의의 두 노드 간에 분배되어지는 경우에는 이를 기반으로 "AP-링크키"를 도출하는 SKKE (Symmetric-Key Key Exchange) 프로토콜이 기동된다.

〔그림 3〕의 ZigBee 키 분배 프로토콜은 ZA와 ZB 간에 공유될 "AP-링크키 "LKAB"를 TC에게 요청하는 경우에 사용된다. (프로토콜 기호의 의미는 표 1 참조) 여기서는 ZA와 TC 그리고 ZB와 TC간에는 각각 "TC-링크키" LKA와 LKB가 사전에 공유되어



〔그림 3〕 ZigBee 키 분배 프로토콜

있음을 가정한다. ZA는 TC에게 Key-Request 메시지를 전송한다. 이 메시지는 TC에게 ZB와 공유할 "AP-링크 키(AP-LK)"의 분배를 요청한다. 이 메시지를 받은 TC는 자신이 생성한 "AP-링크키"인 LKAB를 각각 LKA와 LKB로 암호화해서 Transport-Key 메시지를 통해서 ZA와 ZB에게 전송한다. Key-Request 및 Transport-Key 메시지의 무결성 역시 LKA와 LKB를 기반으로 한 MIC(Message Integrity Check)에 의해서 보장된다.

① Key-Request (ZA, TC) :
{ZB, FCA, MIC(LKA)}
② Transport-Key (TC, ZA) :
{ZB, FCTC, 〔LKAB〕LKA, MIC(LKA)}
③ Transport-Key (TC, ZB) :
{ZA, FCTC, 〔LKAB〕LKB, MIC(LKB)}

III. Yulsel-Nielson의 키 분배 프로토콜

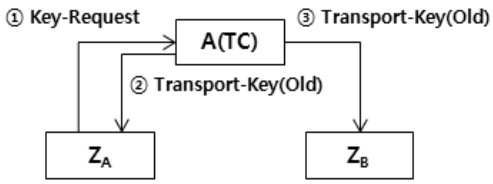
ZigBee 키 분배 프로토콜은 키 분배 센터의 역할을 하는 TC를 기반으로 임의의 두 ZigBee 노드 사이에 안전하게 키를 공유하기 위한 절차이다. 이러한 키의 관리와 분배를 표현하기 위한 프로토콜 기호를 상세히 서술하면 다음 〔표 1〕과 같다. TC 링크키가 노출되었을 경우, 공격자는 TC 링크키를 이용하여 각 노드 간에 전송되는 모든 메시지에 대한 도청이 가능해진다. 그러나 본 논문에서는 공격자가 메시지를 도청한 후 이루어지는 공격에 대한 안전성을 검증하기 위하여, TC와 노드 간의 "TC-링크키"는 안전하게 보관되고, 키의 누출은 없다고 가정한다. 또한 모든 프로토콜 메시지는 ZigBee 노드 간에 공유 되는 "TC-링크키"로 보호되어진다.

3.1 기존 ZigBee 키 분배 프로토콜의 문제점

ZigBee 프로토콜은 메시지의 중복이나 재생 공격

과 같은 문제를 해결하기 위해 전송되는모든 메시지에 Frame Counter(FC)를 포함시킨다. FC는 메시지의 Freshness를 보장하는 32 비트의 순번(sequence number)을 지칭한다. 하지만, FC는 필드길이가 짧을 경우 보안적으로 취약할 수 있다. FC는 메시지를 수신하는 측이 유지하고 있는 카운터 값 보다 작거나 같은 값은 경우에는 해당 메시지는 거부되고 현재 카운터 보다 큰 값만을 허용한다. 이는 만약 공격자가 FC를 최대값 (i.e. 0xFFFFFFFF)으로 변형시킨다면, 더 이상의 프레임은 모두 거부 된다 [4]. 또한, FC의 사용은 메시지의 중복을 제어할 수는 있지만, 필드길이가 짧을 경우에는 오버플로우(overflow)로 인한 초기화의 가능성이 높아진다. 더욱이 카운터는 계층형 아키텍처에서 하위 계층(ZigBee MAC 계층과 Network 계층) 또한 유사한 카운터를 사용하기 때문에 중복된 접근방법을 사용하게 된다.

따라서 ZigBee 키 분배 프로토콜에서는 TC와 각 노드간의 TC 링크키를 오랜 기간 동안 사용하기 때문에 FC를 사용하더라도 재생공격에 취약할 수 있다. 공격자가 기존의 메시지를 도청해 두었다가 FC가 초기화된 후 재생공격을 시도한다면 도청해 둔 메시지의 FC가 현재의 FC 보다 클 수 있기 때문에 재생공격이 성공할 가능성이 크다. 이는 FC의 크기가 32 비트라는 비교적 작은 수이기 때문이기도 하다. 이러한 문제점을 기반으로 Yuksel과 Nielson [3]은 기존 ZigBee 키 분배 프로토콜의 취약점을 지적하였다. 그들이 제시한 재생공격 시나리오는 [그림 4]와 같다.



[그림 4] Yuksel-Nielson 재생공격 시나리오

TC를 가장한 공격자 A(TC)가 ZA로부터 키 요청을 받게 되면, 노드 ZA와 ZB에게 이전에 사용되었던 키를 분배하는 공격이다. 처음 ZA가 TC에게 키를 요청하는 Key-Request 메시지는 다음과 같다.

① Key-Request (ZA, TC) :
{ZB, FCA, MIC(LKA)}

즉, ZA는 ZB와 공유할 AP 링크키(AP-LK)를 TC에게 요청한다. 공격자 A(TC)는 TC에게 전송되는 메시지를 중간에 막고 이전에 전송되었던 Transport-Key 메시지를 재생하여 ZA와 ZB에게 그대로 보내준다.

② Transport-Key (A(TC), ZA) :
{ZB,FCTC,{LKAB(old)}LKA,MIC(LKA)}old

③ Transport-Key (A(TC), ZB) :
{ZA,FCTC,{LKAB(old)}LKB,MIC(LKB)}old

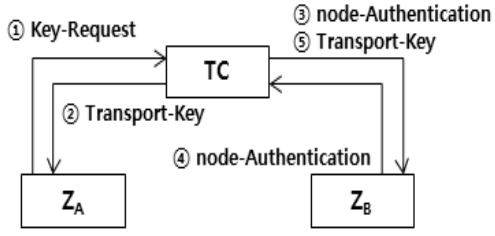
[표 1] 프토토콜 기호

기호	설명
Z_X	X의 identity (IEEE MAC 주소), X = A, B, TC, A(X)
R_X	X에 의해서 생성된 난수
LK_X	TC와 X간에 사전에 공유된 TC-LK
LK_{AB}	TC가 생성하여 배포하는 Z_A와 Z_B간의 공유 AP-LK
$\{m\}_K$	대칭키 K로 m을 암호화
$MIC(K)$	대칭키 K로 메시지 내의 모든 필드에 대한 메시지 인증 값
$\{m\}_{old}$	이전에 TC와 ZigBee 노드(Z_A 또는 Z_B)간에 전송된 메시지 m을 재생한 메시지
$A(X)$	X 노드를 가장한 공격자(Attacker)
FC_X	X 노드의 프레임 카운터(Frame Counter)
$h(.)$	one-way hash function
$msg\ (A\ ,B)$	A가 B에게 전송하는 프로토콜 메시지

ZA와 ZB는 TC와 공유하고 있는 TC-링크키를 이용하여 전송받은 메시지를 검증한다. 만약, ZA와 ZB가 유지하고 있는 FC가 이미 초기화 된 상태이고, 메시지에 포함된 FC 값이 유효하다면 결국 ZA와 ZB는 이전에 사용되었던 AP-LK인 LKAB(old)를 재사용하게 될 것이다. 따라서 만약 공격자가 이전에 사용했던 키를 알고 있다면 ZA와 ZB 사이의 메시지는 공격자에 의해 모두 노출될 수 있다.

3.2 Yuksel-Nielson의 수정된 키 분배 프로토콜

Yuksel-Nielson [3]이 제안한 키 분배 프로토콜에서는 인증 및 재생공격 방지를 위해서 순번 대신에 난수(Random number)를 사용한다. 기존 ZigBee 키 분배 프로토콜 [1]에서는 총 3개의 메시지를 교환하지만 [그림 5]에서와 같이 Yuksel-Nielson의 수정된 키 분배 프로토콜에서는 2개의 메시지가 추가되었다.



[그림 5] Yuksel-Nielson의 제안 프로토콜

①, ②, ⑤번의 메시지는 각각 기존의 ZigBee 키 분배 프로토콜에서 ①, ②, ③번의 메시지와 유사하다. 첫째, Key-Request 메시지를 통해서 ZA가 TC에게 ZB와의 암호통신을 위한 새로운 AP 링크키 분배를 요청한다. 이때 ZA에 의해 생성된 난수 RA는 challenge-response 방식을 통해서 TC를 인증하기 위한 목적으로 사용된다.

① Key-Request (ZA, TC) :
{ZB, RA, MIC(LKA)}

키 분배를 요청 받은 TC는 Transport-Key 메시지를 통해서 ZA에게 새로운 AP-링크키 LKAB를 전송한다. 메시지를 수신한 ZA는 TC와 사전 공유하고 있는 TC-링크키 LKA를 기반으로 TC에 대한 인증을 수행하게 된다.

② Transport-Key (TC, ZA) :
{ZB, RA, [LKAB]LKA, MIC(LKA)}

동시에 TC는 Transport-Key 메시지를 통해서 ZB에게도 동일한 AP-링크키 LKAB를 전송한다. 하지만, 그 이전에 TC와 ZB 간에는 새로이 정의된 node-Authentication 메시지를 통해서 난수 RB와 RTC를 이용한 challenge-response 방식의 상호인증이 수행된다. TC와 ZB간의 상호인증 작업이 성공적으로 진행되면 TC는 ZB에게 AP-링크키 LKAB를 보낸다.

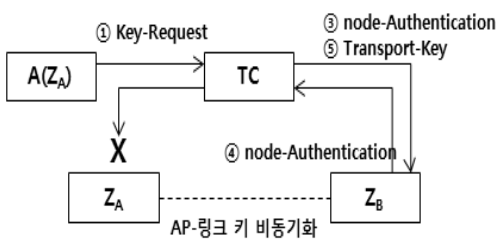
③ node-Authentication (TC, ZB) :
{ZA, RTC, MIC(LKB)}
④ node-Authentication (ZB, TC) :
{ZA, RTC, RB, MIC(LKB)}
⑤ Transport-Key (TC, ZB) :
{ZA, RB, [LKAB]LKB, MIC(LKB)}

Yuksel-Nielson의 제안은 그들이 지적한 문제점, TC를 가장한 공격자가 ZA와 ZB에게 이전에 사용되었던 AP-링크키를 재사용하게 하는 재생공격에 대응할 수 있게 된다.

IV. 개선된 ZigBee 키 분배 프로토콜 제안

4.1 Yuksel-Nielson 키 분배 프로토콜의 문제점

이미 언급한 바와 같이 Yuksel-Nielson의 키 분배 프로토콜은 기존의 ZigBee 키 분배 프로토콜과는 달리 TC를 가장한 재생공격에 대한 방어가 가능하다. 그러나 Yuksel-Nielson의 키 분배 프로토콜은 송신 노드 ZA에 대한 인증이 수행되지 않기 때문에 ZA를 가장한 재생공격에 취약하게 된다. 이것은 ZA에서 생성하는 난수 RA가 기존에 사용되었던 값이었는지를 확인하는 과정이 존재하지 않기 때문이다.



[그림 6] 키 비동기화 공격

따라서 〔그림 6〕에서와 같은 "키 비동기화 공격"이 가능하게 된다. 공격자 A(ZA)는 이전에 ZA가 TC에게 보냈던 AP-링크키 분배 요청 메시지를 도청해 두었다가 TC에게 재전송한다.

① Key-Request (ZA, TC) :
{ZB, RA, MIC(LKA)}old

TC는 공격자가 보낸 메시지를 ZA가 보낸 메시지로 인식한 후 새로운 AP-링크키를 생성하여 ZA와 ZB에게 전송한다. 그러나 동일하게 전송된 메시지는 ZA와 ZB에서 각각 다르게 처리된다.

② Transport-Key (TC, ZA) :
{ZB, RA, 〔LKAB〕LKA, MIC(LKA)}

ZA는 Key-Request 메시지를 요청한 적이 없으므로 Transport-Key 메시지가 무시되지만, ZB는 다음과 같이 TC와의 메시지 교환이 정상적으로 행해진다.

③ node-Authentication (TC, ZB) :
{ZA, RTC, MIC(LKB)}
④ node-Authentication (ZB, TC) :
{ZA, RTC, RB, MIC(LKB)}
⑤ Transport-Key (TC, ZB) :
{ZA, RB, 〔LKAB〕LKB, MIC(LKB)}

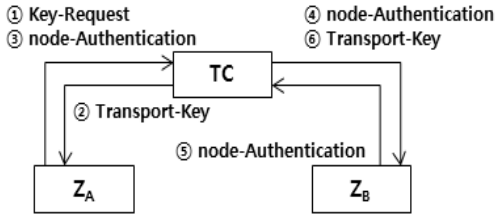
이는 궁극적으로 ZA와 ZB의 AP-링크키에 대한 "비동기화"를 야기 시켜 두 노드 간의 통신을 차단하게 되는 문제로 확장된다. 이러한 문제점의 근본적인 원인은 Yuksel-Nielson의 제안에서는 TC가 ZA가 보내는 메시지에 대한 인증기능이 결여되어 있기 때문이다.

4.2 제안 프로토콜

Yuksel-Nielson이 제안한 키 분배 프로토콜은 기존의 ZigBee 키 분배 프로토콜이 갖는 문제점 중에서 TC를 가장한 공격자의 방어를 가능하게 한다. 그러나 송신노드 Z_A에 대한 인증 절차가 존재하지 않는다는 문제점을 갖고 있다. 이는 TC가 아닌 다른 노드를 가장한 공격에는 취약할 수 있다. 따라서 본 논문에서는 2 가지의 개선된 ZigBee 키 분배 프로토콜을 제안한다.

4.1.1 제안 프로토콜 1

제안 프로토콜 1은 Yuksel-Nielson이 제안한 프로토콜의 단점을 보완하기 위하여 메시지 ③을 추가하였다. Yuksel-Nielson이 제안한 프로토콜은 ZA를 인증하는 절차가 없기 때문에, ZA를 가장한 공격에 취약하다는 단점을 갖고 있다. 따라서 메시지 ③을 통하여 TC가 ZA를 인증하는 과정을 추가하였다.



〔그림 7〕 제안 프로토콜 1

제안 프로토콜 1은 다음과 같다.

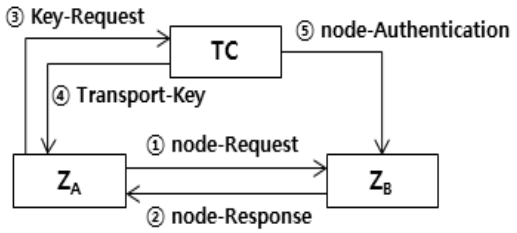
① Key-Request (ZA, TC) :
{ZB, RA, MIC(LKA)}
② Transport-Key (TC, ZA) :
{ZB, RA, RTC, 〔LKAB〕LKA, MIC(LKA)}
③ node-Authentication (ZA, TC) :
{ZA, RTC, MIC(LKA)}
④ node-Authentication (TC, ZB) :
{ZA, RTC, MIC(LKB)}
⑤ node-Authentication (ZB, TC) :
{ZA, RTC, RB, MIC(LKB)}
⑥ Transport-Key (TC, ZB) :
{ZA, RB, 〔LKAB〕LKB, MIC(LKB)}

③번 메시지를 제외한 프로토콜 1에서 수행되는 모든 절차는 Yuksel-Nielson이 제안한 프로토콜과 동일하다. ③번 메시지는 TC가 ZA를 인증하기 위한 목적으로 사용되는 메시지로 ZA는 자신이 정당한 노드라는 것을 증명하기 위하여 RTC를 다시 TC에게 보낸다. 이러한 과정을 추가함으로써 TC는 두 노드와 모두 상호 인증을 하게 되며 ZA를 가장한 공격자의 재생공격을 방지할 수 있다. 공격자가 ZA가 키 분배를 위해 TC에게 보낸 메시지를 도청하여 재전송한다

하더라도 ZA와의 상호 인증 절차가 제대로 수행되지 않기 때문에 ZB는 새로운 키를 분배 받을 수 없다. 따라서 ZA와 ZB가 서로 다른 키를 공유하게 되는 AP-링크키 "비 동기화 문제"는 발생하지 않는다.

4.1.2 제안 프로토콜 2

두 번째로 제안하고자 하는 키 분배 프로토콜의 핵심은 ZA와 ZB간에 공유될 AP-링크키의 생성을 TC와 ZB간에 사전에 공유하고 있는 TC-링크키를 기반으로 생성하는 데에 있다.



〔그림 8〕 제안 프로토콜 2

새로이 정의되는 node-Request 메시지와 node-Response 메시지를 이용한 제안 프로토콜 2는 다음과 같다.

① node-Request (ZA, ZB) :
{RA}
② node-Response (ZB, ZA) :
{RB, h(LKAB)}
③ Key-Request (ZA, TC) :
{RA, RB, ZB, h(LKAB), MIC(LKA)}
④ Transport-Key (TC, ZA) :
{RA, 〔LKAB〕LKA, MIC(LKA)}
⑤ node-Authentication (TC, ZB) :
{RB, MIC(LKB)}

먼저, ZA와 ZB는 메시지를 통해서 난수를 교환한다. 이 과정을 통해서 ZB는 ZA와 공유할 AP-링크키 LKAB = h(LKB, ZA, ZB, RA, RB)를 생성하고, 이를 기반으로 h(LKAB)을 계산하여 ZA에게 전달한다. 이 값은 ZA가 TC를 경유해 ZB를 인증하기 위한 목적으로 사용된다. ZA는 Key-Request 메시지를 통해서 AP-링크키를 요청하면, TC는 먼저 LKAB를 생성하고 h(LKAB)를 검증한다. 검증이 실패할 경우에는 이 값이 유효하지 않은 것으로 판단, 즉 ZB에 대한 인증이 실패한 것으로 간주하여 프로토콜은 더 이상 진행되지 않는다. 검증이 성공적이면 TC는 Transport-Key 메시지를 통해 ZA에게 AP 링크키 LKAB를 전송한다. 특히, 메시지에 포함된 RA는 ZA의 입장에서 TC를 challenge-response 방식으로 인증하기 위해 사용된다. 마지막으로 TC는 node-Authentication 메시지를 ZB에게 전달한다. 이 메시지의 목적은 ZB가 TC를 경유해서 간접적으로 ZA를 challenge-response 방식으로 인증하기 위해서이다. 만약, 인증이 실패한다면 ZB는 키 분배 프로토콜을 즉각 종료한다.

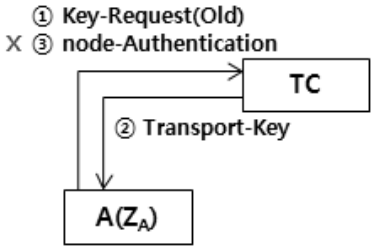
V. 안전성 분석

이번 장에서는 4장에서 제안된 2개의 새로운 키 분배 프로토콜을 세 가지의 공격 시나리오를 기반으로 그 안전성을 검증해 본다. 첫 번째 시나리오(공격 시나리오 1)는 ZA를 가장한 공격자 A(ZA)가 새로운 키 분배 프로토콜을 기동시켜 ZA와 ZB 간의 AP-링크키를 비동기화 시키는 공격이다. 두 번째 시나리오(공격 시나리오 2)는 TC를 가장한 공격자 A(TC)가 ZA로 부터의 새로운 키 분배 프로토콜을 시작하면, 이전에 도청된 Transport-Key 메시지를 재생시켜 이전에 양자 간에 공유된 AP-링크키로 회귀시키는 공격이다. 마지막으로 세 번째 시나리오(공격 시나리오 3)는 TC를 가장한 공격자 A(TC)가 임의로 ZB에게 이전에 보내어졌던 Transport-Key 메시지를 재생시켜 역시 ZA와 ZB 간의 AP-링크키를 비동기화 시키는 공격이다.

5.1 제안 프로토콜 1에 대한 안전성 분석

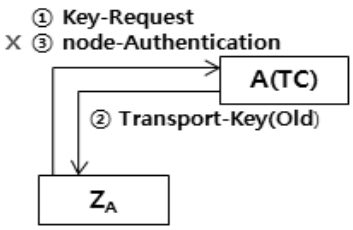
공격 시나리오 1은 ZA를 가장한 공격자 A(ZA)가 TC에게 AP-링크키를 요청하는 것으로 공격이 시작된다. 공격자는 이전에 ZA가 TC에게 AP-링크키를 요청하는 Key-Request 메시지를 도청해 두었다가 TC에게 전송한다. 해당 메시지를 받은 TC는 Transport-Key 메시지를 통해서 ZA를 인증하기 위한 새로운 challenge 값 RTC를 보낸다. 하지만, 공격자는 TC와 ZA간의 공유된 TC-링크키 LKA를 모르기 때문에 정상적인 node-Authentication 메시지를 보낼 수가 없게 된다. 따라서 일정시간이 지난 이후에 정상적인 메시지를 받지 못한다면 프로토콜은

거기서 종료된다. 따라서 ZA를 가장한 공격자의 공격은 방어가 가능하다.



〔그림 9〕 제안 프로토콜 1에 대한 공격 시나리오 1

공격 시나리오 2는 ZA가 TC로 가장한 공격자 A(TC)에게 새로운 AP-링크키를 요청하였을 때, 공격자는 이전에 사용되었던 Transport-Key 메시지를 도청하여 두었다가 그대로 재생하여 ZA에게 다시 전송한다. 그러나 ZA는 A(TC)가 보낸 Transport-Key 메시지를 통해서 자신이 Key-Request 메시지에 포함시킨 challenge 값 RA에 대한 정상적인 response 인지를 확인한다. 하지만, A(TC)는 TC와 ZA간의 공유된 TC-링크키 LKA를 역시 모르기 때문에 정상적인 Transport-Key 메시지를 보낼 수가 없다. 따라서 A(TC)에게 자신이 정당한 노드라는 것을 증명하기 위하여 ZA가 보내는 node-Authentication 메시지는 전송되지 않고, 프로토콜은 더 이상 진행되지 않는다.



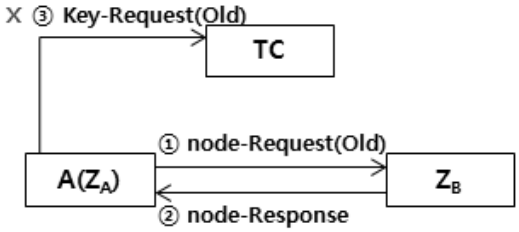
〔그림 10〕 제안 프로토콜 1에 대한 공격 시나리오 2

공격 시나리오 3은 TC를 가장한 공격자 A(TC)가 ZB에게 이전에 TC가 ZB에게 보냈던 node-Authentication 메시지를 도청해 두었다 전송한다. ZB는 ZA가 새로운 키 분배를 요청한 것이라 여기고 TC로 가장한 공격자와의 상호인증을 위한 프로토콜을 수행한다. ZB는 A(TC)에게 보내는 node-Authentication 메시지에 자신이 정당한 노드라는 것을 증명하기 위하여 A(TC)가 보낸 메시지에 포함된 RTC(old)와 TC를 인증하기 위하여 자신이 생성한 RB를 포함하여 A(TC)에게 보낸다. 메시지를 받은 A(TC)는 기존에 도청해 두었던 TC가 ZB에게 보낸 Transport-Key 메시지를 ZB에게 재전송 하지만, 도청해 두었던 메시지의 RB(old)는 ZB가 A(TC)에게 보냈던 RB와 다르므로 ZB는 A(TC)의 메시지를 거부한다. 따라서 TC를 가장한 공격자가 ZB에게 키를 갱신시켜 키 비동기화를 시도하는 공격은 방어가 가능하다.

5.2 제안 프로토콜 2에 대한 안전성 분석

제안 프로토콜 2에서 ZA를 가장한 공격자 A(ZA)가 취할 수 있는 공격 시나리오 1에는 A(ZA)가 ZB에게 node-Request 메시지를 전송하는 공격과 A(ZA)가 TC에게 Key-Request 메시지를 전송하는 공격이 있다.

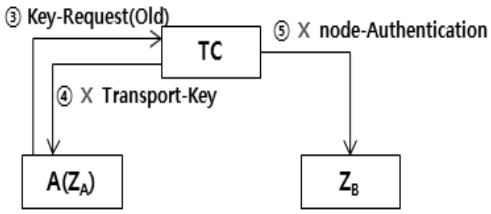
첫번째 공격은 A(ZA)가 ZB에게 node-Request 메시지를 전송하는 것으로 시작된다. 메시지를 받은 ZB는 자신이 새로이 생성한 RB를 포함한 메시지를 공격자에게 전달한다. 공격자의 입장에서는 ZA와 TC간에 공유된 TC-링크키 LKA를 모르기 때문에 ZB가 생성한 RB를 포함하는 유효한 Key-Request 메시지를 구성할 수 없고, 그 이전에 도청한 메시지를 재생할 경우에도 TC에 의해서 거부되어 질 수 밖에 없다. 즉, 프로토콜은 더 이상 진행되지 않는다.



〔그림 11〕 제안 프로토콜 2에 대한 공격 시나리오 1-1

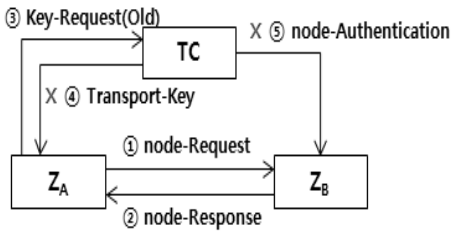
두 번째 공격은 A(ZA)가 TC에게 기존에 도청해 두었던 Key-Request 메시지를 보내는 것으로 시작되며 이 메시지를 보내는 경우는 2가지가 있다. 첫째로 ZA와 ZB사이에 node-Request 메시지와 node-Response 메시지의 교환이 없는 상태에서 A(ZA)가 TC에게 Key-Request 메시지를 재전송하는 경우이다. 이 경우에 ZA와 ZB는 TC로부터 각

각 Transport-Key 메시지와 node-Authentication 메시지를 받게 되지만, node-Request 메시지나 node-Response 메시지의 교환이 없으므로 자신들이 해당 프로토콜을 진행하지 않았기 때문에 TC에게서 받은 메시지를 거부하게 된다. 따라서 프로토콜은 더 이상 진행되지 않는다.



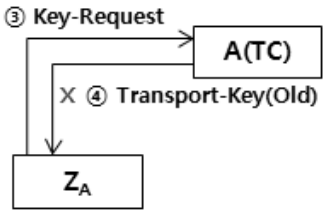
〔그림 12〕 제안 프로톡콜 2에 대한 공격 시나리오 1-2

둘째로 현재 세션에서 ZA와 ZB사이에 node-Request 메시지와 node-Response 메시지가 교환된 후에 A(ZA)가 이전에 도청해 두었던 Key-Request 메시지를 재전송할 경우에는 node-Request 메시지에 있는 난수 RA와 node-Response 메시지에 있는 난수 RB가 A(ZA)가 전송한 Key-Request 메시지에 있는 난수 RA, RB와는 상이하다. 따라서 TC가 보내는 Transport-Key 메시지와 node-Authentication 메시지는 ZA와 ZB에서 거부되어 지므로 프로토콜은 더 이상 진행되지 않는다.



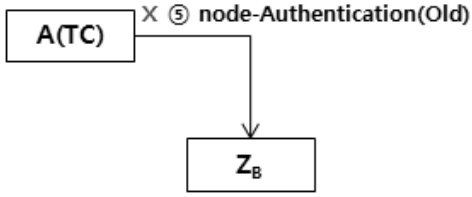
〔그림 13〕 제안 프로토콜 2에 대한 공격 시나리오 1-3

제안 프로토콜 2에 대한 공격 시나리오 2는 ZA가 새로운 키 분배 프로토콜을 시작하면 먼저 ZB와 node-Request 및 node-Response 메시지를 교환해서 수합된 RA 및 RB를 탑재한 Key-Request 메시지를 TC를 가장한 공격자 A(TC)에게 보내는 것으로 시작된다. 이 경우에 공격자는 역시 ZA와 TC 간에 공유된 TC-링크키 LKA를 모르기 때문에 유효한 Transport-Key 메시지를 ZA에게 보낼 수가 없다. 그 이전에 도청된 메시지에는 현재의 RA 및 RB가 포함되어 있지 않기에 재생되어 질 수 없다. 따라서 이 공격은 성공할 수 없다.



〔그림 14〕 제안 프로토콜 2에 대한 공격 시나리오 2

공격 시나리오 3은 TC를 가장한 공격자 A(TC)가 ZB에게 이전에 TC가 ZB에게 보냈던 node-Authentication 메시지를 도청해 두었다 전송한다. 제안 프로토콜 2에 대한 공격에는 두 가지 경우가 있을 수 있다. 첫째는 ZA로 부터의 키 분배 프로토콜이 시작되지 않은 상태에서 공격자가 ZB에게 node-Authentication 메시지를 보내는 경우이다. 하지만 ZB의 입장에서는 node-Request 메시지를 받지 못한 상태에서 그 메시지를 받게 되면 그냥 거부하게 된다. 둘째는 ZA 또는 A(TC)가 node-Request 메시지를 보낸 상태에서 ZB에게 node-Authentication 메시지를 보내는 경우이다. 하지만 이 경우에도 A(TC) 입장에서는 ZB와 TC 간에 공유된 TC-링크키 LKB를 모르기 때문에 유효한 node-Authentication 메시지를 ZB에게 보낼 수가 없다.



〔그림 15〕 제안 프로토콜 2에 대한 공격 시나리오 3

5.3 공격 시나리오에 대한 안전성 비교분석

이미 언급한대로 제안 프로토콜 1과 2는 세 가지의 공격 시나리오에 안전하다. 하지만, Yuksel-Nielson이 제안한 키 분배 프로토콜은 4.1절에서 분석한 바와 같이 공격 시나리오 1을 기반으로 한 "키 비동기화 공격"에 안전하지 않은 것으로 나타났다. 공격 시나

리오 1이 유효한 근본적인 이유는 Yuksel-Nielson이 제안한 키 분배 프로토콜에는 ZA를 인증하기 위한 기능이 결여되어 있기 때문이다. 프레임 카운터에 기반을 둔 ZigBee 키 분배 프로토콜은 Yuksel-Nielson이 지적한 문제점을 그대로 인정한다면 세 가지 공격 시나리오 모두에 대해서 안전하지가 않다.

〔표 2〕 프로토콜별 안전성 분석

O:secure, X:insecure

공격	ZigBee-2007	Yuksel-Nielson	제안1	제안2
scenario1	X	X	○	○
scenario2	X	○	○	○
scenario3	X	○	○	○

VI. 성능분석

6.1 ZigBee 패킷의 구조

ZigBee는 IEEE 802.15.4 표준에서 정의된 PHY와 MAC을 그대로 활용하며 그 상위에 네트워크 계층과 응용 계층을 정의하여 ZigBee 스택을 구성한다. ZigBee 네트워크 계층에서는 기기 간의 상호연결, 데이터 전송과 재전송, 각종 네트워크 형태의 구성 등을 표현한다. 응용 계층에서는 주로 상위 계층과 네트워크 계층 간의 연동 역할을 하며 주변 연결 장치들의 바인딩 정보를 정의하고 저장한다.

ZigBee에서 데이터는 패킷으로 보내지는데 대부분 135 바이트 이하로 구성되며 IEEE 802.15.4의 MAC계층을 이용해서 한번에 123 바이트까지 원하는 데이터를 보낼 수 있다. 나머지 12 바이트에는 패킷의 길이 정보 및 FCS(Frame Check Sequence)를 통해 전송오류를 확인할 수 있는 정보가 포함된다. ZigBee 패킷의 구조는 다음 〔그림 16〕과 같다.

[단위 : Octets]

5	1	5/25	8	8	0/variable	variable	2
SYNC	PHY HDR	MAC HDR	NWK HDR	APS HDR	Auxiliary HDR	Encrypted APS Payload	MIC

〔그림 16〕 ZigBee 패킷의 구조

5 바이트의 SYNC는 물리신호 동기화에 대한 필드이며, PHY HDR(PHY Header)는 1 바이트로 구성된다. MAC HDR(Mac Header)의 경우 전송되는 프레임의 형태를 식별하는 프레임 제어(Frame Control)) 필드, 중복전송을 제어하는 시퀀스 번호(Sequence Number) 필드 및 주소(Address) 필드를 정의한다. MAC 프레임은 비컨(Beacon) 프레임, 데이터(Data) 프레임, Ack 프레임, MAC 명령(Command) 프레임 총 4가지 종류를 가지고 있다. 이 4가지의 프레임 중 비컨 프레임과 데이터 프레임만 상위계층에 전달되고 나머지 2개 프레임은 MAC계층 간의 제어를 위해 사용된다. NWK HDR(Network Header)는 8바이트로 고정되어 있으며 프레임 제어 필드와 주소 필드 및 브로드캐스트와 관련된 필드로 구성된다. 2바이트인 프레임 제어 필드는 데이터 프레임인지 명령 프레임인지 구분하고, 현재 프로토콜 버전 정보와 라우팅과 보안에 관련된 정보를 포함한다. APS HDR(Application Header)는 프레임 제어 필드와 주소 필드로 구성되어 있다. 또한 NWK HDR와 마찬가지로 고정되어 있지만 주소 필드의 경우는 포함되지 않을 수 있다. 프레임 제어 필드에 있는 프레임 타입의 서브 필드에는 데이터 프레임을 지시하는 값이 포함되어 있으며 데이터 프레임의 용도와 목적에 따라 알맞게 설정된다. APS Payload는 각 노드의 상황에 따라서 기존의 ZigBee 표준에서의 Key-Request 명령 프레임, Transport-Key 명령 프레임 및 본 논문에서 새로 제안한 node-Authentication 명령 프레임, node-Request 명령 프레임, node-Response 명령 프레임으로 구성된다. 본 논문에서는 응용계층 부분의 프레임 변경 및 추가를 통해 제안 프로토콜의 보안성과 에너지 소모율을 평가해보고자 한다.

6.2 ZigBee 명령 프레임

본 논문에서 제안되는 프로토콜에는 기존에 ZigBee-2007 표준에서 제시된 명령 프레임(Command Frame) 이외에도 새로운 명령 프레임이 요구된다. 이에 기존 ZigBee-2007 표준에서 제시되었던 명령 프레임의 수정 및 새로이 요구되는 명령 프레임들을 제안한다. Key-Request 명령 프레임의 경우 기존 ZigBee 표준에서 제시되어 있으며 Random number 필드만 추가되었다. 세부 필드는 〔그림 17〕과 같다.

[단위 : Octets]

1	1	1	1	4	0/4	0/8
Frame control	APS counter	APS command identifier	Key type	Random number	Auxiliary random number	Partner address
APS header		Payload				

〔그림 17〕 Key-Request Command Frame

Transport-Key 명령 프레임의 경우 기존 ZigBee-2007 표준에서 제시되어 있으며 Random number 필드만 추가되었다. 세부 필드는 〔그림 18〕과 같다.

[단위 : Octets]

1	1	1	1	4	Variable
Frame control	APS counter	APS command identifier	Key type	Random number	Key descriptor
APS header		Payload			

〔그림 18〕 Transport-Key Command Frame

node-Authentication 명령 프레임의 경우 ZigBee-2007 표준을 기반으로 새롭게 정의되어 제시되었으며 세부 필드는 〔그림 19〕와 같다.

[단위 : Octets]

1	1	1	4	0/4
Frame control	APS counter	APS command identifier	random number	Auxiliary random number
APS header		Payload		

〔그림 19〕 node-Authentication Command Frame

node-Request 명령 프레임 및 node-Response 명령 프레임의 경우 ZigBee-2007 표준을 기반으로 새롭게 정의되어 제시되었으며 세부 필드는 각각 〔그림 6.5〕와 〔그림 6.6〕과 같다.

[단위 : Octets]

1	1	1	4
Frame control	APS counter	APS command identifier	Source Random number
APS header		Payload	

〔그림 20〕 node-Request Command Frame

[단위 : Octets]

1	1	1	4	16
Frame control	APS counter	APS command identifier	Source Random number	Data
APS header		Payload		

〔그림 21〕 node-Response Command Frame

명령 프레임의 Random number 필드를 제외한 나머지 필드인 프레임 제어(Frame Control), APS 카운터(Counter), APS 명령식별(Command Identifier) 프레임은 기존 ZigBee-2007 표준에서 명시한 것과 동일하다. node-Response 명령 프레임의 Data 필드 또한 표준에서 명시된 것과 같이 명령 식별 필드에 의존한다. (세부사항은 〔1〕을 참조)

6.3 에너지 소모율 성능분석

본 논문에서는 ZigBee-2007 표준 명세 및 Yuksel-Nielson 프로토콜, 제안 프로토콜 1, 제안 프로토콜 2의 에너지 소모율을 이용하여 제안 프로토콜의 성능을 분석하였다. 에너지 소모율은 TC, ZA, ZB 각각의 ZigBee 노드에 대해서 분석하였으며 성능분석은 〔5, 6〕에서 사용했던 ATRF230 칩을 기준으로 분석하였다. 성능분석은 ZigBee 노드 간의 메시지 수와 길이를 기준으로 에너지 소모율을 분석하였다. 성능분석시 각 노드에 공급되는 전력은 2.4V로 동일하며 Receive(Rx)와 Transmit(Tx)의 전류(IRx = ITx)는 같다고 가정한다. 또한 메시지 전송 및 수신에 소요되는 에너지 소모율이 훨씬 크기 때문에 암호 알고리즘을 수행하는 에너지는 무시한다. (1비트를 전송하는 데에 소요되는 에너지는 800 ~ 1000 개의 CPU 명령어를 수행하는 데에 소요되는 에너지와 동일함 〔10〕) 〔5〕에서 사용했던 IRIS 데이터시트에서 명시된 것을 기준으로 ATRF230 칩에서의 RX, TX 모드 기간 동안에는 17 mA의 전류를 사용함을 알 수 있다. 또한 각 노드의 파워는 3.2 dBm으로 동일하며 해당 노드 간에 유지되어야 하는 시간은 각 프로토콜에 따라 달라진다. 만약 한 노드가 메시지를 보낼 경우, 해당 노드는 활성모드로 변하고 매체 접근 전에 CCA(Clear Channel Assessment)를 수행한다. CCA 검출시간은 8 symbol (16 μs/symbol) 동안으로 정의 된다 〔6〕.

〔표 3〕 에너지 소모율 계산에 이용되는 기호

기호	설명	단위
U	노드의 전력 공급량 값	V
I	Tx모드 동안의 현재 전류 소비량 값 (파워 = 3.2dBm)	A
t	메시지 처리 시간	sec

〔표 4〕 에너지 소모율

ETx = U〔V〕 x I〔A〕 x t〔s〕

노드가 성공적으로 매체에 접근하면, 메시지를 전송한다. 메시지 처리기간의 Rx/Tx 모드 상에서의 장치 소비시간은 해당 ZigBee 칩 (b=250kbps for

ATRF230)의 bitrate〔kbps〕와 메시지 사이즈로부터 유도된다. 예를 들어 메시지의 크기가 k=12 bytes이면, 메시지의 처리시간은 tk = (k x 8)/(b x 1000) = 0.38 ms과 같다. 그 다음 전송기간 동안의 에너지 소모율 ETx는〔표 4〕와 같다〔5〕. 즉, 앞의 예에 적용해 보면 2.4V x 1.7mA x 0.38ms = 15 uJ 라는 결과를 얻을 수 있다. 다음〔표 5〕은 에너지 소모율 식 중 메시지 처리 시간 계산에 이용되는 기호들을 정리해 놓은 것이다. 각 x, y는 (x y) 형식으로 표현되며 "x 모드로 y 메시지를 처리하는데 소모되는 시간"을 의미한다.

〔표 5〕 메시지 처리 시간에 이용되는 기호

	기호	설명
x	Rx	수신 모드
	Tx	전송 모드
	Rx/Tx	수신 모드에서 전송모드로 전환(192 μs)
		전송 모드에서 수신모드로 전환(192 μs)
y	receiver	수신기 전환
	sender	전송기 전환
	ack	Acknowledgement
	Key-Request	키 요청 메시지
	Transport-Key	키 전송 메시지
	node-Authentication	노드 인증 메시지
	node-Request	노드 링크키 요청 메시지
	node-Response	노드 링크키 응답 메시지

〔표 6〕는 ZigBee-2007 표준 명세, Yuksel-Nielson 프로토콜, 제안 프로토콜 1, 제안 프로토콜 2에 따른 TC, ZA, ZB의 총 전송시간 ttotal에 대한 계산식이다. 메시지는 수신 장치에 의해 수신되며 Rx로부터 Tx 모드로 전이되며 (192μs), ACK 메시지 (ACK 크기: 5bytes)가 역으로 보내진다〔5〕. 전류 IRx = ITx 라는 가정 하에 활성모드에서만 계산되었으며 전송 노드에서 전체 에너지 E의 소모율은 활성모드 동안 소비되는 시간으로부터 유도할 수 있다.

CCA등 각 프레임 및 모드의 전송 처리시간에 대한 자세한 설명은〔6〕에서 찾을 수 있다.

〔표 6〕의 식으로 총 전송시간을 구한 뒤 처음에 언급한 에너지 소모율 계산식〔표 4〕에 대입하여 성능분석 한 결과는 다음〔표 7〕, 〔그림 22〕와 같다.

〔표 6〕 프로토콜에 따른 각 노드의 처리시간

a. ZigBee-2007 표준 명세	
t_{total_TC}	= (CCA) + (Rx/Tx receiver) + (Rx Key-Request) + (Rx/Tx sender) + (Tx Transport-Key) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx sender) + (Tx Transport-Key) + (Rx/Tx receiver) + (Rx ack)
t_{total_ZA}	= (CCA) + (Tx Key-Request) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx receiver) + (Rx Transport-Key)
t_{total_ZB}	= (CCA) + (Rx/Tx receiver) + (Rx Transport-Key)

b. Yulsel-Nielson	
t_{total_TC}	= (CCA) + (Rx/Tx receiver) + (Rx Key-Request) + (Rx/Tx sender) + (Tx Transport-Key) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx sender) + (Tx node-Authentication) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx receiver) + (Rx node-Authentication) + (Rx/Tx sender) + (Tx Transport-Key) + (Rx/Tx receiver) + (Rx ack)
t_{total_ZA}	= (CCA) + (Tx node-Request) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx receiver) + (Rx Transport-Key)
t_{total_ZB}	= (CCA) + (Rx/Tx receiver) + (Rx node-Authentication) + (Rx/Tx sender) + (Tx node-Authentication) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx receiver) + (Rx Transport-Key)

c. 제안 프로토콜1	
t_{total_TC}	= (CCA) + (Rx/Tx receiver) + (Rx Key-Request) + (Rx/Tx sender) + (Tx Transport-Key) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx receiver) + (Rx node-Authentication) + (Rx/Tx sender) + (Tx node-Authentication) + (Rx/Tx receiver) + (Rx ack) +

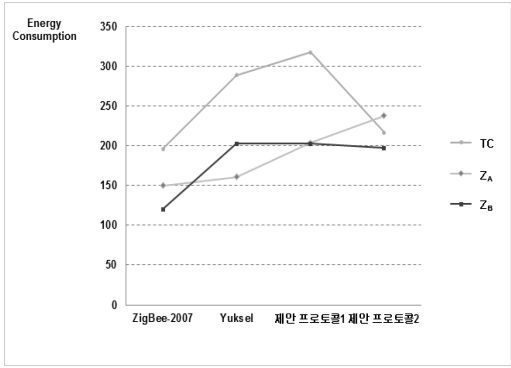
	(Rx/Tx receiver) + (Rx node-Authentication) + (Rx/Tx sender) + (Tx Transport-Key) + (Rx/Tx receiver) + (Rx ack)
t_{total_ZA} =	(CCA) + (Tx Key-Request) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx receiver) + (Tx Transport-Key) + (Rx/Tx sender) + (Tx node-Authentication) + (Rx/Tx receiver) + (Rx ack)
t_{total_ZB} =	(CCA) + (Rx/Tx receiver) + (Rx node-Authentication) + (Rx/Tx sender) + (Tx node-Authentication) + (Rx/Tx receiver +Rx ack) + (Rx/Tx receiver) + (Rx Transport-Key)

d. 제안 프로토콜2	
t_{total_TC} =	(CCA) + (Rx/Tx Receiver) + (Rx Key-Request) + (Rx/Tx sender) + (Tx Transport-Key) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx sender) + (Tx node-Authentication) + (Rx/Tx receiver) + (Rx ack)
t_{total_ZA} =	(CCA) + Tx node-Request) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx receiver) + (Rx node-Response) + (Rx/Tx sender) + (Tx Key-Request) + (Rx/Tx receiver) + (Rx ack) + (Rx/Tx receiver) + (Rx Transport-Key)
t_{total_ZB} =	(CCA) + Rx/Tx receiver) + (Rx node-Request) + (Rx/Tx sender) + (Tx node-Response) + (Rx/Tx receiver + Rx ack) + (Rx/Tx receiver) + (Rx node-Authentication)

〔표 7〕 에너지 소모율 성능분석 결과

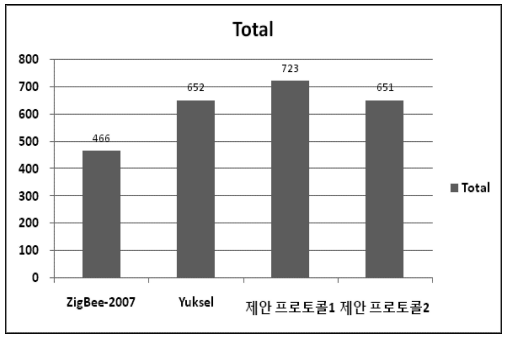
〔단위 : uJ〕

노드 \ 프로토콜	ZigBee-2007	Yuksel-Nielson	제안1	제안2
TC	195	288	317	216
ZA	150	160	203	237
ZB	120	202	202	197



〔그림 22〕 에너지 소모율 성능분석 결과

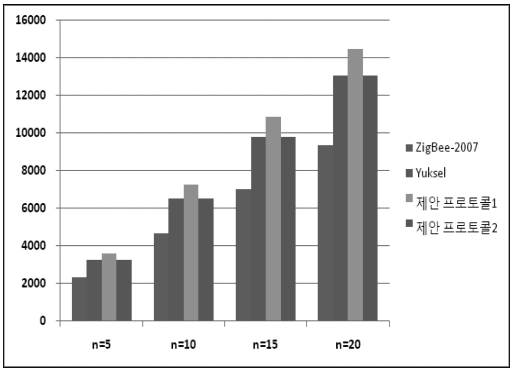
또한 임의의 노드 수 일 때의 전체 에너지 소모율은 프로토콜의 반복 수행 횟수와 비례하는데 이는 2개의 노드와 1개의 TC 또는 20개의 노드와 1개의 TC의 경우에서도 확인 할 수 있다. 궁극적으로 반복적인 프로토콜 수행에 따른 전체 에너지 소모율은 2개의 노드와 1개의 TC간의 반복적인 프로토콜 수행에 따른 전체 에너지 소모율로 표현이 가능하다. 각각의 프로토콜에 소요되는 전체 에너지 소모율의 결과는 다음 〔그림 23〕과 같다.



〔그림 23〕 각각의 프로토콜에 소요되는 전체 에너지 소모율

이러한 과정을 n번 수행할 경우의 에너지 소모율은 프로토콜을 1번 수행하는 데에 필요한 에너지 소모율의 n배가 된다. n=5, 10, 15, 20일 경우의 각각의 프로토콜 수행에 따른 전체 에너지 소모율은 다음 〔그림 24〕와 같다.

성능분석의 결과로 제안된 프로토콜의 TC, ZA, ZB에서의 에너지 소모율에 대한 결과를 알 수 있었다. TC를 기준으로 제안 프로토콜 1은 ZigBee-2007 표준과는 122 uJ의 에너지 소모율의 차



〔그림 24〕 n = 5, 10, 15, 20일 경우

이가 났지만 Yuksel-Nielson의 제안 프로토콜과는 29 uJ 차이 밖에 나지 않았다. 이는 기존 ZigBee-2007 표준에서나 Yukel-Nielson이 제안했던 프로토콜보다 더 보안성이 뛰어난 프로토콜을 제시하였음에도 에너지 소모율에서도 큰 차이가 나지 않음을 의미한다. 보안성 강화를 위해 전체적인 메시지 길이와 ZigBee 노드 간에 주고받는 메시지의 수가 늘어났지만 실제적으로 ZigBee 노드의 에너지 소모율에는 큰 영향을 미치지 않았다. 마찬가지로 제안 프로토콜 2도 ZigBee-2007 표준과 21 uJ 정도밖에 차이가 나지 않으며 이 프로토콜의 경우에는 TC를 기준으로 Yulsel-Nielson 제안 프로토콜보다 에너지 소모율이 낮았다. ZA 기준에서는 97 uJ 정도의 차이를 보이기는 하지만 TC와 ZB 기준에서는 에너지 소모율이 낮다는 것을 알 수 있다. 제안 프로토콜 2도 제안 프로토콜 1과 마찬가지로 보안성의 향상을 위해 메시지 길이와 수가 늘어났지만 제안된 프로토콜과의 에너지 소모율에서 큰 차이가 나지 않았음을 확인할 수 있었다.

VII. 결론

본 논문에서는 Yuksel-Nielson이 ZigBee 표준 명세서에 나타나 있는 키 분배 프로토콜의 문제점을 지적하고 새로이 제시한 키 분배 프로토콜이 키 비동기화 공격에 취약함을 보였다. 그러한 취약점의 근본적인 원인은 그들이 제시한 프로토콜에는 프로토콜을 개시하는 ZigBee 노드에 대한 인증기능이 적절히 작동하지 않기 때문이었다. 이를 기반으로 2개의 새로운 키 분배 프로토콜을 제안 하고, 3가지의 공격 시나리오를 가정해서 본 논문에서 제안된 2개의 키 분배 프로토콜이 안전함을 입증하였다. 마지막으로 Yuksel-Nielson 기법과 본 논문에서 새로이 제시된 2개의 제안 프로토콜들에 대한 효율성을 에너지 소모율 측면에서 비교하였다.

참고문헌

〔1〕 ZigBee Alliance, "ZigBee-2007 Specification," ZigBee Document 053474r17, Jan. 2008.

〔2〕 IEEE, "Wireless Medium Access Control and Physical Layer Specifications for Low-Rate Wireless Personal Area Networks," IEEE 802.15.4-2003, May 2003.

〔3〕 E. Yüksel, H. R. Nielson, and F. Nielson, "A Secure Key Establishment Protocol for ZigBee Wireless Sensor Networks," The Computer Journal, Vol. 54, No. 4, pp. 589-601, Apr. 2011.

〔4〕 N. Sastry and D. Wagner, "Security considerations for IEEE 802.15.4 networks," Proceedings of the 2004 ACM workshop on Wireless security, pp. 32-42, Oct 2004.

〔5〕 M. Simek and P. Moravek, "Modeling of Energy Consumption of ZigBee Devices in Matlab Tool," Elektrorevue, Vol. 2, No. 3, pp. 41-46, Sep. 2011.

〔6〕 Jennic Inc, "Calculating 802.15.4 Data Rates," JN-AN-1035, Jennic Inc, Aug. 2006.

〔7〕 P. Radmand, M. Domingo, J. Singh, J. Arnedo, A. Talevski, S. Petersen and S. Carlsen, "ZigBee/ZigBee PRO security assessment based on compromised cryptographic keys," Proceedings of the 2010 International Conference on P2P, Parallel, Grid, Cloud and Internet Computing, pp. 465-470, Nov. 2010.

〔8〕 H. Li, Z. Jia and X. Xue, "Application and Analysis of ZigBee Security Services Specification," 2010 Second International Conference on Networks Security, Wireless Communications and Trusted

Computing, pp. 494-497, Apr. 2010.
〔9〕 G. Dini and M. Tiloca, "Considerations on Security in ZigBee Networks," Proceedings of the 2010 IEEE International Conference on SUTC, pp.58-65, Jun. 2010.
〔10〕 L. Gheorghe, R. Raazvan, and N. Tapus, "Energy-efficient Optimizations of the Authentication and Anti-replay Security Protocol for Wireless Sensor Networks," The Seventh International Conference on Networking and Services, pp. 201-207, May 2011.

〈著者紹介〉

오 수 민 (Su-min Oh) 학생회원
2011년 2월: 단국대학교 컴퓨터과학과 졸업
2011년 3월~현재: 단국대학교 전자계산학과 석사과정
〈관심분야〉 정보보호

최 수 경 (Soo-kyeong Choi) 학생회원
2011년 8월: 건양대학교 컴퓨터학과 졸업
2011년 9월~현재: 단국대학교 전자계산학과 석사과정
〈관심분야〉 정보보호

권 예 진 (Ye-Jin Kwon) 학생회원
2011년 2월: 단국대학교 컴퓨터과학과 졸업
2011년 9월~현재: 단국대학교 전자계산학과 석사과정
〈관심분야〉 정보보호

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 컴퓨터과학과 교수
〈관심분야〉 정보보호, 네트워크 보안, 무선 인터넷 및 모바일 컴퓨팅 보안, 금융보안