유비쿼터스 센서 네트워크에서의 키 분배 프로토콜 구현 및 분석

조관태*, 이화성*, 김용호*, 이동훈*

*고려대학교 정보보호대학원

The Implementation and Analysis of Key Distribution in USN

Kwan-Tae Cho*, Hwa-Seong Lee*, Yong-Ho Kim*, Dong-Hoon Lee*

*Korea University of Center for Information Security Technology

요 약

유비쿼터스 센서 네트워크(Ubiquitous Sensor Network, USN)는 차세대 네트워크로 각광받고 있고 이러한 센서 네트워크의 보안에 대한 관심은 더욱 높아지고 있다. 왜냐하면 센서 네트워크를 외부 공격자로부터 보호하기 위해서는 센서 노드 사이의 암호화 통신과 인증 등이 필요하기 때문이다. 이를 위해서는 안전한 키 생성 및 폐기가 우선 이루어져야 한다. 지금까지 센서 네트워크를 위해 제안된 키 분배 프로토콜은 매우 다양하다. 그러나 이러한 프로토콜들이 기반하고 있는 가정들이 얼마나 안전하고 현실성 있는 지 분석되어 있지 않다. 키 분배의 대표적 프로토콜인 LEAP(Localized Encryption and Authentication Protocol)은 무선 선세 네트워크에 적합한 가정을 하였다고 알려져 있다. 본 논문에서는 이 LEAP 프로토콜을 USN 상에서 구현을 통해 이 가정의 현실성에 대해 분석해 볼 것이다. 이는 차후 다양한 키 분배 프로토콜을 연구·개발하는 데 있어서 상당한 도움이 될 거라 예상된다.

I. 서론

USN 환경은 유비쿼터스 컴퓨팅 구현을 위한 기반 네트워크로 초경량, 초전력의 많은 센서 노드들로 구성된 무선 네트워크이다. USN은 수많은 센서 노드들이 필드의 지리·환경적 변화를 감지하여 싱크로 그 정보를 전달한 후 싱크와 유·무선으로 연결된 서버를 통해 사용자에게 전달되는 방식으로 정보 수집이 이루어진다[2]. 이러한 USN은 센서 노드를 통한 정보 감지 및 감지된 정보를 처리하는 기능을 수행함으로써 우리의 삶을 자동화시키고 편리함을 제공하지만 일상생활에서 시스템에 의존도가 높아질수록 이로 인한 위험성 또한 높아질 수밖에 없다. 그러므로 USN을 통해 제공되는 정보들을 신뢰하고 동시에 개인의 프라이버시를 보장 받을 수 있도록 하기 위해 많은 센서 네트워크 보안에 관한 연구가 활발히 진행되고 있다.

USN 환경에서의 키 관리 프로토콜은 초경량 센서의 자원 한계로 인해서 현실적인 가정을 기반으로 안전성 혹은 효율성을 높이고 있다. 여기서 중요한 점은 키 관리 프로토콜이 실제로 USN에 적용 되었을 때 가정의 오류로 인한 문제점이 발생하지 말아야 한다는 점이다. 만약 프로토콜이 암호학적으로 안전하다 할지라도 가정이 올바르지 않다면 실제 USN에서는 적용 불가능할 것이다.

따라서 논문에서 제안한 프로토콜의 안전성을 분석하기 전에 그 논문이 주장한 가정이 타당한지 검토해 보아야 된다. 본 논문에서는 USN에서 가장 널리 사용되고 있는 LEAP[1]의 초기 안전성을 실제 USN에 구현해 봄으로써 현실 가능성에 대해 분석해 볼 것이다. 이는 향

* 본 연구는 정보통신부 및 정보통신 연구진흥원의 대학 IT연구센터 지원 사업으로 수행되었음

후 키 관리 프로토콜의 설계에 유용한 정보로 사용 될 것이라 예상한다.

Ⅱ절에서는 이 논문에서 분석한 키 분배 프로토콜인 LEAP에 대해 간략히 소개하고 Ⅲ절에서는 LEAP 프로토콜의 구현에 대해 알아볼 것이다. Ⅳ절에서는 Ⅲ절의 구현을 토대로 LEAP 프로토콜의 가정의 안전성을 분석할 것이며, 마지막으로 Ⅴ절은 앞 절들을 토대로 이 논문의 결론을 내린다.

II. LEAP

USN 환경에서의 노드들은 계산 능력이 뛰어나지 못하고 에너지 사용에 대한 제한이 있다. 이러한 제약 사항을 만족시키면서 안전하고 효율적인 네트워크를 구성하는 것이 USN에서는 중요하다. 그래서 LEAP에서는 전자를 위해 대칭키를 사용하여 초경량으로 메시지 암·복호화를 하고, 후자를 위해 노드가 수신한 메시지를 포워딩(Forwarding) 혹은 프로세싱(Processing)하기 전에 노드 간 MAC 확인을 통해 인증(Authentication)함으로써 도스(DoS)와 같은 악의적인 공격을 막아 에너지 소모를 줄였다.

2.1 가정

LEAP은 노드 배치 전에 동일한 Initial 키를 모든 노드에게 사전 저장시키고 Initial 키를 사용하여 노드간의 Pair-wise 키를 생성한다. 그래서 Initial 키의 안전성은 매우 중요 하다. Initial 키의 안전성을 보장하기 위해 이 논문에서는 $T_{min} > T_{est}$라는 가정을 정의 하였다. T_{min}은 공격자가 노드 하나를 포획하여 비밀 정보를 얻기까지의 최소 시간을 의미하고 T_{est}는 공격자가 이웃 노드를 발견하고 Pair-wise 키를 생성하는데 걸리는 예상 시간을 의미한다. 즉, 주어진 가정의 의미는 공격자가 노드를 포획하여 비밀 정보를 얻기 전에 이미 노드들 간의 Pair-wise 키 생성이 끝났다는 것을 가정한다. 우리가 제안한 논문에서는 프로토콜의 효율성을 높이기 위해 초기 안정성을 떨어뜨리는 이 가정의 타당성을 4절에서 분석하였다.

2.2 Pair-wise 키

Pair-wise 키가 생성되는 과정은 총 4단계로 구성된다.

2.2.1 키 사전 분배

각각의 노드는 사전 저장된 Initial 키와 자신의 ID를 이용하여 Master 키를 생성한다. 예를 들어 노드 u가 존재할 때, 노드 u는 공통의 Initial 키 K_i와 자신의 ID u로 자신의 Master 키 K_u를 생성한다.

$$K_u = F_{K_i}(u)$$

2.2.2 이웃 노드 발견

노드 u는 $T_{\min}$타이머를 동작 시키고 자신의 ID u와 난수 N_u를 브로드캐스트 한다. 이를 수신한 노드 v는 자신의 Master 키 K_v로 MAC을 만든 후, 자신의 ID와 함께 전송한다. 노드 u는 자신의 Initial 키로 노드 v의 Master 키를 계산한 후 노드 v의 MAC을 확인하여 메시지를 인증하게 된다.

$$u \rightarrow * : u, N_u$$

$$v \rightarrow u : v, MAC(K_v, N_u | v)$$

2.2.3 Pair-wise 키 생성

두 노드 u와 v는 앞 단계에서 송·수신한 데이터를 바탕으로 두 노드 사이의 pair-wise 키 K_{uv}를 생성한다. 아이디 u가 아이디 v보다 작은 아이디일 경우, 두 노드 u와 v사이에서의 생성되는 Pair-wise 키를 구하는 식은 아래와 같다.

$$K_{uv} = F_v(u) \, (u < v)$$

2.2.4 키 삭제

타이머가 끝나면 센서 네트워크 안의 모든 노드는 Initial 키와 Pair-wise 키 생성 시 사용한 Master 키를 메모리에서 완전히 삭제한다.

III. 구현

3.1 구현 목적

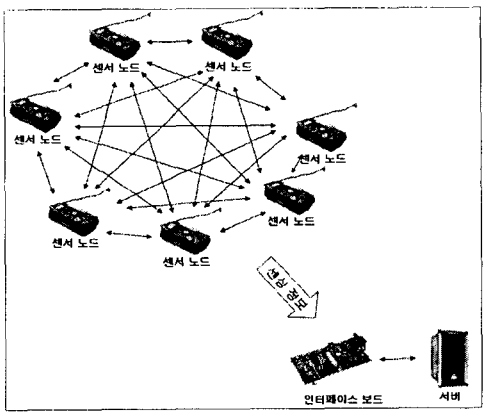
LEAP 프로토콜은 필드 배치 단계에서 Pair-wise키가 생성되기 전에는 공격자로부터 노드 포획 공격을 받지 않는다는 가정 하에 설계되었다. 이러한 초기 안전성을 저하시키는 대신 효율성을 높이는 LEAP 프로토콜의 가정이 현실성이 있는 가정인지 아닌지 구현을 통하여 분석하였다.

3.2 구현 환경

3.2.1 MICA2 소개

MICA 노드는 스마트 더스트(Smart Dust)의 가장 큰 제조업체이자 스마트 더스트 상품과 서비스를 제공하는 업체인 Crossbow Technology 사가 제공하는 저전력 센서 노드이다[6]. 운영체제로 TinyOS를 사용하는 MICA 시리즈는 현재 전 세계적으로 널리 인정받고 있는 센서 하드웨어로 우리는 이러한 MICA 시리즈 중 MICA2 노드를 이용하여 USN을 구성하였다. 우리는 TinyOS의 NesC 프로그래밍 언어[3][4]로 구현한 LEAP 프로토콜을 MICA2 보드에 업로딩 한 후, Pair-wise 키 생성 시간을 측정하였다.

3.2.2 센서 네트워크 구현 설계



<그림 1> 구현한 센서 네트워크 구성도

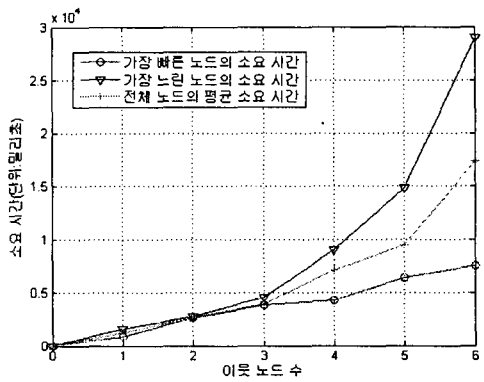
우리는 <그림 1>처럼 7개의 센서 노드, 1개의 인터페이스 노드 그리고 서버 컴퓨터를 사용하여 센서 네트워크 구현 환경을 구성하였다. 7개의 센서 노드가 서로 데이터를 교환하며 키를 생성하는 과정은 서버와 연결된 인터페이스 보드에서 관찰할 수 있다. 이런 구조를 가짐으로써 우리는 키가 생성되는 과정을 서버를 통해 지켜볼 수 있다.

각 센서 노드는 배치 전 전원이 Off 상태였다가 네트워크 시작과 동시에 전원이 On 상태가 되면 통신을 시작한다는 메시지를 브로드캐스트 한다. 서버에는 7개의 타이머가 존재하는데, 인터페이스 보드를 통해 이 메시지를 받은 서버는 신호를 보낸 센서 노드의 타이머를 비로소 작동시킨다. 그리고 7개의 센서 노드들은 LEAP 프로토콜에 따라 서로 Pair-wise 키 생성을 위해 통신을 시작한다. 2.2절의 과정을 거쳐 Pair-wise 키가 생성되면 각 센서 노드는 자신이 가지고 있던 키에 대한 정보를 지우고 자신은 모든 이웃 노드와 Pair-wise 키를 생성했다는 메시지를 브로드캐스트 한다. 인터페이스 보드를 통해 이 메시지를 받은 서버는 해당 센서 노드의 타이머 작동을 중지시키고 해당 노드의 Pair-wise 키 생성 시간을 계산한 후, 기록한다. USN 내의 7개의 센서 노드가 모두 Pair-wise 키를 생성하게 되면 Pair-wise 키의 생성 과정이 끝나게 되고, 우리는 USN 내의 각 노드의 Pair-wise 키 생성 시간을 체크하였다.

IV. 분석

총 10회에 걸쳐 그 생성 시간을 관찰하고 이를 분석하였다.

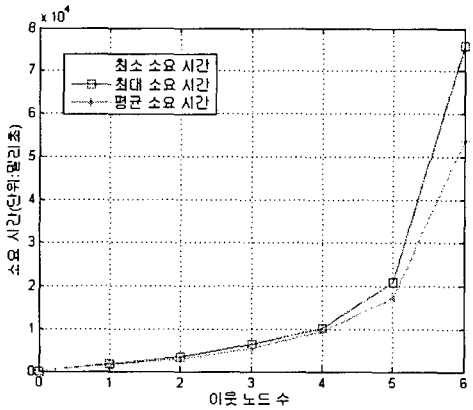
4.1 센서 노드를 이용한 결과 분석



<그림 2> 각 이웃 노드의 Pair-wise 키 생성 시간

Pair-wise 키를 생성하는 데 있어서 모든 노드가 동시에 생성을 완료하는 것은 아니다. 위 <그림 2>는 이웃 노드 수에 따른 Pair-wise 키 생성 시간이 전체 네트워크 중에서 가장 빠른 노드와 가장 느린 노드의 생성 시간 차이를 나타내고 있다. 이 차이는 Pair-wise 키를 형성해야 할 이웃 노드의 수가 많아질수록 더욱 증가한다. 위 <그림 2>를 살펴보면 이웃 노드의 개수가 3개까지는 차이가 뚜렷이 나타나고 있지 않지만 4개부터는 그 차이가 2배에서 많게는 2.6배까지 나오고 있다.

그 이유는 센서 노드가 서로 통신함에 있어서 하나의 채널을 공유하고 있는데, 여러 개의 센서 노드가 서로 통신할 경우, 다른 노드가 그 채널을 점유하고 있다면 그 동안 센서 노드는 통신을 할 수 없기 때문이다. 즉, 이웃 노드의 수가 많아질수록 Pair-wise 키를 형성하는 데 필요한 채널을 확보하기 위한 지연 시간이 길어지게 된다. 그러므로 각 센서 노드가 Pair-wise 키를 형성하는 데 소요되는 생성 시간 차이도 이웃 노드의 수가 늘어날수록 증가하게 된다.



<그림 3> 전체 노드의 Pair-wise 키 생성 시간

위 <그림 3>는 이웃 노드의 수를 1개부터 6개까지 구성함에 따라 전체 USN이 Pair-wise 키를 생성하는 데 소요되는 시간을 10회 측정한 후, 그 결과를 나타낸 것이다.

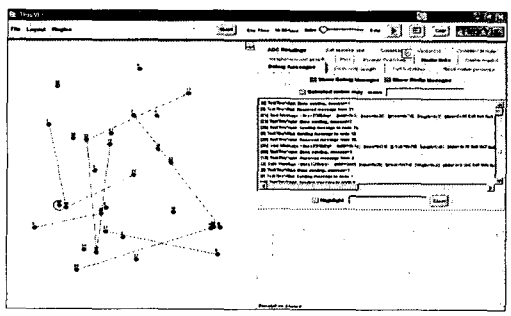
위 그래프는 Pair-wise 키를 맺어야 하는 이웃 노드의 수가 증가할수록 소요되는 시간이 더 많이 요구됨을 나타내고 있다. 이웃 노드의 수가 증가할수록 소요되는 시간이 늘어나는 이유는 <그림 2>에서의 이유와 동일하다.

<그림 2>를 보면 Pair-wise 키를 맺는 이웃 노드가 6개일 때 평균 소요시간이 약 55초가 소요되는 것을 볼 수 있다. 이 정도의 시간은 공격자가 USN이 형성되는 시점에서 공격자의 노드 포획 공격이 없는 이상 공격자로부터 안전하다고 판단된다. 다시 말해 Pair-wise키를 맺어야 하는 이웃 노드의 수가 6개 이하 일 경우에는 LEAP의 가정이 타당하다는 것을 뒷받침하고 있다.

그러나 만일 이보다 더 많은 이웃 노드와 Pair-wise 키를 맺어야 한다면 그 소요되는 시간은 더욱 늘어날 것이다. 그래서 우리는 다음 절에서 시뮬레이션을 이용하여 그 결과를 살펴볼 것이다.

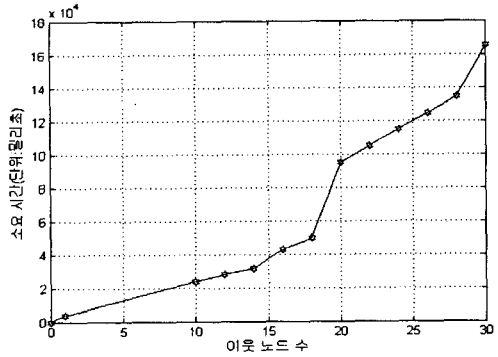
4.2 시뮬레이션 결과 분석

이 논문에서는 실제 구현에 국한하지 않고 TOSSIM(TinyOS Simulator)[5]과 TinyViz[5]를 이용하여 가상 USN을 구성하여 그 결과를 분석하였다. TOSSIM은 TinyOS 코드로부터 직접 컴파일 할 수 있는 시뮬레이터 프로그램이다. 그리고 이 TinyViz 프로그램은 TOSSIM과 함께 쓰일 수 있는 데 이를 이용하여 가상 USN을 시각적으로 구성할 수 있다. 아래 <그림 4>는 우리가 구현한 LEAP 프로토콜을 TinyViz와 TOSSIM을 이용하여 31개의 센서 노드로 가상 센서 네트워크를 구성하여 시뮬레이션 하는 과정을 나타낸 그림이다.



<그림 4> 시뮬레이션 과정

우리는 이러한 시뮬레이션 프로그램을 이용하여 이웃 노드가 10일 때부터 30개까지 2개 단위로 Pair-wise 키 생성 시간을 측정하였다.



<그림 5> 시뮬레이션 결과

위 <그림 5>는 시뮬레이션 측정 결과를 그래프로 나타낸 것이다. 시뮬레이션 결과를 관찰하면 실제 구현한 측정 시간과 차이가 나는 것을 볼 수 있다. 실제 구현에서는 Pair-wise 키를 맺어야할 이웃 노드의 수가 6개일 때 그 소요 시간이 약 55초인데 반해 시뮬레이션 결과에서는 19초 정도 됨을 알 수 있다. 그러나 이러한 차이는 TOSSIM에만 국한된 문제가 아닌 모든 시뮬레이션 도구가 가지고 있는 문제이다. 시뮬레이션 결과가 항상 실제 구현과 같은 절대적 결과를 반영하지 않는다는 것을 감안할 때 우리는 이를 이용하여 상대적 결과를 분석할 수 있다. 먼저 노드가 10개일 때 시뮬레이션 결과는 약 21초, 20개일 때는 약 98초, 30개일 때는 165초 정도가 됨을 살펴볼 수 있다. 이웃 노드의 수가 10개일 때 비해 20개일 때는 약 5배가, 30개일 때는 약 8배가 소요됨을 알 수 있다. 여기서 우리는 Pair-wise 키를 생성하는 데 소요되는 시간이 이웃 노드의 수에 정비례하지 않는다는 사실을 확인할 있다.

LEAP 프로토콜의 초기 안정성을 보장하기 위해서는 Pair-wise 키를 생성하는 데 소요되는 시간이 작아야 한다. 즉, Pair-wise 키를 생성하는 데 소요되는 시간을 줄이기 위해서는 하나의 센서 노드가 Pair-wise 키를 맺어야 하는 이웃 노드의 수를 가능한 적게 해야 한다는 것을 의미한다. 그리고 Pair-wise 키를 형성하는 데 소요되는 시간은 Pair-wise 키를 맺는 이웃 노드의 수에 정비례하지 않는다는 것을 고려하여 적절히 Pair-wise 키를 형성해야 하는 이웃 노드의 수를 결정하여야 한다.

V. 결론

LEAP 프로토콜의 가정의 타당성 여부를 확실히 단정 지을 수 없다. 왜냐하면 Pair-wise 키를 맺어야 하는 이웃 노드의 수가 작을 때에는 안전하지만, 많을 때에는 그 소요 시간이 대폭 늘어나기 때문에 공격자에게 노출될 가능성이 커진다.

우리는 이 논문에서 LEAP 프로토콜의 초기 안정성의 가정이 타당한 지 살펴보았다. LEAP 프로토콜 이외에 많은 프로토콜이 가정을 가지고 시작한다. 앞으로 그러한 다양한 프로토콜들의 가정의 타당성을 검토하고 실제 USN에서 적용가능한지 그 현실성을 살펴보아야 할 것이다.

[참고문헌]

[1] S. Zhu, S. Setia and S. Jajodia. LEAP: Efficient security mechanism for large-scale distributed sensor networks. In 10th ACM Conference on Computer and Communications Security (CCS'03), 2003a.

[2] J. Nah, K. Chae, and K. Chung. Research Trend for Sensor Neetworks Security. In 전자통신동향분석 Vol. 20 No 1. 2005.

[3] David Gay and Philip Levis etc. nesC 1.1 Language Reference Manual. 2003.

[4] Philip Levis and Nelson Lee. TOSSIM : A Simulator for TinyOS Networks. September, 2003.

[5] Philip Levis. "TinyOS Programming". February, 2006.

[6] http://www.xbow.com