주 제

유비쿼터스 센서 네트워크 보안

세종대학교 인터넷학과 **임채훈**

차 례

요 약

무선 네트워킹과 소형화 기술의 발전은 고정된 기반시설이 없는 열악한 환경에서도 초소형의 센서 노드들을 무작위 배치하여 애드혹 네트워킹을 통해 순간적으로 네트워크 인프라를 구축하고 긴밀한 협동작업을 통해 복잡한 환경감지를 가능하게 하는 센서 네트워크의 개발을 이루어냈다. 각 센서 노드들은 매우 제한적인 능력을 가지고 있지만 이들의 협동작업은 마치 미세한 신경세포 조직이 상호작용하여 동물의 신체활동을 가능하게 하듯이 기존의 네트워킹이나 컴퓨팅에서는 불가능했던 많은 응용들을 가능하게 하여 우리의 생활환경을 스마트한, 지능적인 환경으로 만들어 줄 수 있다. 본 고에서는 이러한 센서 네트워크의 기존 네트워크와의 차별화되는 특징 및 응용에 따른 다양한 보안이슈와 최신 보안기술 연구동향들을 살펴보기로 한다.

I. 센서와 센서 네트워크

하드웨어와 무선 통신/네트워킹 기술의 발전은 저비용, 저전력, 다기능의 소형 센서의 개발을 가능하게 하였다. 이러한 센서들이 수천, 수만개씩 특정 지역에 흩뿌려져 애드 호크 센서 노드(Ad hoc sensor nodes)를 형성하고, 이들의 자율적인 상호작용으로 센서 네트워크를 구성하여 할당된 작업을 수행하게 된다. 센서 네트워크는 데이터를 수집/센싱, 처리/분석하고, 또한 통신함으로써 정보를 언제, 어느 곳에서든 접근 할 수 있도록 하여 스마트 환경을 만드는

것을 가능하게 한다[1,30].

센서, 데이터 처리, 통신 요소들로 이루어진 초소형의 센서 노드들은 매우 제한된 컴퓨팅 자원만을 가지며 아주 짧은 거리의 무선통신만이 가능하다. 이들은 대상지역에 대량으로, 매우 밀집되게, 임의로 흩뿌려져 자율적인 센서 네트워크를 구성하며, 소수의 센서 노드들이 동작을 멈추더라도 스스로 네트워크를 재구성하여 본연의 임무를 수행할 수 있어야 한다. 센서 노드들의 위치는 미리 결정되거나 조절될 필요 없이 무작위로 살포하여 접근하기 어려운 지역의 환경감지나 재난 구조작업 등을 가능하게 한다. 센서 노드들은 또한 매우 제한된 전력 및 통신범위로 인해 대부분 인접한 주위의 소수 노드들과만 통신이 가능하므로 센싱결과를 중앙의 제어 노드로 전송하기 위해서는 다수의 노드를 거쳐 목적지까지 도달하는 멀티홉 라우팅(multihop routing)이 기본이 된다.

이러한 센서 네트워크의 독특한 기능/특징들은 센서 네트워크의 매우 다양한 응용을 가능하게 한다. 예를 들어 군사 목적으로는 신속한 배치, 자가 조직화, 그리고 센서 네트워크의 결함 허용 등의 특징들을 이용하여 전장이나 군사작전 등에서 적군의 움직임을 감시하거나 활동하기에 안전한 장소를 탐색하거나 피해사항이나 사상자를 측정하는 등의 지능적인 임무를 수행할 수 있다. 사람의 접근이 어려운 지역이라면 비행기나 대포 등을 통해 쉽게 네트워크를 구축할 수 있다. 재난현장에서는 신속히 임시 네트워크를 구축하여 사상자들의 위치를 찾아낼 수도 있다. 지진이나, 저수지, 또는 화산 주위의 상태 등을 관찰할 수도 있으며, 노인이나 장애자들의 건강상태, 공항이나 경기장에서의 생물학적 공격과 같은 상황파악을 위해 항상 켜져 있는 상태로 동작할 수도 있다.

센서 네트워크는 센서 필드 내에 산재된 다수의 Sensor node들과 외부 네트워크와의 게이트웨이 역할을 하는 소수의 Sink node (혹은 Base station), 그리고 외부 네트워크(유/무선 인터넷, 위성)에 연결된 Manager node (control center)로 구성된다. 센서 필드 내에 흩어져 있는 각각의 센서 노드들은 주기적으로 혹은 관리센터로 부터의 지시에 따라 데이터를 수집하고 부분적으로 가공하여 Sink node로 전송한다. 싱크 노드들은 인터넷이나 위성 등 기존의 통신 인프라를 통하여 관리센터와 통신을 하게 된다.

규모가 큰 센서 네트워크의 경우는 센서 노드들이 계층구조를 가질 수도 있다. 대부분의 센서 네트워크는 그 효율성으로 인해 계층적 클러스터 구조(hierarchical clustering architecture)를 많이 취한다. 즉 일정 범위의 인접 노드들로 클러스터(cluster)를 형성하고 그 중 한 노드가 클러스터 헤드(Cluster head)의 역할을 하여 클러스터 헤드들간의 통신을 통해 싱크 노드와 통신을 하는 것이다. 이 경우 클러스터 헤드들이 데이터 융합 노드(aggregation point)가 되어 클러스터 멤버들로부터 수집된 데이터를 가공하여 압축된 결과를 싱크 노드로 전송하여 효율성을 높일 수 있다.

1.1 센서 네트워크 플랫폼

센서 네트워크는 밀리미터 크기의 초소형 센서로부터 PDA나 랩탑 크기의 강력한 센서 노드에 이르기까지 응용에 따라서 다양한 종류의 디바이스들이 사용되지만, 가장 큰 관심의 대상은 센서 네트워크의 최하위에서 신경세포 역할을 하는 소형 센서 노드들이다.

센서 네트워크의 하드웨어 및 소프트웨어 플랫폼의 개발을 선도하고 있는 곳은 캘리포니아 버클리대학(University of California, Berkeley)이다. Smart dust 프로젝트 통해 밀리미터 크기의 초소형 센서를 위한 기본 아키텍처를 확립하고 이를 기반으로 다양

한 후속 프로젝트를 통해 소형 센서 노드의 하드웨어 플랫폼 (통상 Berkeley mote로 불림: http://www.jlhlabs.com/), 센서 노드에 최적화된 운영체제 (TinyOS: http://www.tinyos.net/) 및 분산 데이터베이스 응용 소프트웨어(TinyDB: http://telegraph.cs.berkeley.edu/tinydb/) 등을 개발하여 거의 대부분의 센서 네트워크의 기본 플랫폼으로 사용되고 있다[12,13,28].

초소형 센서의 대표적인 예는 최근에 UCB에서 개발된 Spec으로 단순한 기능의 초저가, 초소형 센서를 목표로 대부분의 기능들을 2 mm x 2.5 mm 크기의 하나의 칩으로 구현한 것이다(전원이나 안테나, inductor, crystal, oscillator 등은 외부에서 공급). 이는 기존의 모듈형태의 Mica 모트를 하나의 칩으로 구현한 것으로 향후의 초저가 무선 센서 노드의 전형이 될 것으로 보인다.

연구개발이나 시험용으로 가장 널리 사용되는 센서 노드는 버클리 모트, Mica mote로 대표적인 Mica mote는 250개 이상의 회사나 연구조직에 배포되어 사용되고 있다. Mica 모트는 칩 형태가 아닌 COTS(Commercial-Off-The-Self) 부품들로 조립된 모듈/보드형태이며 크기나 비용이 절대적이지 않은 대부분의 응용에 가장 적합한 형태이다. 1.25 x 2.25 inch 크기(AA 배터리 두개 정도의 크기)의 모듈로 TinyOS를 탑재하고 있으며, Mica2가 가장 널리 사용되는 Mica 플랫폼, mote이다. 모트의 mote들(Mica, Mica2, MicaZ)은 모두 Atmel의 Atmega128 MCU (8-bit CPU, up to 16MHz, 4KB RAM, 128KB flash memory)을 채택하였으나 서로 다른 RF radio 모듈을 갖추고 있다 (Mica: RF Monolithics사의 TR1000, Mica2: Chipcon사의 CC1000, MicaZ: Chipcon사의 CC2420).

역시 UCB에서 가장 최근에 개발된 Telos 플랫폼은 Mica 플랫폼에 비해 보다 나은 성능을 가지면서도 전력 소모는 1/10 정도로 줄여 테스트베드용의 새로운 플랫폼으로 자리잡을 전망이다. Telos 플랫폼에서 채택한 MCU는 TI의 MSP 430(16-bit RISC CPU, 2KB RAM, 60KB flash)으로 이는 Atmega128에 비해 전력소모가 훨씬 낮아 새로운 센서 노드의 설계에 가장 인기있는 MCU로 부상하고 있다. 또한 Telos 플랫폼에서 채택한 CC2420 RF radio는 IEEE 802.15.4 표준을 준수하는 2.4GHZ 대역의 radio로 250Kbps의 높은 전송속도를 제공하여 향후 대부분의 중하급 센서 노드의 설계에서 널리 채택될 것으로 전망된다.

II. 센서 네트워크의 보안 이슈

센서 네트워크는 주변의 환경정보를 수집/분석하는 툴로서 매우 민감한 응용들에서 사용된다. 기존의 모든 다른 네트워크에 비해 센서 네트워크만의 매우 독특한 특성은 그 위협요소나 공격방법, 그리고 이를 위한 보안대책의 면에서도 완전히 새로운 탐구가 필요하다.

보안상의 관점에서 가장 이슈가 되는 센서 네트워크의 주요 특징으로는 센서 노드의 제한된 능력, 센서 노드들에 대한 물리적 보안의 취약성, 그리고 브로드캐스팅을 주 통신 수단으로 하는 멀티홉 라우팅 및 데이터 융합(data aggregation) 등을 들 수 있다. 센서 네트워크의 실효성을 위해서는 센서들이 매우 제한된 연산, 통신, 저장능력 및 에너지원만을 가질 수 있어 정상적인 동작을 위한 프로토콜 뿐만 아니라 보안 기능의 사용에도 많은 제약이 따른다. 또한 센서 네트워크는 사람의 접근이 어려운 지역에 설치되어 장기간 방치된 상태로 운영되므로 물리적인 공격에 매우 취약하다. 센서의 제한된 전력과 통신능력은 인접 노드로의 제한된 브로드캐스팅을 주요 통

신방식으로 사용하도록 하여 일대일 통신에 비해 훨씬 많은 취약성과 보안상의 어려움을 가중시키고, 특히 전력이나 대역폭의 효율적인 사용을 위해 중간 노드들이 경유 메시지에 대한 부분적인 프로세싱을 해야 하는 특성은 보안을 더욱 더 어렵게 만든다.

이장에서는 센서 네트워크의 주요 보안 이슈를 간략히 살펴본다[5,14,26,32]. 라우팅 보안이나 키 관리 및 보안 프로토콜 등은 후속되는 장에서 좀 더 상세히 다룰 것이다.

- Node compromise : 센서 네트워크는 수천 수만개의 소형 센서들이 넓은 지역에 흩어져 설치되므로 운영자가 각 센서 노드들을 관리하고 감시하는 것은 불가능하다. 따라서 공격자는 쉽게 센서 노드에 물리적으로 접근하여 비밀키나 중요 데이타를 추출해 낼 수 있으며, 프로그램을 수정하여 네트워크에 재투입하거나 보다 강력한 노드로 교체시켜 공격에 이용할 수 있다. 하드웨어적으로 tamper-resistant한 센서 노드를 값싸게 만드는 것은 매우 어려워 현실성이 떨어진다. 따라서 센서 네트워크를 설계할 때 소수의 악의적인 노드가 존재하더라도 전체 네트워크가 안정적으로 동작하도록 resilient network를 구축하는 것이 필요하다.
- Privacy : 센서 네트워크는 쉽게 악의적인 목적의 감시 네트워크로 악용될 수 있다. 은밀한 센서 네트워크를 통해 직원들의 동태를 감시하거나 백화점의 고객들을 감시할 수도 있고, 정보기관의 경우 각종 공중 장소에 감시 네트워크를 설치하여 시민의 프라이버시를 침해할 수도 있다. 기존의 CCTV 카메라가 유사한 목적으로 합법적으로 사용되고 있으나 센서 네트워크의 경우는 쉽게 이를 감출 수 있고, 또한 완전 자동화된 방법으로 원격에서 대량의 정보 수집 및 가공이 용이하다는 사실이 문제를 더욱 악화시키는 것이다. 또한 합법적으로 설치/운영되는 센서 네트워크라 하더라도 여기서 수집된 정보들이 불법적으로 악용될 수 있는 소지는 얼마든지 존재한다. 불법적인 센서들을 찾아내는 센서 탐지 장치(sensor detector)의 대량 보급이 한 방어책이 될 수 있다. 그러나 기술적인 방법만으로 프라이버시 문제를 해결하기는 어려우므로 사회규범이나 법/제도적인 장치의 확립이 선행되어야 한다. 센서 노드의 존재를 인지시키고 수집된 정보의 사용 목적을 명시하여 거부감을 줄여 주는 노력도 필요할 것이다.
- Secrecy & Authentication : 민감한 센싱 데이터의 도청이나 프로토콜 메시지의 조작 등 일반적인 통신 보안을 위해 다양한 암호학적인 보안 기법들이 사용될 수 있다. 센서 네트워크의 경우 데이터 융합, end-to-end security는 필요상 종단간 보안(end-to-end security)은 대부분의 경우 불가능하므로 링크계층 보안 프로토콜, Berkeley mote이 가장 일반적인 보안대책이 될 것이다. 대표적인 프로토콜로 버클리 모트에서 채택한 TinySec을 들 수 있다 (5장에서 좀 더 상세히 다룰 것임). 여기서 가장 중요하며 또한 어려운 문제는 센서 네트워크의 제한된 자원을 고려하여 계산량이나 통신량에 있어서 얼마나 효율적으로 보안기능을 추가하느냐이다.
- Routing security : 센서 네트워크의 동작에 가장 근간이 되는 것이 라우팅 프로토콜이다. 기존의 애드혹 라우팅 프로토콜들은 대부분 센서 네트워크에 사용되기에 너무 무겁거나 또한 센서 네트워크의 특성상 사용이 불가능하다. 센서 네트워크의 라우팅 프로토콜에 대한 공격방법 및 대응방안들은 3장에서 좀 더 상세히 다룬다.
- Key management : 센서 네트워크는 전혀 혹은 거의 네트워크 인프라가 없는 상태에서 무작

위로 배포된 센서들이 라우터의 역할을 겸하여 안전한 네트워크 인프라를 구축하여야 한다. 여기서 가장 중요한 것이 모든 다른 보안목적에서 사용될 비밀키를 설정하고 관리하는 것이다. 센서 네트워크의 다양한 특성들은 특히 키 관리 문제에 중대한 도전이 되어 키 관리 문제는 센서 네트워크의 보안에서 가장 어려우며 또한 중요한 보안의 출발점이 된다. 4/5장에서 키 관리 문제 및 보안 프로토콜들을 상세히 다룬다.

Secure data aggregation : 센서 네트워크의 센싱 데이터는 센서의 밀접한 배치로 인한 거리의 인접성에 의해 많은 부분 중복이나 불필요한 부분이 존재하여 원래 데이터(raw data)를 그대로 전송하면 귀중한 에너지나 대역폭 등 자원을 낭비하게 된다. 따라서 일부의 중간 노드들(cluster head, sink)이 데이터를 취합하여 중복을 제거하고 특징적인 데이터만을 추출하여 압축된 형태로 전송하게 된다. 문제는 이러한 데이터 융합 노드들이 공격의 주요 목표가 되고 이들이 공격자의 수중에 들어간다면 질의(query)를 무시하거나 거짓의 위조된 융합 결과를 보고하여 센서 네트워크의 기능을 심각하게 훼손시킬 수 있다는 것이다. 이는 라우팅 공격이나 서비스 거부 공격 등에 대한 보안과는 다른 데이터 융합 결과 정확성을 보장할 수 있는 보안대책이 필요하다[21,29,31].

서비스 거부 공격(Denial of Service attack): 센서 네트워크는 매우 제한된 자원만을 갖는 수많은 소형 센서들로 순간적으로 형성되는 네트워크이다. 따라서 기반구조 자체가 매우 취약하며 물리적인 공격에도 무방비 상태이므로 다양한 형태의 서비스 거부(DoS : Denial of Service) 공격이 가능하다. 열악한 환경에서 동작하는 센서 네트워크는 일부 오류가 발생하더라도 지속적으로 동작하도록 설계되지만 이들은 지능적이고 결정적인 공격자들에 대해서는 거의 제 기능을 발휘하지 못한다. 특히 센서의 수명은 곧 전원의 수명과 동일하므로 다양한 방법으로 센서의 전원을 고갈시키는 DoS 공격은 가장 막기 어려우며 치명적이다. 센서 네트워크에 대한 DoS 공격은 다양한 계층에서 이루어질 수 있다. 물리 계층에서의 전파방해(jamming)나 센서 노드의 물리적 파괴를 필두로 링크 계층 및 네트워크 계층에서의 다양한 자원고갈 공격들이 가능하다[34].

III. 라우팅 보안

센서 네트워크의 라우팅 프로토콜이 다양한 공격에 취약한 것은 이들이 기존의 다른 네트워크들과는 달리 매우 제한된 자원을 갖는 노드들이 무선환경에서 상호협력을 통해 인프라를 구축하고 멀티홉 라우팅을, routing과 forwarding을 기반으로 한다는 것이다. 대부분의 라우팅 프로토콜들이 각 노드들이 상호 협력하여 라우팅과 전달(forwarding)을 올바르게 수행한다는 가정을 기반으로 하지만, 공격자에 의해 제어되는 노드나 과부하로 제기능을 못하는 노드, 이기적인 노드(selfish node : 자기가 관심이 있는 패킷을 제외하고는 자원을 소모하지 않으려는 노드), 고장난 노드(배터리 고갈이나 물리적인 피해 등으로 인한) 등 비정상적인 노드들이 존재하기 마련이며, 이러한 소수의 misbehaving node들이 존재하더라도 전체 네트워크 인프라가 크게 훼손되지 않고 유지되도록 하는 것이 안전한 라우팅 프로토콜에서 추구해야 하는 것이다.

센서 네트워크 라우팅 프로토콜은 매우 단순하므로 기존의 애드혹 네트워크에서의 다양한 공격들에

쉽게 당할 수 있다. 라우팅 정보의 위/변조나 재전송, 데이터의 선택적인 전달(selective forwarding), 공격자가 목표로 하는 데이터 흐름 경로상에 자신을 효율적으로 포함시키기 위해 사용할 수 있는 Sink hole/wormhole/Sybil attack이나 HELLO flooding 등 [15].

센서 네트워크에 대한 공격의 유형을 크게 내부공격(insider attack)과 외부공격(outsider attack)으로 나누어 생각할 때, 라우팅 프로토콜에 대한 대부분의 외부공격은 링크계층에서의 암호화 및 인증 메커니즘을 이용하면 쉽게 막을 수 있다. 링크계층의 암호화 및 인증 기능에 의해 공격자가 합법적으로 네트워크 토폴로지에 참여하여 라우팅 프로토콜에 관여하는 것이 불가능하므로 라우팅 정보의 조작이나 선택적인 메시지 전달, sybil attack, sinkhole attack 등은 모두 불가능하다. 그러나 링크계층 보안 메커니즘이 wormhole이나 Hello flooding을 막을 수는 없다. 공격자가 네트워크에 가입하는 것은 불가능하지만, wormhole을 이용하여 네트워크의 한 부분에서 합법적인 노드가 전송한 패킷을 네트워크의 다른 부분으로 터널링시켜 그들이 인접 노드인 것처럼 만들거나, 엿들은 패킷을 충분한 전력으로 전체 네트워크에 재방송하는 Hello flooding은 항상 가능하기 때문이다.

한편 전복된 노드(compromised node)를 이용한 내부에서의 공격은 대부분의 경우 훨씬 막기가 어렵다. 이러한 공격자는 합법적인 노드로 네트워크 동작에 참여할 수 있으므로 위에서 언급된 거의 대부분의 공격이 가능하다. 센서 네트워크가 공격자의 물리적인 접근에 거의 무방비 상태이며, 또한 센서 노드 자체의 물리적 보안 역시 비용을 생각하면 거의 불가능에 가까우므로 이러한 공격은 항상 가능하다고 가정해야 한다.

노드들간의 상호 협력을 기반으로 하는 라우팅 프로토콜에서 가장 문제가 되는 것이 공격자에 의해 전복된 노드나 이기적인 노드를 포함한 비정상적인 노드들이다. 위에서 살펴보았듯이 암호기술을 이용한 보안만으로는 모든 공격을 막는 것이 어려우므로 이러한 비정상적인 노드들을 탐지하여 라우팅 경로에서 배제시키거나 불이익을 당하게 하는 방안들이 유용할 수 있다. 이런 목적으로 모바일 애드 혹 네트워크를 위한 다양한 침입탐지 시스템들이 제안되었으며[22], 또한 비정상적인 노드를 탐지하여 경로에서 배제시키는 Watchdog-Pathrater[21], 다양한 Reputation mechanism(CORE, CONFIDANT 등)을 통해 비정상적인 노드를 탐지하여 아예 네트워크에서 격리시킴으로써 불이익을 당하게 하는 방안[33] 등이 제안되고 있다. 이러한 방안들은 그러나 센서 네트워크와 같은 보다 제약된 자원을 갖는 환경에서는 사용이 쉽지 않다.

IV. 키 관리

센서 네트워크의 보안에서 가장 어려운 부분이 키 관리다. 키 관리 프로토콜은 전혀 인프라나 신뢰할 수 있는 센터가 없는 상태에서 임의로 설치된 센서 노드들간에 신뢰관계를 설정하여 안전한 통신 인프라를 구축해 주고 이 후의 다양한 보안 프로토콜에서 필요한 비밀키들을 생성해 주는 역할을 한다. 키 분배를 위해서는 각 센서들이 어떤 형태로든 비밀정보가 탑재되어 설치되어야 하며, 설치 후에는 이 비밀정보를 이용하여 그들간의 상대적인 위치를 파악한 후 키 분배 및 통신설정을 하게 된다. 우선 키 분배와 관련된 센서 네트워크의 한계 및 이에 따른 요구조건들을 살펴보자:

- Node capture : 센서 노드들은 공공장소나 적대적인 환경에 설치될 수 있고 또한 관리되지 않

는 상태로 운영되므로 공격자에 의한 물리적 접근 및 공격에 취약하다. 센서 노드 내에 저장된 비밀키는 센서가 물리적으로 안전한 메모리를 갖추지 않는 한 물리적인 공격에 의한 노출은 피할 수 없다. 따라서 일부 센서 노드내의 비밀정보가 노출되더라도 해당 노드나 그 주위의 노드들을 제외하고는 전체 네트워크의 안전성이 유지되어야 한다 (damage localization).

- Knowledge of network configuration : 센서 노드들은 통상 무작위로 설치되므로 설치 후의 네트워크 구조에 대한 정보(어떤 노드들이 서로의 무선 통신범위 내에 있는지에 대한 정보)를 미리 알 수는 없다. 비록 수작업으로 하나씩 설치한다고 하더라도 수많은 노드들의 개별적인 위치를 사전에 결정하는 것은 거의 불가능하다. 따라서 설치 후의 네트워크 토폴로지에 대한 사전 지식을 바탕으로 하는 키 관리 방식은 바람직하지 않다.
- Node constraint : 센서 노드들은 매우 제한된 계산력, 전송전력 및 대역폭을 가지므로 프로토콜에 필요한 통신량이나 저장량을 최소화 하는 것이 필요하다. 특히 저가의 소형 센서들에서 계산량이나 통신량이 많이 소요되는 공개키 암호를 자유로이 이용하는 것은 아직까지는 현실적으로 불가능에 가깝다.
- Dynamic topology change : 센서 노드들은 초기 설치 후 다양한 이유로 추가 설치하거나 네트워크에서 제외시켜야 할 필요가 있다. 따라서 노드가 추가 설치되더라도 기존의 노드들과 키 설정이 가능하여야 하며, 또한 비정상적인 행위를 하는 노드가 탐지되면 다이내믹하게 이를 네트워크로부터 제거할 수 있어야 한다.

위와 같은 센서 네트워크의 제한이나 요구사항에서 알 수 있듯이 기존의 대부분의 키 관리 프로토콜들은 센서 네트워크에서 사용이 어렵거나 불가능하다. 이 장에서는 센서 네트워크 키 관리 를 위해 제안된 대표적인 접근방법들을 간략히 살펴본다.

4.1 Key infection

Anderson 등의 Key infection은 센서 네트워크 환경의 보다 현실적인 공격모델을 바탕으로 대부분의 민감하지 않은 응용에서 효율적으로 사용할 수 있는 키 관리 방안으로 비밀키를 평문상태로 교환한다 [2]. 즉 공격자가 최소한 센서 네트워크의 초기 설치 시 키 교환 과정 동안만은 네트워크의 전 영역을 동시에 도청하는 것이 어렵다고 가정한다면 센서 노드의 초기화시 각 노드들이 랜덤하게 생성한 세션키를 이웃 노드들과 평문상태로 교환하더라도 대부분의 세션키들은 안전할 수 있다는 것이다 (물론 네트워크의 정상 운영 중에는 공격자에 대한 아무런 제한이 없다).

이 방법은 얼핏 보기에 역설같지만 실제로 그렇게 중요하거나 민감하지 않은 응용 분야의 경우 안전성과 효율성 사이의 하나의 타협이 될 수도 있다. 센서 네트워크가 넓은 지역에 대규모로 설치되는 경우 네트워크의 초기 설치 중의 짧은 시간동안 공격자가 도청할 수 있는 트래픽은 전체 네트워크 트래픽의 아주 일부에 지나지 않을 것이기 때문이다. 물론 공격자가 일부의 소수 노드들만이라도 전복시켜 자신의 통제하에 두는 경우 앞에서 살펴본 다양한 종류의 라우팅 공격이나 DoS 공격이 가능하므로 이러한 위험부담은 감수를 해야 할 것이다.

4.2 Network-wide shared key

가장 간단한 키 관리 방법으로 네트워크의 모든 노

드들이 동일한 비밀키를 갖도록 하는 방법을 생각할 수 있다. 설치 전에 모든 센서 노드에 동일한 비밀키를 주입하여 설치 후에는 이 비밀키를 이용하여 암호화 및 인증을 수행하는 것이다. 이는 효율성 면에서는 월등하지만 센서 노드가 물리적으로 안전하지 않은 한 하나의 노드만이라도 공격자에 의해 전복된다면 전체 네트워크의 안전성이 파괴될 수 있다는 문제점이 있다. 비록 하나의 공유키를 이용하는 것이 문제점은 있으나 키 관리의 효율성과 단순성으로 인해 많은 기존 네트워크 프로토콜들에서 이 방식을 가정하고 있다 (Pebblenet[3], TinySec[16] 등). 전체 네트워크에서 공통의 비밀키를 이용하는 경우 공격당한 노드가 탐지되거나 그렇지 않더라도 주기적으로 이 키를 갱신시키는 re-keying protocol은 필수적이다.

한편 미리 주입된 마스터키를 사용하여 링크키를 교환한 후 마스터키를 메모리에서 지움으로써 위 방식의 문제점을 상당부분 제거할 수 있다. 즉 센서 노드의 초기 설치시 인접 노드들과의 인증 및 링크키 교환 과정 동안만 마스터키가 노출되지 않는다고 가정한다면 하나의 마스터키만을 이용하여 안전하고 효율적인 키 관리가 가능하다. LEAP는 이러한 가정을 기반으로 다양한 종류의 키(node key, pairwise shared key, cluster key, group key)들에 대한 관리방법들을 다루고 있다[34]. 그러나 최근의 연구결과에 따르면 표준적으로 사용되는 Mica2 mote의 경우 1분 이내에 물리적 해킹이 가능하다고 하므로 이러한 접근방법은 잠재적인 위협에 여전히 취약하다고 할 수 있다[11].

4.3 Base station-node pairwise key

한편 대부분의 센서 네트워크가 기존의 네트워크와의 게이트웨이 역할을 하는 베이스 스테이션을 가지고 있으므로 이 베이스 스테이션을 신뢰할 수 있는 센터로 활용하여 노드들간의 링크키를 공유하는 kerberos와 유사한 프로토콜을 구성할 수도 있다. 이를 위해 각 센서 노드는 베이스 스테이션과의 공유키를 미리 주입한 상태로 설치되고 이 노드키를 이용하여 베이스 스테이션과의 안전한 통신을 통해 다른 노드와 링크키를 공유하는 것이다. SPINS가 이런 방식의 키 분배를 이용하는 대표적인 예이다[27]. 이 방식은 네트워크 전체에 걸쳐 하나의 비밀키만을 이용하는 방식에 비해서는 훨씬 높은 안전성을 제공하지만 베이스 스테이션이 single point of failure로 작용하고 공격자의 주요 공격목표가 될 수 있으며, 또한 베이스 스테이션을 통해 노드간의 링크키를 설정하고 관리하는데 상당한 네트워크 자원이 소모된다는 면에서 한계를 지닌다.

4.4 Random key predistribution

센서 네트워크의 제한된 자원을 고려할 때 가장 합리적인 방안으로 널리 연구되고 있는 것이 Random key predistribution이다[4,7].

각 센서 노드들에는 미리 랜덤하게 생성된 키 풀 (key pool) S로부터 임의로 선택한 m개의 키(key ring)를 저장시킨다. 키 풀 S의 크기는 S에서 임의로 크기 m의 두 부분집합을 선택하였을 때 그 둘 사이에 최소한 하나의 공통된 키가 포함될 확률이 p가 되도록 선택한다 (확률 p는 센서 네트워크의 크기와 노드들의 평균 밀도에 의해 결정된다. 구체적으로 센서 네트워크의 노드의 총 수를 n, 한 노드의 통신 범위 내에 있는 이웃 노드들의 평균 수를 n' 이라 하면 p=n' /n으로 주어진다). 모든 노드들이 m개의 키로 구성된 키 링을 저장한 상태로 설치되면 각 노드들은 node discovery를 통해 이웃 노드와의 공유키를 찾아 서로를 인증함으로써 링크키가 설정된다. 만일 이

웃 노드들간의 공유키가 존재하지 않는다면 다른 연결된 이웃 노드들을 경유하여 그들간의 링크키를 설정할 수 있다.

이 방식은 확률적으로 동작하기 때문에 키설정이 완료된 후라고 하더라도 전체 네트워크가 완전히 연결된 상태가 되지 않을 수도 있다. 이 경우는 전송전력을 증가시켜 범위를 확장해 가면서 키설정을 완성해 나가는 방법을 이용한다.

4.5 Random pairwise key

Random key predistribution의 단점은 노드간의 인증이 불가능하며, 또한 소수 노드의 전복에 매우 취약하다는 점이다. 즉 각 노드들이 공통의 키 풀로부터 선택된 부분집합에서 서로간의 공유키를 찾는 것이므로 같은 키가 복수개의 링크에서 사용될 수 있으므로 노드간 인증이 불가능할 수 있고, 또한 일정 수 이상의 노드들이 공격자의 수중에 떨어지면 키 풀이 노출되어 네트워크 전체의 안전성이 위협받을 수 있다. Random pairwise key 방식은 각 노드들간에 유일한 키를 할당하는 방식으로 노드간의 인증을 가능하게 하며, 노드 전복의 영향을 해당 노드에게만 국한시켜 전체 네트워크의 안전성에 미치는 영향을 최소화시킬 수 있다. Random key predistribution과 유사하게 설계하거나 Blom의 방식을 확장하여 설계하는 방법 등이 연구되고 있다[6,20].

4.6 Public Key Technique

공개키 암호는 센서 노드와 같은 극히 제약된 환경에서는 거의 사용이 불가능한 것으로 생각되어 왔으나, 최근들어 공개키 암호의 센서 노드에의 적용 가능성에 대한 희망적인 연구결과들이 속속 발표되고 있다[9,22,33]. 예를들어 Gunnar Gaubatz 등은 적절한 알고리즘과 파라미터 선택 및 최적화된 구현을 통해 센서 네트워크에서 하드웨어 기반의 공개키 암호의 사용 가능성을 타진하고 있다[9]. 실제로 Rabin 암호와 NTRU의 ASIC 구현 결과를 제시하고 있는데, 특히 NTRU의 경우 gate 수 약 3000 정도와 평균 전력 소모량 20μW 정도를 사용하여 4.5Kbps 정도의 속도를 얻을 수 있음을 보고하고 있다. 또한 David Malan 등은 7.4MHz Mica2 플랫폼에서 PKI의 사용에 필요한 공개키 암호 (이산대수문제기반의 암호 및 타원곡선 암호)의 구현한 결과를 보고하고 있다[22]. 타원곡선 암호의 멱승(point multiplication)을 수행하는데 1KB의 RAM (data memory)과 34KB의 ROM (code memory)를 사용하여 약 34초 정도가 소요되어 Mica2 플랫폼에서 공개키 암호의 사용이 가능할 수 있음을 시사하고 있다.

센서 노드에서 공개키 연산은 주로 키 분배를 위한 것으로 매우 드물게 일어나는 연산이므로 위의 실험결과들은 향후 보다 적극적인 연구가 이루어진다면 상당한 가능성이 있음을 시사해 준다. 물론 공개키 암호의 사용을 위해서는 공개키 기반구조가 확립되어야 하므로 여전히 넘어야 할 산은 높다고 할 수 있다.

V. 보안 프로토콜

키 관리를 포함하여 인증이나 암호화 등 다양한 보안기능을 제공하는 대표적인 보안 프로토콜로 Berkeley mica mote의 링크계층 보안 아키텍처로 구현되어 사용되고 있는 TinySec[16]과 라우팅 프로토콜의 보안용으로 널리 사용되는 SPINS[27]에 대해 간략히 살펴본다.

5.1 TinySec

앞에서 살펴보았듯이 센서 네트워크에 대한 다양한 공격들은 많은 부분 링크계층의 인증 및 암호화를 통해 해결이 가능하다. 센서 네트워크의 데이터 융합의 필요성을 감안하면 종단간 보안은 거의 불가능하며 (이는 또한 최종 수신자만이 패킷의 진위 여부를 파악할 수 있어 중간 노드들의 에너지와 대역폭을 고갈시키는 DoS 공격에 취약하다) 응용에 투명한 최소한의 보안기능을 제공할 수 있는 최적의 계층은 링크계층이라 할 수 있다.

TinySec은 버클리 Mica 모트의 링크계층 보안 아키텍쳐로 설계되어 실제 공식적인 TinyOS release에 포함되어 있는 링크계층 보안 프로토콜이다. 제공되는 보안서비스는 인증과 암호화이며 (TinySec-Auth : authentication only, TinySec-AE: authenticated encryption) 패킷 포맷의 정의 및 응용과의 인터페이스를 포함하여 상세한 성능분석까지 제공하고 있다. 인증기능은 대부분의 공격에 대해 필수적이므로 디폴트로 사용되며 암호화는 선택적으로 사용하도록 하였다. 암호화 및 인증은 블럭암호 Skipjack과 RC5를 기본으로 사용하여 CBC 모드의 암/복호화 및 CBC-MAC을 구현하였다.

TinySec은 특히 보안기능의 추가로 인한 메시지 팽창을 최소화하기 위해 많은 노력을 기울였다. TinyOS의 패킷 길이가 30바이트 전후이며, 또한 통신시의 전력소모를 감안하면 몇 바이트의 오버헤드도 상당한 부담이 될 수 있기 때문이다. 원래의 TinyOS 패킷에 비해 인증만 사용할 때는 3바이트, 인증과 암호화를 동시에 사용하더라도 5바이트 정도의 메시지 팽창이 있을 뿐이다. TinySec은 특별한 키 관리 프로토콜이 없이 하나의 고정된 비밀키를 가정하고 있어 이에 대한 후속 연구도 활발히 진행되고 있다(ECDH[22], TinyPK[33], TinyKeyMan(http://discovery.csc.ncsu.edu/~pning/software/TinyKeyMan/) 등).

TinySec은 nesC(TinyOS를 위해 UCB와 Intel에 의해 개발된 C언어의 모트용 확장판) 3000라인 정도로 구현되었는데 이는 728바이트의 RAM과 7146바이트의 코드 스페이스가 요구된다고 한다. TinySec 개발자들은 이 구현을 바탕으로 Mica2 모트(8MHz 8-bit Atmega128L MCU: 128KB instruction memory, 4KB data RAM, 512KB flash memory, Chipcon CC1000 radio: up to 19.2Kbps)를 이용하여 다양한 성능 테스트(throughput, energy, latency 등)를 통해 소프트웨어로 구현된 TinySec의 효율성을 검증하였다.

우선 가장 주목할 만한 것은 대부분의 실험에서 원래의 TinyOS 패킷과 TinySec 패킷의 성능 차이 (10% 이하)는 암호를 사용함으로써 생기는 계산량 증가 보다는 3바이트(TinySec-Auth) 혹은 5바이트(TinySec-AE)의 패킷 길이 증가에 기인한다는 사실이다. 이는 암호 알고리즘을 위한 전용 하드웨어를 사용하더라도 소프트웨어 구현에 비해 월등한 성능 향상을 얻을 수는 없다는 사실을 입증하는 것으로 하드웨어 구현의 비용과 소프트웨어 구현의 다양한 장점들을 고려하면 향후 저가의 소형 센서 노드에서의 보안기능 구현과 관련하여 시사하는 바가 크다고 할 수 있다. 물론 센서 노드의 하드웨어 플랫폼이 바뀌면 (예를들어 Telos와 같이 250Kbps의 radio를 사용하는 경우) 상황이 많이 다를 수 있으나, 최소한 매우 제한된 자원을 갖는 센서 노드에서도 소프트웨어로 구현시의 오버헤드가 생각보다 크지 않다는 것은 중요한 결과이다.

5.2 SPINS

SPINS는 센서 네트워크 보안을 위한 기본적인 보안 도구로 점대점 통신보안을 위한 SNEP(Secure Network Encryption Protocol)과 브로드캐스트 메

시지의 인증을 위한 μTESLA(Micro Timed Efficient Stream Loss-tolerant Authenti-cation)로 구성되어 있다. SNEP은 RC5 CTR 모드와 CBC-MAC을 기반으로 하는 암호화 및 인증을 위한 프로토콜로 두 통신 센서들간의 공유 카운터(counter)를 기반으로 하는 CTR 모드를 사용하여 데이터의 신선도(data freshness)를 보장한다. 통신 효율을 위해 카운터는 메시지와 함께 전송하는 것이 아니라 두 통신 노드가 매 블록마다 공유 카운터를 증가시켜 자체 관리를 하고, 카운터의 초기 공유나 재동기(re-sync)를 위한 별도의 카운터 교환 프로토콜(counter exchange protocol)을 제공한다. CTR 모드를 사용하므로 암호화로 인한 메시지 팽창은 없으며 8바이트의 MAC이 2바이트의 CRC를 대신하여 6바이트의 오버헤드가 발생된다. SNEP을 위한 키 관리는 베이스 스테이션과 각 노드 사이의 링크키를 기반으로 하며 이 키로부터 각 방향으로의 암호화키와 MAC키를 유도하여 사용한다.

SPINS의 가장 큰 공헌은 브로드캐스트 메시지에 대한 인증기능을 제공할 수 있는 μTESLA 프로토콜을 제안한 것이다. 이는 디지털 서명과 방향 키 체인(one-way key chain)을 이용한 기존의 TESLA 프로토콜을 제약된 환경의 센서 네트워크에 적합하도록 비밀키 암호 기반으로 수정한 것이다. TESLA 프로토콜의 기본 아이디어는 일방향 키 체인과 키의 지연 노출(delayed key disclosure)을 이용하여 비밀키 기반의 MAC을 통해 부분적인 공개키 암호의 기능을 구현할 수 있다는 것이다[25].

우선 송신자는 임의의 비밀키 K_n을 선택하여 길이 n인 키 체인의 초기값으로 삼아 일방향 함수(one-way function) f를 이용하여 n개의 키 체인 $K_i = f(K_{i+1})$ (i=n-1, ... 1,0)을 생성하여 저장해 둔다. 키 체인의 마지막 값 K_0(key chain commitment)는 모든 수신자들에게 인증 가능한 방법으로 전달한다. 각 키 K_i는 time interval i에서의 MAC 키로 사용하며 일정한 지연시간 (최소한 round-trip delay보다는 큰 time interval) 후 K_i를 노출시켜 모든 수신자가 이를 이용하여 이전의 수신 메시지에 대한 MAC을 검증하게 하는 것이다.

TESLA/μTESLA는 일방향 키 체인과 키의 지연 노출을 기반으로 브로드캐스트 메시지의 인증에서 가장 문제가 되는 키 관리 문제를 간단히 해결하였다는 점에서 큰 의의를 가진다. 즉 송신자는 임의의 비밀키를 초기값으로 선택할 수 있고, 단지 key chain의 맨 마지막 값만 모든 수신자들에게 비밀유지가 필요없이 인증 가능한 방법으로만 전송해 주면 된다. 단점이라면 수신자가 메시지의 MAC에 사용된 키가 노출될 때까지 일정한 지연시간 동안 패킷들을 저장하고 있어야 하며 (DoS 공격에 취약), 또한 모든 수신자들이 느슨하게라도 클럭의 동기가 맞아야 한다는 점이다.

TESLA에서는 마지막 키 체인 값의 전송을 위해 베이스 스테이션이 디지털 서명을 이용하여 브로드캐스트 하였으나 μTESLA에서는 센서 네트워크의 제약을 고려하여 이를 베이스 스테이션과 노드간의 공유키를 이용한 MAC 기반의 유니캐스트(unicast)로 수정하였다. 이는 공개키 암호의 사용을 피함으로써 계산이나 통신, 에너지 소모 등의 오버헤드를 상당히 줄였지만 대규모 센서 네트워크에서는 확장성에 있어서 여전히 큰 문제가 될 수 있다. Liu 등은 (TESLA를 다단계 키 체인(multi-level key chain) 방식으로 확장하여 초기의 유니캐스트 통신을 제거하고 DoS 공격에 대한 저항성을 가질 수 있는 개선방안을 제안하였다[19].

아직은 많은 문제점이 지적될 수 있으나 TESLA/μTESLA는 애드혹 네트워크의 각종 라우팅 프로토콜의 보안에 필수적인 브로드캐스트 메시지에 대한 인증을 위해 가장 널리 사용되는 프로토콜이다.

VI. 센서 노드용 암호 알고리즘

센서 노드가 센서 네트워크의 다양한 보안기능들을 수행하기 위해서는 최소한의 기본적인 암호 알고리즘을 탑재하는 것이 필요하다. 센서 노드는 연산, 통신, 메모리, 전력 등 모든 면에서 매우 제한된 자원만을 가지므로 암호 알고리즘도 가능한 적은 메모리를 사용하고 코드 사이즈가 작으며 계산량이 적은 것을 선택하는 것이 필요하다. 물론 장기적으로는 하드웨어로 센서 칩에 통합되는 것이 안전성이나 효율성 면에서 바람직하나 아직까지는 주로 소프트웨어 구현에 대한 연구가 주류를 이루고 있다.

다양한 능력을 갖는 센서 노드들이 존재하지만 가장 관심이 있는 것은 센서 네트워크에서 주류를 이루는 자원의 제약이 심한 소형 센서들이다. 이들 소형 센서 노드는 특히 메모리 사이즈가 매우 제한되어 있으므로 암호 알고리즘의 선택에 있어서도 코드 사이즈 역시 중요한 고려사항이 될 수 있다. Smart dust의 경우 4MHz, 8-bit CPU와 512B RAM, 8KB flash, 그리고 10Kbps 정도의 전송 속도를 가지며, Spec 모드의 경우 아직 테스트 단계로 많은 것이 알려져 있지는 않으나 8-bit custom CPU와 3KB의 RAM을 갖추고 있으며 통신 속도도 19.2Kbps 정도로 테스트 성공했다는 보도가 있다.

대부분의 센서 노드에서 사용하고 있는 TinyOS만으로 이미 수 KB 정도의 메모리를 사용하고 있으므로 이러한 최하위의 초소형 센서 노드들은 실제 보안 및 타 응용 프로그램이 사용 가능한 메모리는 역시 수 KB 정도 밖에 여유가 없다.

현재 상용화되어 가장 널리 보급되어 있으며 개발 환경이 제대로 갖추어진 센서 노드는 버클리 Mica 모트로 암호 알고리즘의 구현/테스트도 이를 기반으로 한 경우가 대부분이다. 가장 표준적으로 널리 사용되고 있는 사용되고 있는 Mica2 (8MHz 8-bit CPU, 4KB RAM, 128KB flash; 19.2Kbps Radio)나 Telos(8MHz 16-bit CPU, 2KB RAM, 60KB flash; 250Kbps Radio) 정도만 되어도 메모리의 제약은 상당 부분 완화된 것으로 볼 수 있으므로 최우선 과제인 전력소모의 최소화에 보다 집중할 수 있을 것이다. 전력소모는 계산량과 통신량에 의해 대부분 결정되므로 결국 암호 알고리즘의 선택에 있어서도 이들이 가장 중요한 요소가 된다.

센서 노드의 가장 중요한 자원은 에너지이다. 대부분의 현재 모트들은 2개의 AA 배터리를 필요로 하고 있는데 이것이 센서의 크기를 줄이지 못하는 가장 큰 이유이기도 하다 (예를들어 Spec 노드는 그 자체 크기는 기껏해야 5 평방 밀리미터 정도이지만 여전히 외장 배터리가 필요하니 결국 배터리의 크기가 전체 센서 노드의 크기를 좌우하는 셈이다). Mica2의 경우 전력을 최대로 사용할 경우 기껏해야 2주일 정도면 배터리가 동이 난다고 한다. 대부분의 응용에서 배터리의 수명이 센서 노드의 수명을 좌우하므로 최대한 전력소모를 줄여야 센서 노드의 수명(몇 년 정도는 살아 있어야 경제성이 있을 것임)을 늘릴 수 있을 것이다. 따라서 대부분의 시간을 sleep mode에 머물도록 하고 (1%이하의 duty cycle) active mode에서도 최대한 계산량이나 전송량을 줄일 수 있는 알고리즘 및 프로토콜들이 필수적이다. 향후 센서 노드의 대규모 상용화가 가능하기 위해서는 배터리의 크기를 줄이는 노력 뿐만 아니라 궁극적으로는 self-powered sensor node를 위해 주위의 환경(빛이나 열, 소음, 진동 등)으로부터 전력을 모을 수 있는 장치(power scavenger라 불림)의 개발에도 노력을 기울여야 할 것이다 (실제로 주위의 진동만을 이용해 약 8μW의 전력을 모을 수 있는 scavenger가 제안되기도 하였다)[10].

센서의 동작 중 전력 소모는 크게 계산과정과 통신과정으로 나눌 수 있다. 센서의 하드웨어 플랫폼에

따라 다를 수 있으나 대체로는 연산보다는 통신시에 전력 소모가 더 많은 것이 일반적이다. 따라서 알고리즘이나 프로토콜의 파라미터 선정시 통신량을 최소화하는 것이 무엇보다 중요하다. 연산과 통신시의 에너지 소모의 차이는 MCU 동작속도와 Radio 통신속도, 그리고 통신과 연산시의 전력소모량에 의해 결정되므로 센서의 하드웨어 플랫폼마다 다를 수 있다. 예를들어 8MHz 클럭 및 19.2Kbps 통신속도의 Mica2 모트의 경우 1바이트의 데이터 전송에는 대략 800-1000개의 명령어를 수행하는데 필요한 만큼의 전력을 소모한다고 알려져 있다. Telos의 경우도 통신속도는 대략 10배정도 높아졌지만 전력소모가 10배정도 줄었으므로 유사한 특성을 보일 것으로 짐작할 수 있다.

센서 노드에서 필요한 가장 기본적인 암호 알고리즘은 블록 암호로 아직까지는 센서 노드에 특화된 암호 알고리즘의 설계나 구현에 대해서는 그다지 많은 연구가 이루어지지는 않았으나 블록암호의 구현이나 선택기준 등에 대한 연구 결과들은 속속 보고되고 있다:

- 우선 Mica 플랫폼의 TinyOS에서 사용되는 링크 계층 보안 프로토콜 TinySec에서는 Skipjack과 RC5를 이용한 CBC모드의 암/복호화 및 CBC-MAC을 사용하고 있고, 네트워크 계층의 보안을 주 목표로 개발된 SPINS에서는 RC5를 이용한 CTR 모드의 암/복호화 및 CBC-MAC을 사용하고 있다[16,27].
- Yee Wei Law 등은 16-bit RISC기반의 TI MSP430F149(Telos나 EYES 등에서 사용되는 MCU)에서 RC5, RC6, AES, MISTY1, KASUMI, Camellia 등 6종의 블록암호에 대한 벤치마킹(데이터 메모리, 코드 메모리, CPU 사이클/에너지 효율성 등)을 바탕으로 센서 네트워크에 적절한 블록암호 및 운영모드에 대한 의견을 제시하고 있다[18]. 이들의 결론은 운영모드로는 정적인 네트워크(static network)에서는 OFB 모드를, 동적인 네트워크(dynamic network)에서는 CTR 모드를 권고하고 있고, 블록암호로는 에너지 효율성 면에서는 speed-optimized AES를, 자원의 제약이 심한 경우는 size-optimized MISTY1을 권고하고 있다. 주목할 만한 사실은 비록 코드 사이즈는 작지만 RC5가 대부분의 다른 블록암호에 비해 그다지 좋은 성능을 주지 못한다는 것이다. 이는 RC5가 많이 사용하는 가변 순환 이동(variable-bit rotation)이 8비트나 16비트의 저성능 엠베디드 프로세서들에서는 지원되지 않기 때문으로 보인다.
- Prasanth Ganessan 등은 스트림 암호 RC4, 블록암호 IDEA와 RC5, 해쉬함수 MD5와 SHA1 등에 대해 다양한 하드웨어 플랫폼(8비트 Atmega 103/128, 16비트 MC16C/10, 32비트 SA1110, PXAA250, UltraSparc2 등)에서의 실험 결과를 보고하고 있다[8]. 이 실험은 다양한 종류의 암호 알고리즘들에 대한 성능을 다양한 하드웨어 플랫폼에서 비교 분석한 것으로 각 플랫폼 별로 알고리즘의 선택이나 새로운 알고리즘의 설계시에 도움이 될 수 있을 것으로 보인다. 대부분의 mote에서 사용하고 있는 Atmega 플랫폼에서는 RC4가 RC5보다 우수하며, 해쉬함수가 블록암호에 비해 오버헤드가 훨씬 크다는 것은 주목할 만한 사실이다.

VII. 결 론

센서 네트워크의 대규모 상용화를 위해서는 아직까지 해결해야 할 과제들이 산적해 있으나, 센서 네트워크에서 무엇보다도 중요한 것은 프라이브시와

보안을 설계 단계에서부터 고려해야 한다는 것이다. 기존의 네트워크 환경에서와 같은 애드온(add-on) 형태의 보안은 더 이상 유효할 수 없으며, 보안이 제대로 갖추어지지 않은 센서 네트워크의 전개는 상상할 수 없는 부작용과 역기능을 초래할 수 있음을 인지해야 한다. 센서 네트워크의 다양한 제약 및 요구조건들은 센서 네트워크에 대한 위협요소들을 확장/증폭시켜 훨씬 더 보안을 어렵게 한다.

다양한 분야에서 센서 네트워크의 보안에 대한 연구가 진행되고 있으나 현재까지의 연구들은 아직은 초기단계이며 대부분 특정 분야의 부분적인 솔루션에 국한되어 있다는 문제점이 제기될 수 있다. 어느 분야에서나 마찬가지이지만 전체적인 보안을 위해서는 다양한 계층의 보안 프로토콜들이 통합적인 프레임워크 하에서 서로 연계되어 설계되고 구현되어야 하며, 또한 각 보안 프로토콜들이 경우에 따라서는 특정 네트워크 프로토콜이나 응용과 긴밀하게 결합되어 사용되어야 한다.

[참 고 문 헌]

[1] I.F.Akyiliz, W.Su, Y.Sankarasubramaniam and E.Cayirci, A survey on sensor networks, *IEEE Communications Magazine*, August 2002, pp.102-114.

[2] R.Anderson, H.Chan and A.Perrig, Key infecition: Smart trust for smart dust, 12th *IEEE International Conference on Network Protocols* (ICNP), Oct.2004.

[3] S.Basagni, K.Herrin, D.Bruschi and E.Rosti, Secure pebblenets, *MobiHOC 2001*, pp.156-163.

[4] H.Chan, A.perrig, and D. Song, Random key predistribution schemes for sensor networks, *IEEE Symposium on Research in Security and Privacy*. 2003, pp.197-213.

[5] H.Chan and A.Perrig, Security and privacy in sensor networks, *IEEE Computer*, Octber 2003, pp.103-105.

[6] W.Du, J.Deng, Y.Han and P.K.Varshney, A pairwise key pre-distribution scheme for wireless sensor networks, *CCS' 03*, October 27-30, 2003, Washington, DC.

[7] L.Eschenauer and V.D.Gligor, A key-management scheme for distributed sensor networks, *ACM CCS 2002*, Nov.2002.

[8] P.Ganessan, R.Venugopalan, P.Peddabachagari, A.Dean, F.Muller and M.Sichitiu, Analyzing and modeling encryption overhead for sensor network nodes, *WSNA' 03*, September 19, 2003, San Diego, California, USA.

[9] G.Gaubatz, J.Kaps and B.Sunar, Public key cryptography in sensor networks-revisited, *1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004)*, 2004.

[10] P.Gorder, Sizing up smart dusts, *Computing in Science & Engineering*, 5(6), Nov.-Dec. 2003, pp.6-9.

[11] C.Hartung, J.Balasalle and R.Han, Node compromise in sensor networks: The need for secure systems, *Technical Report CU-CS-988-04*, Dept. of Computer Science, University of Colorado at Boulder, 2004.

[12] J.Hill and D.E.Culler, Mica: a wireless platform for deeply embedded networks,

IEEE Micro, Nov.-Dec.2002, pp.12-24.

[13] J.Hill, M.Horton, R.Kling and L.Krishnamurthy, The platforms enabling wireless sensor networks, *Communications of the ACM*, Vol.47, No.6, June 2004, pp.41-46.

[14] F.Hu, J.Ziobro, J.Tillett and N.Sharma, Secure wireless sensor networks: Problems and solutions, *J.of SCI*, to appear, 2004.

[15] C.Karlof and D.Wagner, Secure routing in wireless sensor networks: Attacks and contermeasures, *Ad Hoc Networks*, vol. 1, issues 2-3 (Special Issue on Sensor Network Applications and Protocols), Elsevier, Sep. 2003, pp.293-315.

[16] C.Karlof, N.Sastary and D.Wagner, TinySec: A link layer security architecture for wireless sensor networks, *ACM SenSys 2004*, Nov. 3-5, 2004.

[17] Y.W.Law, S.Dulman, S.Etalle and P.Havinga, Assessing security-critical energy-efficient sensor networks, *18th IFIP International Information Security Conference*, May 2003.

[18] Y.W.Law and J.M.Doumen and P.H.Hartel, Benchmarking block ciphers for wireless sensor networks (Extended abstract), *1st IEEE Int. Conf. on Mobile Ad-hoc and Sensor Systems*, Fort Lauderdale, Florida, Oct. 2004.

[19] D.Liu and P.Ning, Efficient distribution of key chain commitments for broadcast authentication in distributed sensor networks, *NDSS' 03*, 2003.

[20] D.Liu and P.Ning, Establishing pairwise keys in distributed sensor networks, *ACM CCS 2003*, Oct.2003.

[21] S.R.Madden, M.J.Franklin, J.M.Hellerstein and W.Hong, TAG: a Tiny AGgregation service for ad-hoc sensor networks, *OSDI' 02*, Dec.2002.

[22] D.Malan, M.Welsh and M.D.Smith, A Public-key infrastructure for key distribution in TinyOS based on elliptic curve cryptography, *IEEE SECON 2004*.

[23] S.Marti, T.J.Giuli, K.Lai and M.Baker, Mitigating routing behavior in mobile ad hoc networks, *Annual Internation Conference on Mobile Computing and Networking*, Aug.2000.

[24] A.Mishra, K.Nadkarni, and A.Patcha, Intrusion detection in wireless ad hoc networks, *IEEE Wireless Communications*, February 2004, pp.48-60.

[25] A.Perrig, R.Canetti, D.Song and D.Tygar, Efficient and secure authentication for multicast, *NDSS' 01*, Feb.2001.

[26] A.Perrig, J.Stankovic and D.Wagner, Security in wireless sensor networks, *Commun. Of ACM*, 47(5), June 2004, pp.53-57.

[27] A.Perrig, R.Szewczyk, J.D. Tygar, V.Wen and D.Culler, SPINS: Security protocols for sensor networks, *Wireless Networks* 8, Kluwer Academy Publishers, 2002, pp.521-534.

[28] J.Polastre, R.Szewczyk and D.Culler, Telos: enabling ultra-low power wireless research, *Application notes 002*, www.moteiv.com.

[29] B.Przydatek, D.Song and A.Perrig, SIA: secure information aggregation in sensor networks, *SenSys' 03*, Nov.2003.

[30] M.Tubaishat and S.Madria, Sensor networks: An overview, *IEEE Potentials*, April/May 2003, pp.20-23.

[31] D.Wagner, Resilient aggregation in sensor networks, *Proc. of 2nd ACM workshop on Security of Ad hoc and Sensor Networks (SASN '04)*, pp.78-87, ACM Press, 2004.

[32] J.P.Walters, Z.Liang, W.Shi and V.Chaudhary, Wireless sensor network security: a survey, http://www.cs.wayne.edu/~weisong/papers/walters05-wsn-security-survey.pdf.

[33] R.Watro, D.Kong, S.Cuti, C.Gardiner, C.Lynn and P.Kruus, TinyPK : Securing networks with public key technology, *SASN '04*, pp.59-64, ACM Press, 2004.

[34] A.D.Wood and J.A.Stankovic, Denial of service in sensor networks, *IEEE Computer*, October 2002, pp.54-62.

[35] P-W.Yau and C.J.Mitchell, Reputation systems for routing security for mobile ad hoc networks, *SymoTIC' 03*, Oct.2003.

[36] S.Zhu, S.Setia, and S.Jajodia, LEAP: Efficient security mechanisms for large-scale distributed sensor networks, *ACM CCS 2003*, pp.62-72.

임채훈

1982년 ~ 1989년 서울대 전자공학과 학사
1990년 ~ 1992년 포항공대 전자전기공학과 석사
1992년 ~ 1996년 포항공대 전자전기공학과 박사
1989년 ~ 1990년 ㈜데이콤 사원
1997년 ~ 2002년 ㈜퓨쳐시스템 기술이사
2002년 ~ 현재 세종대학교 인터넷학과 조교수