센서 네트워크를 위한 경량 키 관리 프로토콜

김경태, 김형찬, R. S. Ramakrishna
광주과학기술원 정보통신공학과
e-mail : {ktkim, kimhc, rsr} @gist.ac.kr

A Lightweight Key Management in Wireless Sensor Network

Kyeong Tae Kim, Hyung Chan Kim, R. S. Ramakrishna
Department of Information and Communications,
Gwangju Institute of Science and Technology (GIST)

요 약

본 논문에서는 센서네트워크에서 키 관리 및 안전한 채널을 통한 개체 인증, 메시지의 기밀성 및 무결성을 보장하는 프로토콜을 제안한다. 제안된 프로토콜은 레벨기반의 트리 라우팅 프로토콜을 지원하여 외부 공격, 노드 실패, 그리고 노트 전복에 대하여 강인하며, 노드 상호간의 통신을 위하여 경량 키 관리 알고리즘을 사용하였다. 또한, In-networking 프로세싱을 통해 Data-aggregation 과 Trusted-delivery 메커니즘을 적용하여 가볍고, 견고한 특성을 지니고 있다. 제안된 프로토콜은 TinyOS/Mica 플래폼을 기반으로 TinySec 과 함께 구현된 결과를 제시한다.

1. 서론

센서네트워크의 응용분야는 군사목적을 위한 감시 시트템, 생태계 모니터링, 스마트 트랜스포트 분야 등 많은 분야에 걸쳐 연구가 진행되고 있다. 이러한 기술을 신뢰성 있고 안전하게 사용하기 위하여 네트워크 노드의 개체 인증, 메시지에 대한 기밀성과 무결성에 대한 지원이 요구된다. 본고에서는 센서네트워크의 보안 서비스를 제공하기 위하여 외부 공격, 노드의 실패와 전복 등의 문제에 대해 강인한 키 관리 프로토콜을 제안한다.

제안된 키 관리 프로토콜은 신뢰할 수 있는 중앙서버의 의존하지 않고 Self-organizing 방법으로 노드들 사이에 암호화된 키를 분배함으로써, 관리하는데 드는 비용을 최소화 하였다. 이를 이용함으로써 얻어지는 장점은 다음과 같이 요약될 수 있다.

우선, 공개키 암호화 기법을 사용하는 것이 아니라, 센서 네트워크 크기와 관계없이 비대칭 키 집합을 저장하고 사용하기 때문에 연산하는데 드는 비용과 키 분배면에서 가볍다. 또한 베이스 스테이션(Base Station)에서 브로드캐스트(Broadcast) 방법으로 키를 분배함으로써, 노드의 Pre-distribution 과정을 필요로 하지 않는다. 추가적인 클러스터 헤더(Cluster Header)를 선출할 필요가 없고, 오직 키 설정을 관리하는 약간의 오버헤드만이 요구된다. 네트워크 Topology 에 대한 레벨키와 노드 자신에 대한 노드키를 설정할 때 가벼운 일방향 함수를 사용하여, 트리구조의 Topology 에서 부모노드가 모든 서브 자식노드에 해당하는 키를 저장해야 하는 단점을 극복한다.

제안된 프로토콜은 Pairwise-key 를 이용하여 안전한 로컬 채널을 형성하고 있다. 라우팅 과정에서 각각의 노드들은 다른 키를 사용하여 다양한 경로로 부모노드에게 정보를 보낼 수 있다. 또한 하나의 서버에 대한 의존성이 없기 때문에 Bottleneck 과 서버 노드의 실패에 대해 문제가 적다. In-network 프로세싱을 위해서 레벨키와 Pairwise-key 를 이용하여 Data-aggregation 과정에 적용되는 Trusted-delivery 메커니즘을 개발하였다.

본문의 구성은 다음과 같다. 2 장에서는 배경기술을 소개하고 3 장에서는 제안된 프로토콜을 크게 두 부분, 라우팅 프로토콜과 키 관리 서비스으로 나누어 설명한다. 4 장에서 프로토콜 구현에 대해서 다루고, 5 장에서는 다른 키 관리 프로토콜과 특징을 비교하여 설명했으며, 6 장에서 결론을 맺는다.

2. 배경

2.1 Data-Aggregation

Data-Aggregation 은 한 노드로 들어온 여러 개의 입력 데이터 패킷을 하나의 패킷으로 내보내는 과정을 일컫는다. 이것은 중복된 패킷 전송을 제거함으로써, 에너지를 소비를 줄이는 기술이다. 이를 달성하기 위해서 Aggregation 노드는 들어오는 모든 패킷의 내용을 파악해야 하는데, 이 때, 메시지 내용에 대한 기밀

성이 필요하다.

2.2 Trusted-delivery 메커니즘

Trusted-delivery 메커니즘은 악의 있는 노드를 구별할 수 있는 해결책을 제시한다. 자식노드가 메시지를 조상노드에게 전달하고자 할 때, 자식노드는 부모노드가 메시지를 조상노드에게 제대로 전달했는지를 확인할 수 있어야 한다. 제안된 키 관리 프로토콜은 이를 확인 할 수 있는 메커니즘을 포함하고 있다.

2.3 Aggregation 트리기반의 라우팅

싱크(Sink)노드가 초기에 Request 메시지를 Flooding 하면, 이 메시지를 받은 각각의 노드들은 주기적으로 데이터를 Aggregation 트리에 보낸다. 이 때, 싱크노드는 이 메시지를 받으면서 전체 네트워크의 Topology 정보를 확인할 수 있다. Flooding 기법은 각각의 노드가 트리구조에서 누구로 부터 Request 메시지를 받았는지 확인하고, 이를 부모노드로 삼고 Reverse-path 트리를 설정한다. 이 기법에 대해 다양한 연구(Diffusion[2], Bless, Surge[3]) 가 진행되었으며, 일반적으로 Reverse-path 가 가장 좋은 방법은 아니지만 단순한 구현을 위해 이 기술을 적용하였다.

트리기반의 라우팅은 크게 두 가지 정보, 즉, 부모노드의 아이디와 홉 카운트(Hop count)로 구성되는데, 라우팅 트리는 루트노드가 브로드캐스팅할 때 이에 선택적으로 응답하는 자식노드들의 응답을 보고 경로를 설정한다.

많은 트리 기반의 라우팅 프로토콜이 이 기법을 주로 이용하는 이유는 Topology 구성이 단순하고 Forwarding 기술을 이용함으로써 쉽게 구현할 수 있기 때문이다. 하지만 안정한 Topology 유지와 다양한 라디오 통신환경에 좋은 채널연결상태를 관리하는데 많은 주의가 요구되는 단점이 있다.

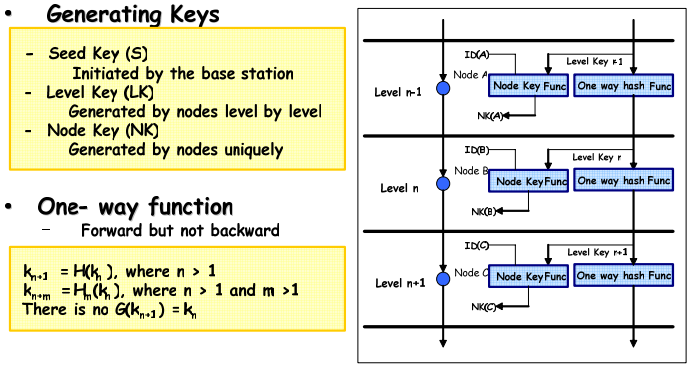
3. 키 관리

3.1 키 관리

제안된 프로토콜은 센서네트워크의 제약사항인 에너지 소모의 최소화를 위해 data-aggregation 을 적용하고, 보안을 위해 Trusted-delivery 메커니즘을 도입하였다. 이를 시뮬레이션 하기위해서 Berkeley 대학에서 개발한 TinyOS 를 바탕으로 TOSSIM[9]에서 실험하였다. 우선, 트리구조 기반의 네트워크와 Flooding 기법[1]을 이용하여 라우팅 프로토콜을 구현하였다. 이러한 기술들은 In-network 프로세싱을 통해 수행되는데, 이것은 데이터가 생성된 근처의 노드에서 여러 노드들이 협력하여 센서에 의해 획득한 데이터를 처리하는 것을 의미한다. 또한 키 관리 서비스는 암호화된 키를 처리하는 일을 담당하는데, 이는 키의 생성 및 분배, 그리고 암호화 방법 등을 포함한다.

3.2 키 생성

그림 1 은 키 생성과정을 보여준다. 모두 세 가지 키가 사용되는데, 레벨키(Level Key)와 노드키(Node Key)는 시드키(Seed Key)로부터 계산된다. 시드키는 베이스 스테이션으로부터 생성되며 보안 세션을 결정



(그림 1) 키 생성 과정.

하는데 사용된다. 레벨키는 같은 홉 카운트에 있는 노드들끼리 공유하고 있으며 레벨숫자가 바로 홉 카운트를 나타낸다. 레벨키는 일방향 속성을 갖고 있는데, 낮은 레벨키로부터 높은 레벨키를 계산할 수 있지만, 반대의 작업은 불가능하다. 이것은 높은 레벨의 노드, 즉 홉 카운트가 높은 자식노드가 홉 카운트가 낮은 부모노드들사이에 통신을 알아 차릴 수 없게 만든다. 보통 일방향 함수는 MAC 에 많이 사용되지만, 여기서는 경량화를 위해 키 생성에 이용되었다. 각 노드마다 두 개의 함수를 관리하는데, 부모노드로부터 자신의 레벨키를 받으면, 이 때부터 두 개의 함수를 통해 키 생성이 시작된다. 첫번째는 부모로부터 받은 레벨키를 입력 값으로 자식노드의 레벨키를 생성하는 일방향 함수이고, 다른 하나는 레벨키와 자신의 아이디로 노드키를 생성하는 함수이다.

3.3 키의 암호화

키를 사용하여 데이터를 암호화하는 방법으로 대칭키를 이용했다. Data-aggregation 을 수행하기 위해서 낮은 레벨의 노드는 높은 레벨의 노드로부터 받은 데이터 메시지를 해독할 수 있어야 한다. 이를 위해 낮은 레벨의 노드는 높은 노드의 레벨키를 알고 있어야 하는데, 3.2 절에서 소개한 일방향 함수는 이러한 요구조건을 만족한다. 또한 높은 레벨의 노드는 낮은 벨의 레벨키를 알 수가 없기 때문에 부모노드의 통신내용을 해독할 수 없다.

3.4 Pairwise-key

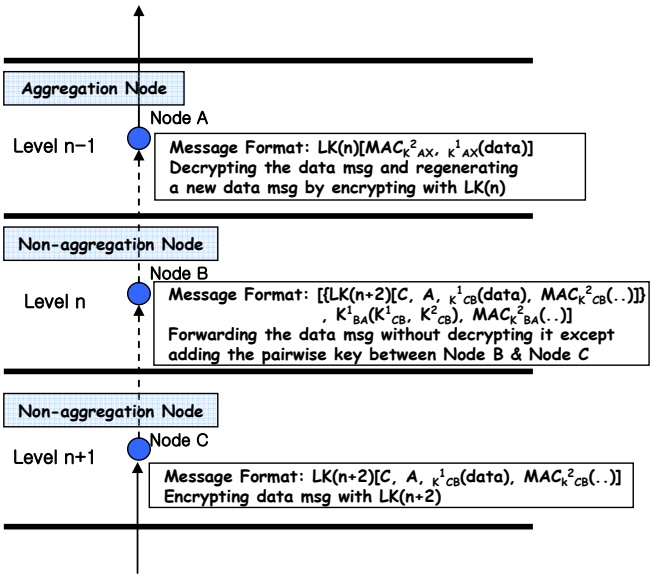
부트스트래핑 프로토콜[4]은 부모노드와 자식노드 사이에 인접한 안전한 채널을 구축한다. 이것은 자식노드의 레벨키를 이용함으로써 쉽게 구현될 수 있다. 자식노드 C 가 임의의 넌스(Nonce), Nc 를 만들고 Hello 메시지를 부모에게 보냄으로써, 프로토콜을 초기화 할 수 있다. 메시지 포멧은 다음과 같다.

<Hello, C, Nc, MAC_{LK}(Hello, C, Nc)>

메세지는 자식노드의 아이디(C)와 넌스(Nc), 그리고 레벨키(Lk)를 키 입력값으로 한 MAC(Message Authentication Code)를 포함한다. Hello 메시지를 수신한 부모노드 P 는 MAC 을 통해 C 의 인증을 확인한 후 C 에게 ACK 메시지를 보낸다.

<ACK, C, P, Np, MAK_{LK}(ACK, C, P, Nc, Np)>

이 메세지도 P 로부터 생성된 넌스(Np)를 포함하고



(그림 2) Data-aggregation.

있으며, C 에게 Nc 를 잘 받았고 자식노드의 레벨키를 알고 있다는 것을 확인시켜 준다. 이러한 메시지들을 교환한 후에, C 와 P 는 각각의 넌스를 추출하고 Pairwise-key 를 생성한다. $K_{CP} = G_{NK(C)}(Nc, Np)$, G 는 자식노드의 노드키를 입력으로 하는 일방향 함수이고, K_{CP} 는 두 부분의 서브키, K^1_{CP}, K^2_{CP} 로 쪼개어진다. 각각 데이터를 암호화하는 키와, 인증에 필요한 MAC 에 대한 키 입력값으로 쓰인다.

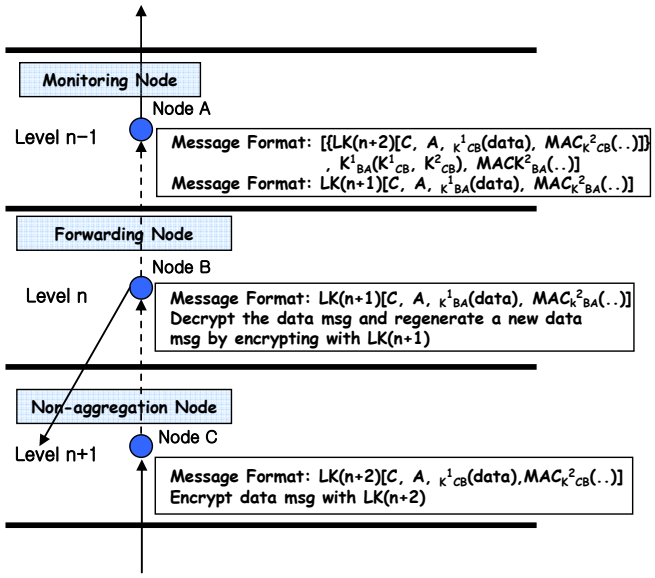
3.5 Data-aggregation 과 Trusted-delivery 메커니즘

그림 2 는 Data-aggregation 과정을 보여준다. 노드 A 를 Aggregation 노드라고 가정하고 레벨이 n+1 인 Leaf 노드 C 로 정보를 수집한다고 한다면, 노드 C 는 먼저 메시지를 Non-aggregation 노드인 B 에게 전송해야 한다. 메시지 포멧은 다음과 같다.

<LK(n+2)[C, A, K^1_{CB}(data), MAC K^2_{CB}(C, A, K^1_{CB}(data))]>

Lk(n+2)는 Leaf 노드 C 의 다음 레벨키이고, Pairwise-key, 즉, K^1_{CB}, K^2_{CB} 가 각각 데이터 암호화와, MAC 의 키 입력으로 사용된다. 이 때, 노드 B 는 C 로 부터 받은 데이터를 해독하는 과정없이 그대로 노드 A 에게 전달하는데, 다만 노드 C 와 B 사이에 Pairwise-key 를 K^1_{BA} 로 암호화한 결과와 K^2_{BA} 를 입력으로 한 MAC 이 추가된다. 결국, 노드 A 는 B 의 인증을 확인한 후, C 와 B 사이의 Pairwise-key 를 추출하고 노드 C 로부터 전달된 내용을 Aggregation 할 수 있게 된다.

Trusted-delivery 메커니즘은 Data-aggregation 과정과 함께 응용될 수 있다. 노드 C 가 B 의 행위를 감시하는 모니터링 역할을 한다고 가정하자. 일단 노드 C 가 B 에게 메시지를 전달하면, 노드 B 는 A 에게 Data-aggregation 과 마찬가지 방법으로 메시지를 전달한다. 하지만, 여기서는 한 가지 과정이 더 추가된다. 노드 B 는 노드 A 로 부터 받은 메시지를 해독한 후에 자식노드의 레벨키 즉, Lk(n+1)로 데이터를 암호화하고 부모노드에게 전달한다. 이것은 노드 B 가 MAC$_{K^2_{CB}}$ 를 MAC$_{K^2_{BA}}$ 로 바꾸고, 레벨키 Lk(n+1)로 암호화 하기 전에 K^2_{BA} 로 데이터를 암호화하는 과정을 포함한다. 결국, 노드 A 는 각각 노드 C 와 B 로부터 받은 메시지를 비교함으로써, 메시지가 노드 B 로부터 제대로 포워딩되었는지를 확인할 수 있다. 그림 3 은 이 메커니즘을 나타낸다.



(그림 3) Trusted-delivery 메커니즘.

4. 구현

4.1 Framework

Mica 플랫폼은 TinyOS[5] 기반 위에서 동작한다. TinyOS 는 작은 센서네트워크 플랫폼을 위해 개발된 모듈러 오퍼레이팅 시스템으로, 인터페이스와 이를 구현한 모듈을 연결한 여러개의 컴포넌트로 구성된 소프트웨어 어플리케이션이다. NesC [6, 7]는 TinyOS 컴포넌트를 기술하는 언어로 C 언와 유사한 문법체계를 지녔다. 모든 컴포넌트들은 TinyOS 1.1.14 버젼으로 구현되었다.

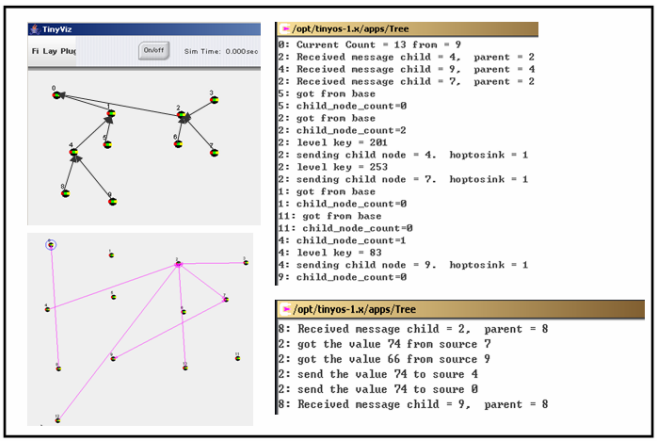
TinyViz 는 자바를 기반으로 한 GUI 시뮬레이터로 디버그 메시지와 라디오 통신, 그리고 UART 패킷의 내용 등을 분석하고 시각화하여 보여준다.

TinySec[8]은 경량의 링크계층의 암호 메커니즘으로 TinyOS 에서 통신할 때 기본으로 주고받는 TOSMsg 패킷의 구조 중에 CRC 를 MAC 으로 대체하였으며, 페이로드를 암호화 한다. 이렇게 암호화된 메시지의 사용은 보안이 필요한 라디오 채널 통신에 반드시 필요하며, 모든 메시지의 교환은 TinySec 기반위에 구현되었다.

4.2 Aggregation-Tree 라우팅 구축

Reverse flooding aggregation 트리를 구축하는데 크게 두 단계가 필요하다. 싱크노드가 request 메시지를 전체 네트워크에 Flooding 하는 과정과, request 메시지를 받은 각각의 노드들이 Flood 모듈을 통해 싱크노드로 response 메시지를 보내는 과정이 그것이다. 이 때, 각각의 노드는 Flood 모듈에게 누구로부터 패킷을 받았

는지 질의할 수 있는데, 이 노드가 바로 부모노드가



(그림 4) 보안이 강화된 Data-aggregation.

된다. 멀티홉 환경을 구성하기 위해 TOSSIM[9]에서 Non-lossy 모델을 이용하였고, 싱크노드는 Leaf 노드로부터 주기적으로 정보를 수집함으로써, 네트워크 Topology 를 감독하고 예측할 수 있다.

통신중에 패킷의 분실을 해결하기 위해 TOSMsg 메시지의 Ack 필드를 이용하였다. 이것은 수신자가 메시를 제대로 수신한 경우에, 이 필드를 1 로 설정하여 송신자에게 보내준다. 또한 패킷의 충돌을 최소화하기 위해 Binary Exponential Back-off 알고리즘을 사용하였다.

4.3 Data-aggregation 과 Trusted-delivery 메커니즘

그림 4 는 Data-aggregation 과 Trusted-delivery 메커니즘 과정에 대한 시뮬레이션 과정을 보여준다. Leaf 노드는 0-99 사이에 임의의 숫자를 생성하고 매 30 초마다 부모노드에게 보낸다. 이 때, Aggregation 노드는 자식들로 부터 받은 값들을 추적하고 최대값이 검출되면 조상노드에게 그 값을 전달한다. 결국 베이스 스테이션은 Data-aggregation 과정을 통해 네트워크에서 최대값을 지닌 노드의 값을 알아낼 수 있게 하는 시나리오를 실험하였다.

5. 고찰

센서네트워크에서 비대칭 암호기법의 사용은 많은 양의 에너지를 소모하고 시간 소모가 크기 때문에 효율적이지 못하다. 반면에 대칭 암호키의 사용은 간단하고 에너지 사용 측면에서 효율적이지만 하나의 노드 전복에 전체 네트워크의 통신 내용이 노출될 수 있는 문제가 있다. 제안된 프로토콜은 이 두가지 프로토콜의 단점을 최소하기 위하여 설계되었다. 첫째로 경량화를 위해 Data-aggregation 을 도입하였다. 이 때, Data-aggregation 에 참여하지 않는 노드는 어떠한 암호해독의 연산없이 암호화된 데이터를 그대로 전달한다. 반면에 Aggregation 노드는 암호화된 데이틀 해독할 수 있고 Trusted-delivery 메커니즘을 사용하여 악의적인 노드의 Forwarding 행위를 모니터링 할 수 있다. 즉, 부모노드의 실패나 전복이 발생할 경우에도, 자식노드는 다른 부모노드의 경로로 우회하여 암호화된 데이터를 보내고 이를 모니터링 함으로써, 변화무쌍한 센서네트워크의 환경에 적응할 수 있었다.

6. 결론

본 연구에서는 적은 에너지 소모와 메시지 인증과 기밀성을 제공하는 안전한 센서 네트워크 환경을 위해 Data-aggregation 과 Trusted-delivery 메커니즘을 제안하였다. 우선, 가벼운 일방향 함수를 이용함으로써, 레벨키와 노드키를 생성하고 레벨키를 인자값으로 Pairwise-key 를 구축하여 노드들 사이에 안전한 채널을 설정하였다. 이것은 라우팅 경로마다 다른 키를 사용함으로써, 노드 전복에 견딜 수 있고 Non-aggregation 노드의 행위를 모니터링 할 수 있는 장점이 있다. 또한 In-network 프로세싱을 위해, 레벨키와 Pairwise-key 를 이용하여 Data-aggregation 과 Trusted-delivery 메커니즘을 접목시켜, 통신채널을 보다 강인하고 안전하게 유지할 수 있다. 제안된 프로토콜은 TOSSIM 에서 시뮬레이션 할 수 있도록 nesC 로 구현되었다.

라우팅 프로토콜을 보다 안정적으로 구현하고 rekeying 메커니즘을 추가하는 것이 과제로 남아있다. 또한 다른 경량의 키 관리 프로토콜과 성능을 비교하고, 다양한 암호화 모듈의 성능을 테스트 하는 작업도 추가 될 것이다.

참고문헌

[1] L. Phillip, and M. Sam, "The Emergence of Networking Abstractions and Techniques in TinyOS" In Proc. of the First USENIX/ACM Symposium on Networked Systems Design and Implementation (NSDI) (2004).

[2] I. Chalermek, G. Ramesh, and E. Deborah, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks" Technical Report 00-732, University of Southern California (2000).

[3] L. Phillp et al. " Ad-Hoc Routing Component Architecture" : http://www.tinyos.net/tinyos-1.x/doc/ad-hoc.pdf (2003).

[4] D. Bruno, C. Steven, and L. Joshua, "Lightweight Key Management in WSN by Leveraging Initial Trust" SDL Technical Report SRI-SDL-03-02 (2004).

[5] TinyOS Tutorial: http://www.tinyos.net.

[6] G. David, and L. Philip, "NesC 1.1 Language Reference Manual " : http://www.tinyos.net/tinyos-1.x/doc/nesc/ref.pdf (2003).

[7] G. David, and L. Phillp, "The nesC Language: A holistic approach to Network Embedded Systems by" http://nescc.sourceforge.net.

[8] K. Chris, S. Naveen, and W. David, "TinySec (A Link Layer Security Architecture for WSN)" ACM UC Berkeley: http://www.cs.berkely.edu/~nks/tinysec/ (2004).

[9] TOSSIM (A simulator for TinyOS network): http://www.tinyos.net/tinyos-1.x/doc/nido.pdf.