유비쿼터스 센서네트워크에서 에너지효율을 고려하는 비동기적인 키관리 기법

정회원 윤 미 연*

Asynchronous Key Management for Energy Efficiency over Wireless Sensor Network

Mi-youn Yoon* ***Regular Member***

요 약

최근 유무선 네트워크는 IP 코어망을 중심으로 가입자망의 형태를 가지는 BcN(Broadband convergence Network)로 진화되고 있으며, 이와함께 인간 외부환경의 감지를 수행할 유비쿼터스 센서네트워크(USN : Ubiquitous Sensor Network)가 새로이 연구되어 오고 있다. 감지하는 대부분의 데이터의 경우에 악의를 가진 노드에게 노출되거나 위변조 되어서는 안될 정보들이기 때문에 센서네트워트에서 에너지 효율을 고려한 정보보호 기법을 요구한다. 본 논문에서는 임의의 센서노드부터 싱크까지의 에너지 효율적이고 안전한 데이터 전달을 위한 정보보호기법으로, 단순한 해쉬함수의 계산을 중심으로 에너지 소모를 줄이는 키관리기법을 제안하였다. 제안하는 키관리기법의 보안성 분석을 위하여 만족해야 하는 보안성에 대해 정의·증명하였으며, 에너지효율성을 측정하였다. 각 기법은 기존의 관련연구와 비교분석하여 본 논문에서 제안한 기법이 우수함을 증명하였다.

Key Words : Sensor Network, Key Management, Energy, Security

ABSTRACT

WSN(Wireless Sensor Network) performs to detect and collect environmental information for one purpose. The WSN is composed of a sink node and several sensor nodes and has a constraint in an aspect of energy consumption caused by limited battery resource. So many required mechanisms in WSN should consider the remaining energy condition. To deploy WSN, the collected information is required to protect from an adversary over the network in many cases. The security mechanism should be provided for collecting the information over the network. we propose asynchronized key management considering energy efficiency over WSN. The proposed key management is focused on independence and difference of the keys used to deliver the information over several routes over the network, so disclosure of any key does not results in exposure of total key information over the overall WSN. Also, we use hash function to update key information for energy efficiency periodically. We define the insecurity for requested security properties and proof that the security properties are guaranteed. Also, we evaluate and analyze the energy efficiency for the proposed mechanism.

Ⅰ. 서 론

USN(Ubiquitous Sensor Network)는 자료 수집을 위한 네트워크로 초경량, 저전력의 많은 센서 노드들이 넓은 지역이나 조밀한 지역에 설치되어 무선으로 자료를 주고받는 네트워크이다. 센서노드는 싱크 노드라는 관리 노드로부터 자료 요청을 받으며 수집된 자료를 송신하게 되는데 이 과정에서 센서노

* 한국정보보호진흥원 (myyoon@kisa.or.kr), ** 숭실대학교 (goodwin77@cherry.ssu.ac.kr)
논문번호 : KICS2005-08-332, 접수일자 : 2005년 8월 11일, 최종게재논문통보일자 : 2006년 7월 27일

드와 센서노드들 간의 통신이 무선으로 이루어지며 센서 노드와 싱크 노드사이에서도 데이터의 전송이 무선으로 이루어진다. 이러한 USN은 인간 외부환경의 감지와 제어 기능을 수행하게 될 것이며, 예를 들어 홈자동화를 유도하는 홈네트워크의 구성에도 USN이 적용될 것이며, 도로의 교통관리 시스템, 의료시스템, 물류시스템 하물며 인간의 경제 활동에도 USN의 기술이 적용될 것으로 기대된다.

그러나 USN은 인간을 중심으로 한 여러 외부 환경에 대한 정보를 수집·관리하는 네트워크이기 때문에 수집된 여러 정보들이 악의를 가진 제 3자에게 노출될 경우 개인정보의 노출, 나아가서는 국가 기밀 정보의 유출 등의 문제로 악화될 우려가 높다.여기서 센서노드가 저전력 노드인 점을 감안하면 에너지를 효율적으로 사용할 수 있어야 한다. 따라서 본 논문에서는 USN 상의 정보보호를 위하여 에너지의 효율성을 고려하는 키관리 기법을 제안한다.

사전에 오프라인 상으로 비밀키 정보 K_s를 갖으며 인증키 및 초기의 세션키는 비밀정보 K_s에 의해 생성 및 갱신이 이루어진다. 또한, 데이터의 암호화가 요구될 경우, 싱크로부터 분배 받은 부분키 비밀정보 Kc를 이용하여 암·복호키를 생성한다. 생성된 인증키, 암복호키는 제안한 주기에 따라 갱신하도록 하여, 키의 전방향 보안성 및 비구별성을 보장한다. 각 주기는 데이터를 송신하는 소스별로 주기를 상이하게 함으로써 소스에서 싱크까지의 경로상의 유일키를 설정할 수 있는 효과를 주도록 하였다. 오프라인 상으로 서로 협의된 키비밀정보를 공유하게 함으로써 초기의 키분배에 따른 에너지 소모를 줄이고, 갱신주기를 달리함으로써 센서네트워크 전체의 키값의 동기화로 인한 보안성 약화를 해결하였으며, 한번의 주기로 인증키 및 암복호키의 갱신이 모두 가능하도록 하여 에너지소모를 줄이고자 하였다.

2장에서는 본 논문과 관련된 여러 기존 연구들을 소개하고 기존연구의 장단점을 분석하고 3장에서는 에너지 효율적인 키관리기법에 대해 제안하고 보안성에 대한 분석을 수행한다. 그리고 4장에서는 제안하는 기법을 에너지 효율적인 측면에서 성능을 분석하도록 한다. 마지막으로 5장에서는 본 논문의 결론을 맺고 향후 과제를 제시한다.

Ⅱ. 관련 연구

2장에서는 센서네트워크상에 존재하는 물리적 여러 제약조건들을 포함하여 USN에 대해 소개한다. 그리고 기존의 관련된 연구와 센서네트워크 환경에서의 대표적인 보안프로토콜로서 SPINS[2]에 대해 알아보고 그 특징을 비교분석한다.

2.1 USN의 개요

USN은 자료 수집을 위한 네트워크로 초경량, 저전력의 많은 센서노드들이 넓은 지역이나 조밀한 지역에 설치되어 무선으로 자료를 주고받는 네트워크로 유비쿼터스 컴퓨팅환경의 중요한 요소이다. 유비쿼터스 컴퓨팅 환경이란 어떤 기기(Any device)로든 언제(Anytime) 어디서나(Anywhere) 사용자가 PC를 활용할 수 있는 환경이다. 이러한 환경에서 인간 외부환경의 감지와 제어기능을 수행하기 위한 네트워크가 바로 USN이다. 그림 1에서와 같이 USN은 자신의 주변정보를 감지하고 전달하는 센서노드와 여러 센서로부터 전달된 정보를 저장하는 싱크로 구성된다. 센서노드는 이동성을 가지며 싱크는 이동성을 가질 수도 있으며 한 개 이상으로 구성될 수도 있다.

각 센서노드는 자신이 담당하고 있는 영역에서 목표의 상태정보를 수집하고 이를 무선채널을 통하여 싱크까지 전달한다. 이때, 데이터의 전달은 센서노드에서 싱크까지 한 홉이상 존재할 수 있다. 따라서 싱크와 센서들 간의 애드 혹 네트워크가 형성되어야 하며 형성된 애드 혹 네트워크상의 센서노드들을 라우터로 하여 싱크까지 데이터를 전달할 수 있다. 이렇게 전달된 데이터는 싱크를 통해 본 정보를 요구하는 관찰자에게 보내지며 싱크와 관찰자 시스템은 동일하거나 상이할 수도 있다. 상이한 경우, 싱크부터 관찰자 시스템까지는 싱크가 가지는 가입자망의 형태에 따라 무선네트워크 또는 유선네트워크를 통해 전달될 수 있다.

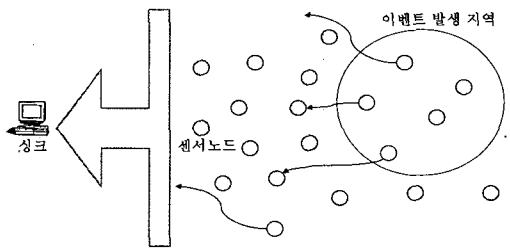


그림 1. USN의 구성

USN은 타 네트워크와 비교할 때 하드웨어적인 제약조건들이 여러 소프트웨어적 기능에 끼치는 영

향이 상대적으로 큰 네트워크이다. 각각의 센서노드들은 저전력의 배터리를 가지고 있으며, 각 노드가 가지는 각각의 프로세서 또한 500mW미만의 전력과 200MHz미만의 클럭속도를 가지고 있다[10]. 따라서 기존의 다른 네트워크에서의 단말들과 비교하면 가용에너지가 현저히 낮기 때문에 센서노드에 적용되는 모든 기법은 에너지 소모의 정도를 고려해야만 한다. 또한 USN에서는 네트워크적인 제약이 발생할 수 있다. 각 센서노드들이 소스노드이자 라우터의 역할을 수행하는 애드 혹 환경이기 때문에 센서노드의 고장은 바로 네트워크의 분리를 초래하여 급기야는 네트워크 전체의 고장을 야기할 수 있다. 따라서 구성된 네트워크의 유지 또한 관찰자의 요구에 따라 지속적으로 이루어져야 한다.

2.2 키관리기법

기존에 제안된 유니캐스트 및 멀티캐스트 환경에서 통신을 원하는 개체(peers)들간의 키교환 및 갱신기법에 대해 사전분배형, 중앙분배형[3, 4, 5], 분산분배형[6]으로 나누어 각각의 연구들에 대해 살펴보고 특히, 센서네트워크에서 제안된 키관리 기법에 대한 연구[2, 7]에 대해 알아본다.

키관리를 위해 키교환을 원하는 개체 A, B가 존재하며, 키교환 절차가 성공시에 두개체가 갖는 공통키 K_{pair}를 갖는다.

사전분배형 중 한 가지는 가장 간단하고 에너지 효율적인 방식으로 네트워크가 형성되기 이전에 미리 허가된 센서노드들에게 공통된 키를 분배하는 방식이다. 또는 한 개 이상의 키재료를 가질 수도 있다. 다른 한가지는 센서노드들간의 모든 가능한 공유키를 오프라인상에서 미리 계산하는 방식이다. 일단 네트워크가 형성되면 각 노드들은 오직 상대노드의 상대노드의 식별정보만을 알면 안전하게 통신이 가능하다. 그러나 이 방법은 일단 네트워크가 형성되고 나면 새로운 노드의 추가 등이 불가능하고 센서네트워크에서의 통신을 위해서 각 노드가 다른 노드들의 키정보를 각각 유지하고 있어야 하므로 융통성과 확장성이 부족하다.

중앙분배형으로 대표적인 방식은 커버로스[3, 8]이다. 커버로스 프로토콜은 두 센서노드간의 비밀 공유키를 생성하기 위해 KDC(Key Distribution Center)를 이용한다. 인증된 노드는 KDC로부터 티켓을 발행받을 수 있으며 그 티켓은 세션키 K_{pair} 정보를 담고 있다. 따라서 KDC로부터 허가를 받은 노드는 티켓을 이용하여 다른 인증된 노드와 세션키 교환을 기밀하게 할 수 있다.

분산분배형인 패어와이즈기법[6]은 각 센서노드가 공개키 기반의 인증서를 가지고 있다. 센서노드 B는 노드 A에게 자신의 인증서 정보를 보낸다. 노드 A는 B의 인증서를 검증하고 노드 A는 세션키를 생성하여 B의 공개키를 이용하여 세션키를 암호화 하고 자신의 디지털서명과 인증서를 함께 붙여 B에게 보낸다. 노드는 자신이 받은 세션키를 이용하여 넌스(Nonce)값을 암호화하여 A에게 보내고 A는 B와 교환된 세션키를 확인한다.

USN에서의 LEAP[7]은 사전분배형과 분산분배형을 조합한 방식으로 오프라인상의 초기의 키값을 이용하여 각 통신영역에 따라 다른 키를 설정하도록 한다. 이웃 노드와의 패어와이즈키, 하나의 클러스터상의 클러스터키, 그리고 싱크와의 통신을 위한 개인키와 전체 네트워크에서 사용할 그룹키를 가지고 있다. LEAP은 일부노드의 키정보 노출이 전체 네트워크의 보안에 영향에 끼치치 않게 하기 위해서 제안되었다.

SPINS[2]의 경우, 중앙집중분배형 방식으로써 싱크에 의해 주기마다 현재 주기의 키가 센서노드들에게 브로드캐스팅되며, 각 브로드캐스팅된 마스터키는 센서노드에서 MAC함수를 통하여 인증키, 암·복호키를 생성하여 키값을 갖는다. 이는 중앙집중형분배기법[3-12, 19, 20]의 성격을 가지면서 에너지 소모를 고려하여 인증서나 디지털서명값 등을 생성하지 않고, 단순한 MAC함수를 이용하여 킷값을 생성하도록 하였다.

그림 2는 SPINS의 키관리기법을 나타낸 것으로, $F_k(x)$는 MAC함수에 키값 K를 가지고 x를 입력으로 하여 새로운 키를 도출하는 함수이다. 마스터키 Ki는 싱크에 의해주기적으로 갱신되며, 마스터키는 암호화키와 인증키로 $F_k(x)$에 의해 새로이 생성된다.

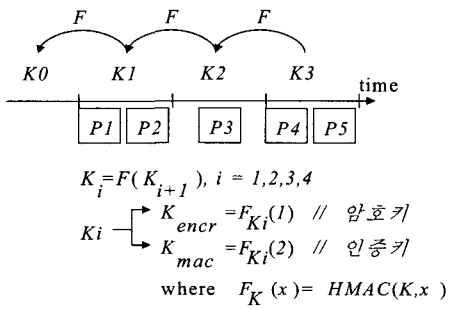


그림 2. SPINS 키관리기법

2.3 기존연구의 비교분석

기존연구들에 대한 장단점을 분석하고 제안하는 기법들이 가져야 하는 표 1에서의 요구사항에 대하여 비교분석하도록 한다.

표 2는 키관리시의 각각의 고려해야 할 사항에 대하여 각각의 기존 연구들에 대해 비교 분석하였다. 사전분배형은 에너지효율성과 메모리 효율성은 우수하나 적시성, 기밀성과 전방향보안성을 만족시지 못한다.

반면 중앙집중형은 중앙서버에 의해 적시성 및 기밀성 그리고 전방향 보안성을 만족시켜준다. 하지만, 키교환시 소모되는 에너지측면을 볼때, 두 노드와 신뢰기관의 세 기관에서 각각 키교환을 위한 제어메시지를 평균 메시지 처리 소모 에너지 $Eu(P)$로 평균 N개 처리해야 한다고 할 때, 센서노드 2개의 노드가 메시지 처리시 소모하는 비용은 $2N \cdot Eu(P)$이며 통신비용은 단위 메시지 송수신시 소모되는에너지 $Eu(T,R)$에 대하여 $2N \cdot Eu(T,R)$의 에너지를 소모한다. 또한, 메모리 또한 키를 생성 및 갱신하는데 필요한 부분비밀정보를 V개를 가져야 한다.

표 1. 키관리시의 요구사항

설명	설명
에너지 효율성	전체에너지에 대한 소모에너지의 정도
적시성	현재 키정보가 외부 공격자의 간섭없이 정상적인 노드와의 통신에만 사용여부
기밀성	키정보의 공격자로부터의 은닉여부
전방향 보안성	현재 키정보와 이전 키정보 노출과의 독립여부

표 2. 키관리기법의 비교

	요소	사전분배	중앙집중	분산분배	SPINS
적시성	키갱신주체	없음	중앙서버	노드	싱크
기밀성	1-키노출위험도	낮음	높음	보통	보통
전방향 보안성	p=1-이전의 노출된 키로 현재의 키의 유도가능확률	0	p>0	p>0	p>0
에너지 효율성	노드당 처리 비용	1	2N · EU (P)	M · N · EU(P)	EU(P)
	노드당 통신 비용	0	2N · EU (T,R)	M · N · EU(T,R)	EU(R)

분산분배형은 중앙서버의 간섭이 없기 때문에 적시성, 기밀성과 전방향보안성 모두 센서노드에서 이루어지는 형태라고 볼 수 있다. 이에 따라 메시지 처리비용은 평균 M개의 노드와의 부분비밀정보 교환을 통해 키의 확립 및 갱신이 이루어진다면, $M \cdot N \cdot Eu(P)$이고 통신비용은 $M \cdot N \cdot Eu(T,R)$이다. 또한 메모리에 대해서는 이웃노드와의 교환된 비밀부분정보를 유지하고 있어야 하므로 중앙집중형보다 많은 개수의 부분키정보를 유지하고 있어야 한다.

SPINS의 경우는 여러 보안성을 만족시켜주면서 싱크에 의해 키가 갱신되고 이를 싱크가 각 센서들에게 보내주면 센서가 검증하는 형태로 키를 교환하기 때문에 하나의 키정보에 대해서는 처리해주면 되므로 $Eu(P)$이고 싱크로부터 현재 주기의 키를 전송받으므로 단위 데이터의 수신시 소모에너지 $Eu(R)$의 에너지만을 소모하면 된다.

따라서, 적시성의 측면에서는 중앙집중형, 분산분배형, SPINS 모두 만족해주었으며, 기밀성의 측면에서는 중앙집중형이 가장 높은 성능을 보인다. 그리고 전방향 보안성에 대해서는 사전분배형만이 만족시켜주지 못하고, 에너지효율성 및 메모리효율성의 측면에서는 사전분배형이 가장 좋았으며, 중앙집중형이 분산분배형 $2/M$의 에너지만을 소모하고 더 적은 메모리를 요구함으로써 더 좋은 성능을 보였다. SPINS의 경우는 여러 보안성을 만족시켜주면서 중앙집중형에 비해 처리해야 하는 키정보 및 통신에너지가 적다는 것을 볼 수 있다.

따라서, 센서네트워크의 특성상 공통의 목적을 가지고 구성되는 네트워크이기 때문에 온라인 상의 키교환보다는 센서노드의 에너지효율성 및 메모리효율성의 측면에서 오프라인상으로 사전에 분배하는 것이 좋으며, 악의의 노드로부터의 키정보의 노출에 의한 전체네트워크의 노출을 방지하기 위하여 전체 네트워크의 키정보의 동기화를 지양하는 온라인상의 키갱신기법을 요구한다.

Ⅲ. 에너지 효율적인 동적 키관리기법

본 장에서는 기본적으로 에너지효율성을 고려하고, 키관리기법에서는 특별히 전체 센서노드간의 키정보 및 시간의 비동기성을 지향하였다.

3.1 동적인 키관리기법

데이터의 안전한 전달을 위해서는 인증과정과 암복호과정이 필요하다. 이에 따라 인증키와 암복호키

2개가 필요하다. 이를 위해 인증키이면서 초기의 세션키의 역할을 수행할 초기 비밀정보값 K_s는 센서네트워크가 구성되기 전 오프라인을 통해 미리 센서노드가 가지고 있다고 가정한다. 이는 센서네트워크의 고유한 특성으로 싱크를 가졌다는 것과 사전에 공통의 목적을 가지고 센서네트워크 구성된다는 점을 감안하여 사전에 키를 생성하기 위한 비밀정보를 갖도록 하였다.

3.1.1 키생성 기법

오프라인상에서 분배받은 K_s정보를 이용하여 키 갱신 절차를 통해서 인증키(K_j)를 해쉬함수를 통해 생성하며, 수식 1에 의해 생성된 초기의 세션키 $K_{session}$를 이용하여 싱크로부터 수신한 부분키 정보 K_c를 수신함으로써 그림 3과 같이 암복호키를 생성한다. 본 알고리즘은 센서노드 중 자신이 소스 노드가 되어 암호화가 요구되는 경우에만 동작한다. 그림 3의 ***EncryptionKeyCreation***알고리즘에서 사용된 명령함수의 정의는 다음과 같다.

- Verify : 송신지로부터 전달된 메시지에 대한 인증값의 검증을 수행하는 명령
- $E(x)$: x를 대칭키암호화 방식으로 암호화함수
- $D(x)$: 암호화된 x를 대칭키암호화 방식으로 복호화함수, $E^{-1}(x)$

그림 3의 암호키 생성 알고리즘은 수식 4에 의해 얻어진 현재 주기의 인증키 K_j와 세션 신호에 함께 수신된 비밀키정보 K_c를 이용하여 해쉬함수를 적용함으로써 암호키를 생성한다. 암호키는 데이터의 암복호화가 필요한 경우에 그 노드에서만 수행된다.

$$K_{session} = H(K_s, sq),\ sq \in \mathbb{N} \qquad (1)$$

수식 (1)에서 sq는 세션 신호의 순서로 1을 시작으로 하여 세션 신호가 올때마다 1씩 증가하는 값으로 싱크와 단말간의 세션키로 동작한다. 초기의 세션키로서 세션 신호에 포함된 비밀정보를 센서노드들에게 전달하거나, 세션 메시지에 대한 인증 및 검증을 수행한다.

$$i = (descript|EKsession(Kc|n)|\ HMAC(Ksession, descript|Ks|Kc|n)) \qquad (2)$$

알고리즘 1. 암호키생성 알고리즘

입력 : 세션 신호 i와 K_s

출력 : 암호키 K_o

Procedure ***EncryptionKeyCreation***(i, K_s)
  p := **Verify** the signal i
  If p = T **Then** // T is true
  Begin
    $K_c := D_{Ksession}(E_{Ksession}(K_c))$
    // Decrypt the encrypted K_c, $y=E(x)$ iff, $x=D(y)=E^{-1}(y)$
    $Ko := H(K_j, K_c)$
   // K_j can be obtained by equation 3-4
  End
  Else $K_o := 0$ // 0 means invalid.

{Ko is an encrytion key for this period. If verification of transaction i is failed, Ko is invalid}

그림 3. 암호키 생성알고리즘

수식 (2)는 싱크로부터 세션시작을 알리는 세션 신호를 나타낸 것으로, 요청정보(descript)와 세션키 $K_{session}$를 갖는 암호함수 $E_{Ksession}(x)$을 이용하여 센서노드들의 데이터 전달시 암호키의 부분비밀정보 K_c와 갱신시 사용할 n값을 암호화하고 인증함수 $HMAC(K_{session}, x)$함수를 이용하여 데이터에 대한 인증메시지를 붙여서 각 싱크로 전달한다. $HMAC$함수는 키 $K_{session}$로 x의 해쉬값을 만들어낸다. 수식 (2)와 같은 세션 신호를 수신 받은 센서 노드들은 본 정보를 저장하고 있으며 자신이 소스이거나 중계노드 일때, 자신의 키비밀정보 K_s에 대한 현재 주기의 키값과 비밀키정보 K_c를 단방향 해쉬함수 H에 적용하여 암복호시 사용할 키 K_o를 생성한다. 수식 (3)은 암복호키 K_o를 생성하는 과정을 나타내고 있다.

$$K_o = H(K_j, K_c) \qquad (3)$$

이때, 오프라인상으로 분배되는 키비밀정보 K_s를 가지고 있어야만이 정상적인 검증절차와 복호화를 통해서 키생성 및 갱신을 위한 비밀정보를 얻을 수 있으므로 악의의 노드가 온라인상의 교환되는 데이터를 통해 키정보를 획득하려는 행위를 무력화 시킬 수 있다.

3.1.2 키갱신기법

그림 4와 수식 (4)는 인증키(K_j)를 갱신하는 과정을 보여주고 있다. 키의 갱신과정은 네트워크 전체에서 일어나지 않고 소스노드이거나 데이터의 전달이 필요한 경우에 수행된다. 이는 각 센서노드의 불필요한 에너지 소모를 막고 네트워크 전체적인 키 갱신을 통한 전체 네트워크의 키정보의 동기화를 지양하고, 각 소스마다 데이터를 전달하기 시작하는 시점부터 키갱신을 수행하게 함으로써, 일부경로에서의 키정보의 노출이 전체 키정보의 노출을 초래하지 않도록 하여 키정보의 보안성을 강화하였다.

$$K_j = F^{n-j+w}(K_s)$$
where $F(K_j) = H(K_{j+1}, K_s)$ *and* $j \in \mathbb{N}$ *is period value,* $n \in \mathbb{N}$ *is obtained by transaction i* (4)

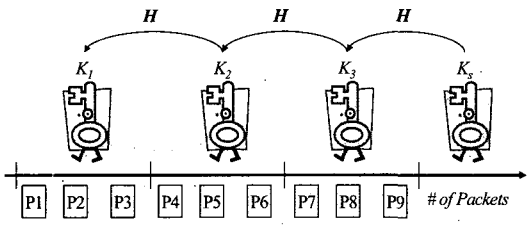


그림 4. 키갱신기법(n=9, T=3)

수식 (4)에서 보듯이 싱크로부터의 세션 신호에서 수신한 n정보를 가지고 현재 세션의 전체 갱신 횟수를 결정할 수 있다. n은 싱크가 어플리케이션의 특성에 따라 결정하며, 수식 (5)에서와 같이 센서로부터 데이터를 받을 전체시간과 요청정보에 담겨진 수신주기시간을 이용하여 설정할 수 있다.

$$n = \left[\frac{t_{unit}}{t_{session}} + 1\right] \quad (5)$$

수식 (5)에서 t_{unit}는 싱크가 센서로부터 수신받을 주기시간이며, $t_{session}$은 싱크가 센서로부터 데이터를 수신받을 예정된 전체 수신시간이다. 이를 이용하여 본 세션동안 전달될 데이터의 개수를 결정할 수 있다.

최종적인 갱신주기는 데이터를 보낼 센서노드가 결정하며 이때, n값과 자신 및 이웃노드가 가지고 있는 에너지수준을 이용하여 키정보의 갱신주기를 계산한다. 키갱신주기는 수식 (6)과 수식 (8)에 의해 결정한다. 키갱신주기를 결정짓는 요소로는 적용 어플리케이션의 요구 보안 수준과 각 센서의 잔존 에너지량을 이용한다. 이 또한 센서노드의 에너지 상태를 고려함으로써 에너지의 소모 정도를 조절하기 위함이다. 데이터를 보낼 데이터가 있는 노드들은 자신의 데이터 주기를 계산 후에 이웃노드의 P_E정보와 자신의 P_E정보를 가지고 주기 T를 수식 (8)과 같이 계산하여 자신의 이웃노드에게 첫 번째 데이터를 보낼때, 함께 첨부하여 전송하므로써 주기를 알린다. 해당 주기는 한소스가 일련의 메시지를 싱크에게 전달할 때까지 바뀌지 않는다.

$$SL = N \pm [k]$$
where $N > k$ *and* $N,\ k \in \mathbb{N}$ (6)

수식 (6)에서 주기를 결정짓기 위해 보안강도를 세단계로 나누어 중간단계 정도의 어플리케이션에서 요구하는 한 주기의 데이터의 개수를 N개라고 할때, 강한 보안을 요구하는 어플리케이션은 k만큼을 빼줌으로 주기를 빠르게 하고 약한 보안을 요구하는 어플리케이션은 k만큼을 더함으로 주기를 길게 하여 갱신회수를 줄이도록 한다. 여기에 각 센서노드들이 가지는 잔존 에너지량을 고려하여 잔존에너지량이 많으면 같은 어플리케이션이라도 주기를 조절해줄 수 있다. 주기의 설정은 소스노드에서 수행하며 이웃으로부터 받는 수식 7과 같은 에너지확률(P_E)값을 기준으로 설정한다.

$$P_E = \frac{1}{E_T + E_c + 1}$$
where ET and EC are maximum and current energy level, (7)

$$T = S_L \cdot (2 - P_E) \quad (8)$$

수식 (6)와 수식 (8)에서 갱신주기의 기준은 현재의 키로 전달할 수 있는 데이터의 개수로 정의하였다. 이는 일정 시간 간격으로 키갱신시 발생하는 시간 동기화 문제를 해결하기 위함이다.

키정보의 갱신시 단방향 해쉬함수를 적용함으로써 해쉬함수의 단방향 성질에 의해 현재의 키정보를 알아낸다 할지라도 다음에 사용될 키를 알 수 없도록 하여 전방향 보안성 및 키에 대한 비구별성을 제공할 수 있다.

그림 4는 n이 9이고 T가 3인 경우에 키를 갱신하는 경우를 나타낸 것이며, 주기와 전체 n에 의해 총 3번의 갱신과정이 필요하므로 j는 1부터 3까지 증가

하면서 단방향 해쉬함수에 의하여 갱신되는 것을 나타내고 있다. 이때, $K1$의 키를 악의의 노드가 알아내었다고 할지라도 함수 H의 단방향 특성으로 인해 $K1(=H(K2,K_s))$이므로 $K2$의 정보를 알아낼 수 없다.

3.2 보안성분석

센서네트워크상에서 요구하는 보안조건과 공격유형을 정의하고, 각 공격유형에 따라 각각의 보안성을 만족할 수 있음을 증명하도록 한다. 센서네트워크에서 요구하는 보안조건으로는 데이터의 무결성 및 기밀성과 데이터와 키정보에 대한 적시성이 있다. 키정보에 대한 적시성을 보장하기 위해서는 키의 비구별성과 전방향 보안성을 만족해야 하며, 데이터에 대한 적시성을 보장하기 위해서는 현재 데이터에 대한 시간정보 등의 추가적 정보를 이용하여 현재 새로운 데이터임을 보장해야 한다. 이와 함께, 센서노드의 고장 등으로 인한 돌발상황에도 정상적으로 운용될 수 있는 가용성을 요구한다.

공격유형에는 크게 단-대-단 단말간의 교환시의 데이터에 대한 공격과 싱크까지의 데이터 전달을 방해하는 라우팅 공격으로 나눌 수 있다. 표 3은 각 경우에 대하여 가능한 공격유형을 기술하였다.

표 4는 각 공격성공에 대한 어드밴티지(Advantage) 및 인시큐리티(InSecurity)를 정의하기 위한 기호이다.

표 3. 공격유형

공격유형	설명
CAM-ATK	악의를 가진 제3의 노드 AN은 싱크가 보내는 세션 신호 q를 인근의 센서노드에게 싱크로 위장하여 보낸다. 각 세션 신호 q를 받은 센서노드는 이에 대한 응답으로 인증된 메시지를 악의의 노드 AN에게 보낼 수도 있다. 이를 통하여 현재의 키정보 Kj를 획득하고자 시도한다.
AKC-ATK	AN은 CAM-ATK를 통하여 현재의 키정보 Kj를 획득하였을 때, 이후 사용될 Kj+1를 알아내기 위하여 Kj=Fi-t(Ka)가 되는 어떤 키 Ka와 t를 추측하여 전체 키체인 정보를 알아내고자 시도한다.

표 4. 기호정의

기호	정의
$Adv_{S,C}^{ATK}(AN,x)$	채널 C상에서 보안기법 S에 대하여 악의 노드 AN이 x를 가지고 ATK공격에 성공할 어드밴티지
$P(ATK_{atk}^{x}(x)=1)$	x를 가지고 공격에 성공할 확률
$INSec_{S,C}^{ATK}(x,y)$	채널 C상에서 보안기법 S에 대하여 x,y를 가지고 ATK공격에 성공할 최대 어드밴티지
ε	무시할 만큼 작은 값, $0<\varepsilon<1$

정의 1. 키의 적시성－비구별성

센서노드와 싱크간의 채널 C상에서 제안하는 보안기법 S에 대하여 AN의 어드밴티지(Advantage) Adv와 인시큐리티(Inscurity) $INSec$는 다음과 같이 정의한다.

$$1)\ Adv_{S,C}^{CAM}(AN,m) = |P(CAM_{atk}^{m}(1^m)=1) - P(CAM_{atk}^{r}(r)=1)|$$

$$2)\ INSec_{S,C}^{CAM}(m,r) = \max\left\{Adv_{S,C}^{CAM}(AN,m)\right\}$$

여기서, m은 공격자 AN의 세션 신호 q에 의해 도출된 임의의 비트열이며, r은 어떤 규칙도 없이 임의로 생성된 비트열이다.

정리 1. 키의 적시성－비구별성

CAM-ATK에 대하여 $INSec_{S,C}^{CAM}(m,r) \le \epsilon$, 여기서, ε은 무시할 만한 값이다.

증명. 제안하는 기법의 현재 키정보는 단방향 해쉬함수 H에 의해 생성된다. 따라서 해쉬함수이 갖는 성질에 의해 입력값과 상관성이 존재하지 않는 임의의 값을 출력한다. 본 공격의 어드밴티지(Adv)는 현재 키정보를 알아낼 수 있는 확률과 불규칙한 비트열을 생성하여 키정보를 알아내는 확률값의 차이이며 해쉬함수 H의 성질에 의해 CAM-ATK에 의해 인시큐리티(InSec)은 무시할 만하다. 즉, CAM-ATK공격에 의해 키정보를 획득할 확률은 매우 희박하다.■

정의 2. 키의 적시성－전방향 보안성

AKC-ATK의 어드밴티지 Adv와 인시큐리티 $InSec$은 다음과 같이 정의한다.

$$1)\ Adv_{S,C}^{AKC}(AN,t,K_a) = |P(AKC_{atk}^{K_a,t}(F^{i-t}(K_a)=K_j)=1)|$$

$$2)\ INSec_{S,C}^{AKC}(K_a,t) = \max\left\{Adv_{S,C}^{AKC}(AN,t,K_a)\right\}$$

여기서, AN이 획득한 현재 주기의 키정보 K_j와 임의로 생성한 비트열 t와 키값 Ka이다.

정리 2. 키의 적시성－전방향 보안성

AKC-ATK에 대하여, $INSec^{AKC}_{S,C}(K_a, t) \leq \varepsilon$

여기서, ε은 무시할 만한 값이다.

증명. 정리1에 의해 CAM-ATK에 대한 *INSec*를 p라고 하자. 즉, $p<\epsilon$이다. 이와 함께 임의의 비트열 t를 입력값으로 하여 단방향 해쉬함수를 적용하여 K_j와 일치할 *INSec*값을 q라고 할 때, AKC-ATK의 *INSec*은 정의에 의해 $p \cdot q$로 나타낼 수 있으며 $0<p,q<1$이므로 $pq<p<\epsilon$이다. 따라서, $INSec^{AKC}_{S,C}(K_a, t) \leq \epsilon$이다. ■

제안하는 키관리 기법과 SPINS 모두 키정보의 적시성과 전방향 보안성, 그리고 기밀성을 만족시킨다. 하지만, SPINS의 경우 중앙의 베이스스테이션으부터 주기적으로 수신하는 공통의 키정보를 사용하기 때문에 하나의 센서네트워크가 시간이 동기화되어 있어야 한다는 문제와, 일부분의 키정보 노출이 전체의 네트워크의 키정보 노출을 야기할 수 있는 위험성이 있다.

그러나, 제안기법의 경우는 사전에 센서노드가 가진 비밀정보와 싱크로부터 수신한 키정보를 이용하여 새로운 키를 생성해내며, 생성해내는 키정보의 갱신 주기 또한 동일키로 보낼 수 있는 패킷의 개수로 정의하여 시간동기화 문제를 해결할 수 있었다. 또한 주기값은 송신노드가 이웃노드가 가진 에너지 수준을 고려하여 산정함으로써 패킷을 보내는 송신노드마다 마른 주기를 가짐으로써 일부 키정보의 노출이 전체 네트워크에서 사용되는 키정보의 노출로 이어지는 것을 방지하였다.

Ⅳ. 성능분석

제안하는 보안기법의 에너지 효율을 분석하기 위하여 키관리 기법과 인증기법을 적용하기 위해 추가적으로 소모되는 에너지량을 측정하고 제안하는 기법을 적용하여 데이터를 전달시 전체 소모되는 에너지에 대하여 보안기법의 적용시 소모되는 에너지량을 비교함으로써 에너지 효율을 분석하고, 이를 전체 센서네트워크에서 소모되는 에너지량과 소스노드에서의 에너지량과 중계노드에서의 에너지량을 각각 분석함으로써 보안기법을 적용시의 에너지 효율을 알아보고자 한다. 마지막으로 SPINS[2]와의 에너지 소모정도를 비교함으로써 보안기법 적용을 위한 추가 에너지 소모가 크지 않음을 분석하였다.

4.1 네트워크 및 시스템 환경

실험을 위한 네트워크 환경은 100개의 센서노드와 한개의 싱크로 구성되며 센서노드로부터 싱크까지는 배치에 따라 직접 통신을 할 수 있거나 센서노드들을 중계노드로 하여 여러홉을 경유하여 싱크로의 데이터의 전달이 가능하다. 네트워크 가정사항은 SPINS와 동일하며 적용된 암호 알고리즘은 표 7과 같으며 표 5에서와 같이 단위 데이터의 크기는 128 바이트로 가정한다.

표 5. 네트워크 실험환경

요소		설정값
토폴로지	노드 개수	100노드
	영역 크기	500 x 500좌표, 단위 1
작업부하	싱크 근접노드 (라우팅)	90번 노드
	단위 패킷크기	128 Bytes

싱크는 일반 컴퓨터 등과 같이 대용량의 시스템으로 가정하며 싱크노드는 표 6와 같이 WINNG[9]의 Sensoria 노드를 기준으로 실험하였다. 각 센서노드는 10kbps의 송수신률로 비트당 각각 21μJ과 14μJ을 소비하므로 128byte의 데이터를 송수신시에는 약 0.021J과 0.014J을 소모한다. 각 노드가 Sleep모드시의 소비전력은 고려하지 않았다.

표 6. 시스템 실험환경

요소	설정값
초기에너지	26kJ(7.2 volt battery pack)
송/수신기준전력	210mW/140mW
비트당 송/수신 소모에너지	21μJ/14μJ

표 7과 같이 메시지를 보내고자 하는 모든 소스는 1KB의 메시지를 보낸다고 가정하였으며, 시스템 환경은 표 6의 기준에 따른다. 그리고 인증기법으로서 HMAC함수를 적용하고 이를 실행하는데 필요한 단위 블록 512비트에 대한 소모 에너지와 128비트 블로의 암복호시 소모되는 에너지를 NABI 기술보고서[9]에 근거하여 설정한다.

표 7. 보안기법에서의 실험 설정값

요소		설정값	비고
전체 메시지 크기	Messge_Size	12800B	B : Byte
암복호 에너지	ENC(Packet) /DEC(Packet)	1.15μJ /128bit-block	AES 알고리즘 MIPS R4000
해쉬 에너지	MAC(Packet)	1.9μJ /512bit-block	입력 : 512bits HMAC with MD-5 MIPS R4000

표 8은 제안기법을 분석하기 위해 사용될 변수 및 기호를 정의하였다.

표 8. SPINS와 제안하는 키관리 및 인증기법에서의 변수정의

변수정의	설명
E_{Pro}	소스노드 및 중계노드에서 인증 및 키관리시 소모되는 전체 에너지
S_{Pro}^{Auth}, s	하나의 소스노드에서의 인증시 소모에너지, 소스노드의 개수
IAuth,r	하나의 중계노드에서의 인증시 소모에너지, 중계노드의 개수
T	설정된 주기의 크기
|Mtotal|	전체 패킷의 개수
Msize	전체 데이터의 크기
K_{Pro}^{E}	하나의 센서노드에서 키갱신시 소모에너지
HE	512비트당 해쉬함수의 단위 소모에너지
TE	비트당 전송시 단위 소모에너지
RE	비트당 수신시 단위 소모에너지
j	현재 주기값
ESPINS	소스노드 및 중계노드에서 인증 및 키관리시 소모되는 전체 에너지
S_{SPINS}^{Auth}	하나의 소스노드에서의 인증시 소모에너지, 소스노드의 개수
K_{SPINS}^{E}	하나의 센서노드에서 키갱신시 소모에너지

$$E_{Pro}=\sum_{i=1}^{s}S_{Pro}^{Auth}+\sum_{i=1}^{r}I_{Auth}+\sum_{i=1}^{r+s}\left|\frac{|M_{total}|}{T}\right|K_{Pro}^{E} \quad (9)$$

$$S_{Pro}^{Auth}=\sum_{i=1}^{|M_{total}|}(2[\frac{M_{size}}{512}+1]\cdot H_E+T_E) \quad (10)$$

$$I_{Auth}=\sum_{i=1}^{|M_{total}|}(2[\frac{M_{size}}{512}+1]\cdot H_E+R_E+T_E) \quad (11)$$

$$K_{Pro}^{E}=\sum_{i=j}^{\frac{|M_{total}|}{|T|}}(\frac{|M_{total}|}{|T|}-j+1)H_E \quad (12)$$

수식 (9)에서는 키관리 및 데이터 인증에 소모되는 에너지를 나타낸 것으로, 소스노드에서의 인증시 소모되는 에너지(S_{Pro}^{Auth})와 중계노드들이 소모하는 인증시 소모에너지(IAuth)의 합과 소스노드 및 중계노드에서의 키관리시 소모에너지(K_{Pro}^{E})의 합으로 나타낼 수 있다. 수식 (10)은 소스에서 데이터 인증시 소모하는 에너지로 패킷의 개수 및 메시지 블록의 수만큼 인증메시지를 만들고, 다시 패킷단위 인증메시지를 생성해야 하므로 단위 해쉬 에너지 H_E의 4배의 에너지가 소모되며, 데이터를 다음노드로 전달하는 단위송신에너지를 소모한다. 수식 (11)은 중계노드에서의 인증시 소모에너지로 인증값에 대한 검증 후 다시 인증값을 생성하여 다음 노드로 전달하기 때문에 4배의 단위 해쉬에너지와 단위 송·수신 에너지가 소모된다. 수식 (12)는 키관리 에너지로 자신의 주기마다 해쉬함수가 $\frac{|M_{total}|}{|T|}-j+1$만큼 수행되어야 하며, 이러한 과정을 전체 주기 동안 이루어져야 한다.

$$E_{SPINS}=\sum_{i=1}^{s}S_{SPINS}^{Auth}+\sum_{i=1}^{r}I_{Auth}+\sum_{i=1}^{N}\frac{T_{total}}{T}K_{SPINS^E} \quad (13)$$

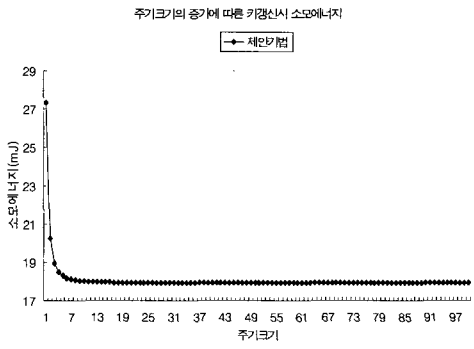
$$S_{SPINS}^{Auth}=\sum_{i=1}^{|M_{total}|}([\frac{M_{size}}{512}+1]\cdot H_E+T_E) \quad (14)$$

$$K_{SPINS}^{E}=[\frac{M_{size}}{512}+1]\cdot H_E+R_E \quad (15)$$

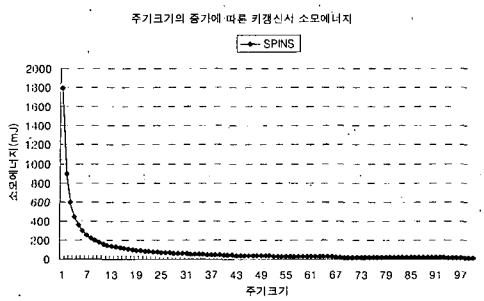
수식 (13)은 SPINS기법에서 키관리 및 인증을 수행할 때 소모되는 에너지를 나타내며, 시간의 주기를 사용하므로 키갱신횟수는 전체시간에서 주기시간을 나누어준 값이 되고 전체노드가 함께 갱신하므로 N개의 노드가 갱신을 수행해야 한다. 수식 (14)는 SPINS기법의 소스노드의 인증시 소모되는 에너지로 패킷에 대한 인증값을 생성하므로 한번의 HMAC과정이 필요하고, 패킷이 전달하는데 필요한 송신에너지가 소모된다. 수식 (15)는 하나의 노드가 한번의 키갱신시 소모되는 에너지값으로 싱크로부터 킷값을 수신받고 검증하는 과정에 필요한 에너지가 소모된다.

4.2 실험결과 및 분석

그림 5는 하나의 노드가 키를 갱신하는데 걸리는 시간을 분석한 그래프이다. 제안한 기법의 전체 주기를 100개로 하고 SPINS의 경우 100초로 하여 주



(a) 제안기법

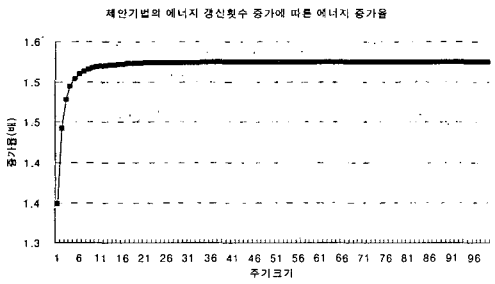


(b) SPINS

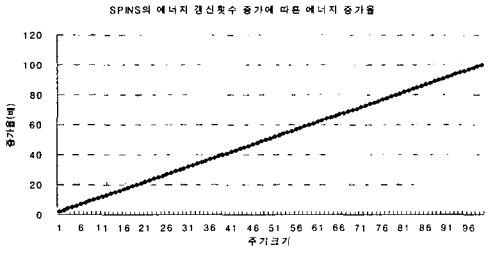
그림 5. 키갱신시 소모에너지

기를 1에서 100까지 증가함에 따라 키갱신을 위해 소모하는 에너지를 산출하였다. 여기서, 싱크와 노드의 홉은 동일하게 1홉으로 가정하였다.

제안하는 기법과 SPINS모두 감소하는 형태를 띄는 것은 한 주기의 간격을 점점 증가시킴으로 주기 간격이 증가하면 키갱신회수는 줄어들기 때문이다. 그러나, 제안하는 기법은 주기가 증가함에 따라 [17.92,27.33]구간에서 에너지 변화를 보였으나, SPINS는 [17.92, 1792.19]구간의 에너지 변화를 보임으로 제안하는 기법은 에너지의 최대변화가 약 9.41mJ이고 SPINS는 1774.27mJ를 보였다. 이는 갱신주기가 늘어남에 따라 소모에너지가 제안기법보다 평균 약 5배 정도 더 많은 변화값을 보이며 두 기법간의 에너지의 차이는 최대의 경우 주기가 1일때, 약 1764.87mJ까지 보인다.



(a) 제안기법

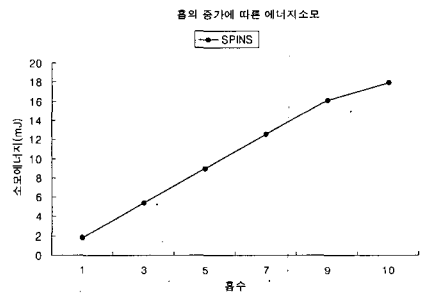


(b) SPINS

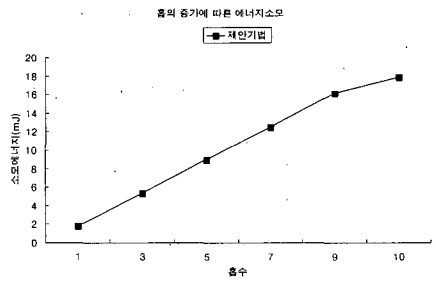
그림 6. 주기크기의 증가에 따른 에너지증가율

그림 6은 주기의 간격을 증가시키면서 주기가 1일때의 에너지 소모의 증가율을 나타낸 것이다. 주기가 약 12 이후 증가율이 로그적인데 반해, SPINS의 경우 주기 1일 때의 값에 비례하여 선형적으로 증가하는 것을 볼 수 있다. 제안기법이 로그적인 형태를 보이는 것은 주기가 변하여도 통신시 소모되는 에너지량이 일정하고 키갱신시 소모에너지만 증가하게 되므로 통신에너지에 비해 키갱신 에너지가 극히 적기 때문에 일어나는 현상이다.

주기변화에 따른 소모에너지의 차가 크다는 것은 보안요구가 강한 어플리케이션은 주기를 빨리하면 할수록 그에 비례하여 에너지가 키갱신 에너지가 선형적으로 증가하여 갱신주기의 설정상의 유연성이 부족함을 의미한다. 그러나, 제안기법의 경우 약 주기 12이후로 주기 1일때의 소모에너지와 그 이상의 간격의 주기일때의 차이가 급격히 변화하지 않으므로, 주기 설정시 에너지 소모를 크게 고려하지 않아도 된다. 또한 제안하는 기법의 증가율이 점점 감소되면서 수렴하는 형태를 볼 수 있는데, 이는 주기가 증가함에 따라 소요되는 증가에너지는 검증에너지와 해쉬함수를 처리하는데 소요되는 에너지이므로 그 값이 크지 않고, 또한 메시지 수신시 노드에서 소모되는 단위에너지가 2배 정도 커서 통신회수를 줄임으로서 증가율이 키갱신시의 증가율이 2배를 넘지 않는 결과를 알 수 있었다.



(a) SPINS



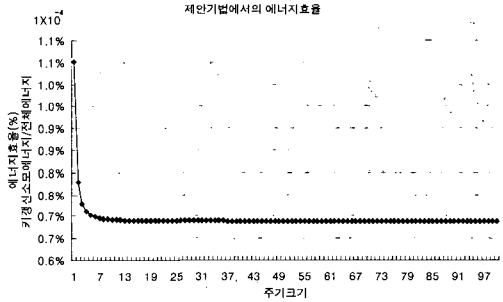
(b) 제안기법

그림 7. 중계노드의 증가에 따른 소모에너지

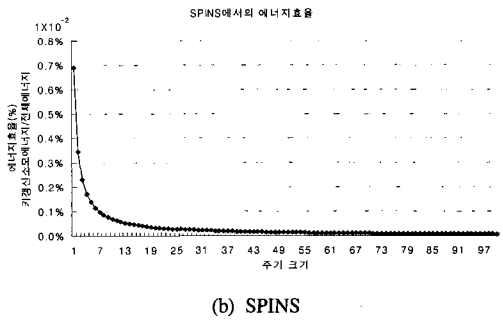
그림 7의 주기는 100으로 하여 키갱신회수를 1번으로 고정하고, 키갱신시 싱크에서 각 센서들로 전달되는 홉수를 10홉까지 증가시키면서 키갱신시의 소모에너지를 분석하였다. 그림 7에서 보듯이 거의 같은 값이 도출되었다. 이는 제안하는 기법이 키갱신시의 처리과정이 더 많은데도 불구하고 유사한 값을 갖는 것을 미뤄볼때, 센서노드의 통신시의 에너지 소모가 크므로 통신회수를 줄일 수 있는 기법의 에너지효율이 더 좋다는 것을 알 수 있다.

그림 8은 센서노드가 가지는 최대에너지에 대하여 키갱신시 소모되는 에너지를 나타낸 것으로, 그 값이 작을수록 효율이 좋음을 알 수 있다. SPINS의 경우 평균 0.4×10^{-3}%의 에너지 효율로 약 최소 0.7×10^{-4}%에서 최대 0.7×10^{-2}%까지 증가하는 것을 볼 수 있으며, 제안하는 기법의 경우 평균 약 0.7×10^{-4}%의 에너지 효율을 보임을 알 수 있다. 제안하는 기법이 평균적으로 약 5배의 더 높은 에너지효율을 보이고 있다.

SPINS와 제안기법 모두 해쉬함수를 기반으로 한 키갱신을 수행함으로써 전체 센서노드의 에너지 소모에 극히 작은 부분을 차지함을 알수 있으며, 제안기법은 통신에너지를 줄임으로써 주기간격의 감소로 인한 소모에너지의 선형적 증가를 방지하였다.



(a) 제안기법



(b) SPINS

그림 8. 키갱신시 소모되는 에너지의 효율성

SPINS에서의 전체 센서네트워크에서 키정보의 동기화로 인한 보안적 취약점을 해결하도록 각 소스별로 상이한 갱신주기를 설정하여 서로다른 키정보를 사용하여 데이터를 전달할 수 있도록 하였다. 또한 제안하는 키관리기법의 키교환 및 갱신시의 에너지효율이 더 좋으며, 갱신주기의 증가에 따른 증가율 또한 SPINS의 경우 선형적으로 소모되는 에너지도 함께 증가하나, 통신에너지를 일정하게 조정함으로써 제안하는 기법은 로그적 패턴으로 소모에너지가 증가함을 알 수 있다. 이러한 로그적 증가의 특성을 고려하여 싱크에서는 각 전체 패킷 개수에 따른 최적의 주기를 계산하여 주기를 결정짓는 중대한 요소인 N값을 전달하고, 각 소스에게 자신의 에너지확률을 고려하여 키갱신주기를 결정할 수 있다.

표 9. 키관리기법의 에너지효율

특성	제안기법	SPINS	실험결과
제안특징	-필요에 의한 키갱신으로 각 소스별 데이터가 전달되는 경로로 서로 다른키 사용 -키정보의 네트워크 동기화 지양		
에너지 소모	-해쉬기반 -연산에 의한 자체키갱신	-해쉬기반 -싱크로부터 주기적 키수신	주기크기별, 중계노드별 소모에너지 -제안기법<SPINS -통신소모에너지 감소
키갱신주기에 따른 소모에너지증가	로그적	선형적	주기크기별, 각 주기크기별 주기1에 대한 에너지 증가율 -제안기법<SPINS -통신소모에너지 감소
비고	SPINS의 경우 시간적 동기화 필요		

V. 결 론

본 논문에서는 USN네트워크 상에서 안전한 데이터 전달을 수행하기 위하여 키관리기법을 제안하였다. 제안한 기법은 공통적으로 에너지효율성을 높이는 것에 초점을 두었으며, 키관리기법에서는 전체 네트워크상의 키정보의 동기화를 지양하고자 하였다. 제안하는 키관리기법은 에너지효율성의 측면에서 통신횟수를 줄이고 센서노드내 처리량을 늘임으로써 SPINS보다 더 적은 에너지로 키갱신을 수행할 수 있었다. 또한 키갱신주기를 시간기반으로 하지 않고 데이터기반으로 설정하여 전체 네트워크의 시간동기화 문제를 해결하였다. 제안하는 키관리기법은 자신이 데이터를 보낼 것이 있는 경우에만 자

신 및 이웃노드의 에너지확률정보를 이용하여 자신에게 맞는 키갱신주기를 설정함으로써 소모에너지 정도를 고려하였다. 또한 필요시만 키갱신주기를 설정하고 이루어지므로, 센서노드가 가지는 에너지량과 데이터를 처음 보내는 시점에 따라 같은 시간이라도 서로 다른 키정보를 갖게 함으로써 키정보의 전체네트워크의 동기화문제를 해결하였다.

향후 실제 제안한 기법을 프로토콜로서 구현할 예정이며, 하드웨어적인 보안기법과 함께 적용함으로써 강건한 보안성을 제공하는 USN서비스가 가능할 것이라고 기대한다.

참 고 문 헌

[1] A. Perrig, R. Canetti, B. Briscoe, D. Tyger, and D. Song. "TESLA: Multicast Source Authentication Transform," Internet Draft, IETF, November 2000.

[2] A. Perrig, R. Szewczyk, J.D. Tygar, Victorwen and E. Cullter, "SPINS: Security Protocols for Sensor Networks," Wireless Networks, 521-534, 2002.

[3] Ateniese G., M. Steiner, and G. Tsudik. New Multiparty Authentication Services and Key Agreement Protocols. IEEE Journal on Selected Areas in Communication, 2000.

[4] B. Krishnamachari, D. Estrin, and S. Wicker., "The Impact of Data Aggregation in Wireless Sensor Networks.," International Workshop on Distributed Event-Based Systems, (DEBS '02), Vienna, Austria, July 2002.

[5] Bellare, M. and P. Rogaway, " Entity Authentication and Key Distribution, " Proceedings of Crypto93, LNCS 773, Springer-Verlag, 232-249, 1993.

[6] McGrew, D., and A., Sherman, "Key establishment in large dynamic groups using one-way function trees," TIS Report No. 0755, TIS Labs at Network Associates, Inc., Glenwood, MD(May 1998).

[7] Sencun Zhu, Sanjeev Setiam,Sushil Jajodia, "LEAP: Efficient Security Mechanisms for LargeScale Distributed Sensor Networks," CCS'03, 2003.

[8] Newman, B. and T. Ts'o, "Kerberos: an Authentication Service for Computer Networks", IEEE Communications Magazine, 33-38, September 1994.

[9] David W. Carman, P. Kruus and B. Matt, "Constraints and Approaches for Distributed Sensor Network Security," *NAI LABS Technical Report*, 2000.

윤 미 연 (Mi-youn Yoon) 정회원

2000년 2월 가톨릭대학교 수학과/컴퓨터공학과 졸업

2002년 2월 숭실대학교 컴퓨터학과 석사

2005년 8월 숭실대학교 컴퓨터통신 박사

2005년 6월~현재 한국정보보호진흥원 선임연구원

<관심분야> IP/신뢰/어플리케이션 멀티캐스트 통신 및 보안, 애드 혹 통신 및 보안, 센서 네트워크 통신 및 보안, IPv6 및 MIPv6 프로토콜, UMTS