극 좌표를 이용한 클러스터 기반 센서 네트워크의 키 관리 기법

홍성식* · 유황빈**

A key management scheme for the cluster-based sensor network using polar coordinated

Seong-sik Hong* · Hwang-bin Ryou**

요 약

센서 네트워크를 구성하는 센서 노드는 대부분 보안성이 낮으며, 낮은 연산 능력과 적은 저장 용량으로 효율적인 보안 알고리즘을 적용할 수 없다. 따라서 불법적인 노드의 침입을 억제할 수 없으며, 센서 노드의 전송 알고리즘만 알게 되면 전송되는 정보를 쉽게 도청할 수 있는 문제점을 갖게 된다.

본 논문에서는 센서 네트워크를 클러스터로 구분하고, 클러스터 내에서 센서 노드가 안전하게 정보를 전송할 수 있으며 불법적인 센서 노드의 침입을 억제할 수 있는 극좌표를 이용한 클러스터 기반의 센서 네트워크의 키 관리 기법을 제안한다. 제안된 기법에서는 클러스터 내에서 모든 센서 노드는 CH(Cluster Header)가 제공하는 피벗값을 기반으로 인증키를 설정하도록 하고 있다. 시뮬레이션 결과 기존의 페어와이즈(pair-wise) 기법에 비하여 키 관리측면에서 안전한 키 관리와 불법적인 노드의 침입을 억제할 수 있는 우수함을 증명하였다.

ABSTRACT

The level of security of most sensor nodes that comprise the sensor networks is low, but because of the low computing power and small storage capacity, it is even very difficult to apply a security algorithm efficiently to the sensor nodes. Therefore, preventing the join of an illegal node to a sensor network is impossible, and the transmitting information is easily exposed and overheard when the transmitting algorithm of the sensor node is known.

In this paper, we propose a group key management scheme for the sensor network using polar coordinates, so that the sensor nodes can deliver information securely inside a cluster and any illegal node is prevented from joining to the cluster where a sensor network is composed of many clusters. In the proposed scheme, all of the sensor nodes in a cluster set up the authentication keys based on the pivot value provided by the CH. The intensive simulations show that the proposed scheme outperforms the pair-wise scheme in terms of the secure key management and the prevention of the illegal nodes joining to the network.

키워드

sensor network, cluster, key management scheme

* 혜전대학 컴퓨터과 교수

** 광운대학교 컴퓨터소프트웨어학과 교수

접수일자 2008. 01. 21

Ⅰ. 서 론

센서 네트워크는 기존의 네트워크와는 달리 물리적인 전송 매체를 비롯한 네트워크 인프라가 구축된 상태에서 통신을 수행하는 것이 아니고, 인프라가 존재하지 않는 상태에서 각 단말기 상호간의 자발적인 라우팅으로 구성될 수 형태의 네트워크를 말한다[1]. 센서 네트워크에서 센서 노드의 위치는 미리 결정될 필요가 없으므로, 접근이 어려운 영역이나 재난 구조를 위한 응용을 위해 임의로 배치될 수 있다. 그러므로 센서 네트워크 프로토콜은 자가 구성 능력을 가지며, 센서 노드들이 서로 협력하여 동작한다. 센서 네트워크는 센서 노드와 베이스 스테이션(BS : Base Station)으로 구성된다[2].

센서 네트워크에서 요구되는 보안 요소는 이웃 노드와의 보안키를 안전하고 효율적으로 설정하는 것이다. 가장 간단한 방법은 모든 센서 노드가 동일한 그룹키를 공유하게 하는 방식이다[3]. 하지만 이 경우 하나의 센서 노드라도 공격자에게 포획되면 그룹키가 유출되어 안전하지 못하게 된다. 다른 극단적인 방법으로는 충분한 메모리 공간을 확보한 모든 센서 노드의 쌍마다 유일한 키들을 할당하는 방식이 있다[4]. 이 경우에는 임의의 센서 노드가 공격자에게 포획되어도 다른 센서 노드 간의 통신은 안전할 수 있다. 그러나 이 방법은 센서 노드의 저장 용량의 제약조건 때문에 비현실적인 방법이다.

또 다른 방식으로 공개키 기반구조 (PKI : Public key infrastructure)에 의한 공개키 방식을 사용할 수 있다[5]. 그러나 센서 노드가 보유한 자원의 제약 때문에 많은 자원을 요구하는 공개키 방식의 키 교환을 직접 사용할 수 없다[6]. 또한 포획된 센서 노드들의 비밀 정보가 유출될 경우 전체 센서 네트워크에 미치는 영향력이 매우 크다.

그러므로 전체 센서 네트워크를 클러스터 단위로 분할 구성하고 각각의 클러스터는 클러스터 헤더(CH : Cluster Header)가 자신이 속한 클러스터 내의 노드를 관리하는 클러스터 단위의 키 관리 기법을 적용하는 것이 보편적인 방식으로 되어 있다[7].

본 논문에서는 센서 네트워크를 클러스터로 구분하고, 클러스터 내에서 센서 노드가 안전하게 정보를 전송할 수 있으며 불법적인 센서 노드의 침입을 억제할 수 있는 극좌표를 이용한 클러스터 기반의 센서 네트워크의 키 관리 기법을 제안한다. 제안된 기법은 센서 노드에서 수집된 정보 전송은 CH를 거쳐 BS로 전송하도록 하며, 전송되는 정보의 기밀성을 위해 초기 설치 시 BS가 부여한 비밀키를 이용한 AES128 암호 알고리즘을 사용하여 암호화하여 전송하도록 한다. 또, 클러스터 내에서 모든 센서 노드는 CH가 제공하는 피벗값을 기반으로 인증키를 설정하여 CH에 접근하도록 함으로써, 클러스터 내에 있는 모든 센서 노드들과 안전하고 효율적인 정보 전송을 가능하게 하는 키 관리 기법이다.

Ⅱ. 관련 연구

BS와 클러스터 구조를 중심으로 중간에 aggregator를 두는 기본 구조를 기반으로 한 그룹키 관리 연구 중 Jing Deng, Richard Han,Shivakant Mishra의 연구가 있다[8]. 각 센서 노드는 사전에 BS와의 1대1의 비밀키를 갖는다는 가정 하에 단방향 해시 함수와 μTESLA를 사용하여 안전한 그룹키를 전달하기 위한 세 단계의 메커니즘을 제시하였다. 각 aggregator를 통해 BS로부터 효율적으로 키를 분배할 수 있으며 BS와 aggregator로부터의 메시지를 각 센서 노드가 인증할 수 있을 뿐 아니라, 각 센서로부터 전달된 메시지를 상위레벨에서 효과적으로 인증할 수 있는 방안을 제시하였다.

Sencun Zhu, Sanjeev Setia, Sushil Jajodia에 의해 제안된 LEAP[9]는 일부 노드의 노출이 근접 이웃 노드까지 노출시키는 위협을 최소화하기 위해 제안된 기법으로 노드, 페어, 클러스터, 그룹 단위로 키를 달리하는 다중 키 기법을 사용하고 있다. 이것은 노드의 집합 단위 별로 다른 키를 사용하며, BS로부터 공급되는 마스터 키를 통해 유도될 수 있는 특징을 가지고 있다. LEAP의 장점은 제안하는 이웃 노드가 보안 위협에 노출된 경우라도 노드의 보안을 유지할 수 있다는 점이다. 그러나 LEAP는 위치 정보를 반영한 지역적 인증 및 키 생성을 지원하지 않는다는 단점을 지니고 있다. 모든 노드들이 페어와이즈(pair-wise) 키를 생성할 수 있는 초기키를 동일하게 소유함에 따라 실제로 자신과 통신할 수 있는 범위에 있는 노드 외에도 BS로부터 초기키를 전송받은 모든 노드가 인증 및 키 생성을 시도할 수 있으므로 악의적인 공격자가 이 특성을 통해 공격을 시도할 수 있다는 문제점이 있다.

L. Eschenauer, V. Gligor[10] 는 센서 노드 간 페어와이즈 키 설정을 위해 제안한 프로토콜로 BS가 먼저 다량의

랜덤 키를 생성하여 이를 키 풀(pool)에 저장하고 키 풀에서 임의의 키 집합을 선택하여 키 링을 생성하여 이를 각 센서 노드에게 분배한다. 센서 노드들은 자신이 갖고 있는 키 링의 키 정보를 이웃 노드들에게 브로드 캐스팅 함으로써 통신 반경 내에서 자신의 이웃하는 노드들과 공유키를 찾는다. 이 구조는 노드의 개수가 매우 많더라도 수백 개 정도의 키로 기존의 페어와이즈 키와 동일한 안전성을 제공한다는 장점을 갖는다.

D. Liu, P. Ning[11]은 센서 노드 간 페어와이즈 키를 유도할 수 있는 다항식을 생성하여 분배하는 방식을 제안하였다. 이 방식은 실제 키 값을 사용하지 않고 키를 유도할 수 있는 다항식을 사용하기 때문에 임의 키 체인 분배 기법에서의 키 중복 현상을 방지할 수 있다. 그러나 n개의 키를 생성하기 위해 총 n번의 다항식 연산을 추가적으로 수행해야 하므로 다항식 링에 존재하는 다항식의 숫자가 많으면 많을수록 오버헤드가 커진다는 단점이 존재한다.

D. Liu, P. Ning[11]은 센서 노드들이 m × m 그리드 상의 행과 열이 교차하는 지점에 위치된다고 가정하고 페어와이즈 키를 생성할 수 있다. 셋업 서버는 2m개의 다항식을 생성하여 i 열 j 행에 있는 센서 노드에게 두 개의 다항식 $fi(x, y)$ 와 $fj(x, y)$ 를 배분하여 동일한 행 또는 열에 위치한 노드들끼리는 바로 페어와이즈 키를 생성할 수 있도록 한다. 동일한 행이나 열에 위치하지 않는 노드가 페어와이즈 키를 설정하는 경우에는 각 센서 노드의 ID를 통해 센서 노드의 위치를 파악할 수 있고, 센서 위치 정보를 이용하여 상대 센서 노드에 이르는 패치를 찾을 수 있다.

Ⅲ. 제안 모델

본 논문에서는 센서 네트워크를 BS, 클러스터 헤더(CH), 센서 노드로 구성되는 3 단계의 관리 구조를 가지며, 본 논문에서 제시한 키 관리 기법은 클러스터 내에서 CH가 센서 노드를 인증하기 위한 인증키 구성 시 CH가 제공하는 피벗값을 기준으로 인증키를 구성하도록 함으로써 불법적인 노드의 침입을 억제할 수 있다.

3.1 제안 모델의 특징

3.1.1 위치정보를 이용한 인증키 생성 방법

본 논문에서 제안한 키 관리 기법은 극좌표를 기반으로 하여 위치 정보(거리와 각도, 기준점)를 이용해 인증키를 생성하도록 함으로써, 기존 연구들에서 나타나는 초기 그룹 생성작업을 위해 필요한 키 전달 과정의 트래픽 오버헤드를 줄이고 "중앙 서버독립적인 키 관리" 구조를 갖는다.

센서 노드의 상대적인 위치값을 계측하는 방법은 저용량의 센서 노드에서 쉽게 구현할 수 있는 위치 검출 기법인 TDoA (Time Differnce of Arrival)[12] 와 AoA (Angle of Arrival)[13] 의 두 가지 방식을 사용한다. TToA는 거리 측정을 위해 사용되며 1 센치미터 이내의 오차율을 보이며, AoA는 1도 이내의 오차율을 갖고 있다[12][13].

클러스터 내에서 CH가 결정되면 CH는 자신의 상대 위치 좌표를 전송하고, 이를 수신한 센서 노드는 CH와의 상대적인 위치값(거리와 각도)을 이용하여 자신의 인증키를 생성한다.

3.1.2 CH의 기준각도 피벗

센서 노드들이 설치 될 때 각 센서 노드는 기본 피벗을 설정한다. 피벗은 극좌표에서 각도를 표시할 때 기준이 되는 값이다. 실제 센서 노드의 위치를 알 수 있다하여도 CH에서 센서 노드간의 거리는 알 수 있으나 피벗을 모르는 경우에는 각도 값을 유추하지 못하므로 위치 정보를 이용한 인증키 생성 기법에서 가장 치명적이 될 수 있는 위치 정보의 노출에 따른 보안 위험성을 보완할 수 있다.

그림 1은 데카르트 좌표에서는 CH에서 센서 노드까지의 상대위치가 같더라도 피벗에 따라 극좌표는 다른 값으로 좌표가 표시되는 것을 나타낸다.

그림 1에서 CH에 대한 노드 1과 노드 2의 상대 위치를 표시할 때 데카르트 좌표를 이용하면 노드 1과 노드 2는 모두 (x,y)이다. 그러나 극좌표의 피벗을 적용하면 노드 1의 좌표는 (d,θ)이고 노드 2의 좌표는 (d,θ')로 서로 다르게 적용된다.

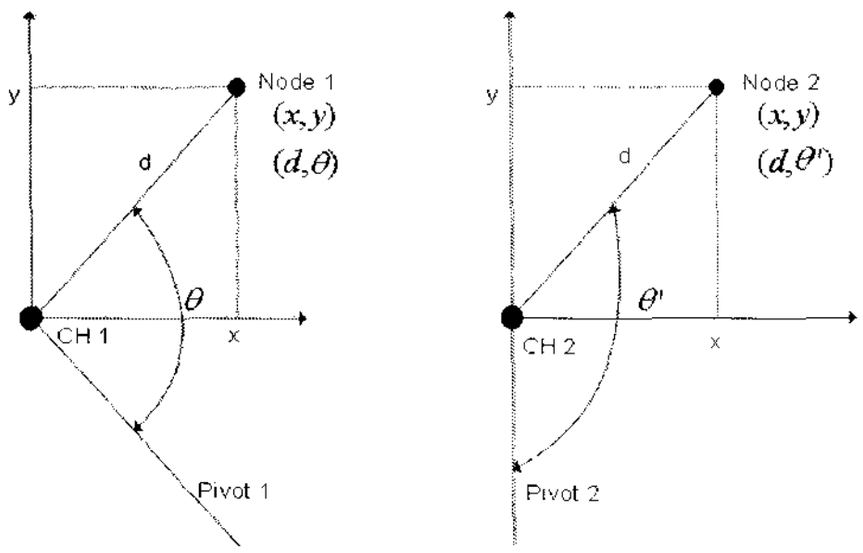

그림 1. 피벗 예제
Fig. 1 An example of pivot

3.1.3 극좌표 기반 구조

모든 센서 노드들은 클러스터 내에서 CH가 설정되면 CH가 전송하는 피벗을 수신하여 자신의 위치 검출 기능에 의해 자신의 위치값을 설정하여 인증키로 사용한다. 기존의 논문에서는 제 3의 공격자가 위치값과 이 알고리즘만 알게 되면 클러스터 내에서 공격자의 위치에 의한 가짜 ID 값을 설정할 수 있다.[11]

제안 모델에서는 CH와 센서 노드 간에 키 생성을 위하여 극 좌표계에서의 좌표값을 사용하기 때문에 인증키 생성을 위한 초기 인증키 배분에 인증 노드나 관리 노드의 개입이 없이 CH가 인증키 관리 노드의 역할을 수행할 수 있다.

극 좌표에 의한 피벗의 구성은 그림 2와 같다.

그림 2에서 노드 1과 노드 2를 일반적인 원형 좌표에 의한 위치값으로 표시하면 다음과 같다.

- CH로부터의 거리 : $d_{CH(C)\rightarrow Node(x)} = d_{Node(x)}$
- CH로부터의 각 : $\theta_{CH(C)\rightarrow Node(x)} = \theta_{Node(x)}$
- Pivot : θ_{Pivot}

이때, 노드 2는 원형 좌표의 의하여 x, y축을 기준으로 거리와 각도(θ_{xy}, $d_{CH,x}$)가 설정되지만, 노드 1은 피벗을 기준으로 거리와 각도(θ_P, $d_{CH,x}$)가 설정되기 때문에 공격자는 CH의 위치만 알게 되면 노드 2의 위치값을 쉽게 알게 되어 이를 이용하여 ID 값을 설정할 수 있다. 그러나 노드 1과 같이 CH의 피벗을 알지 못하면 CH로부터의 거리와 각도를 알 수 없어 공격자는 ID 값 설정할 수 없기 때문에 공격자의 침입을 막을 수 있다.

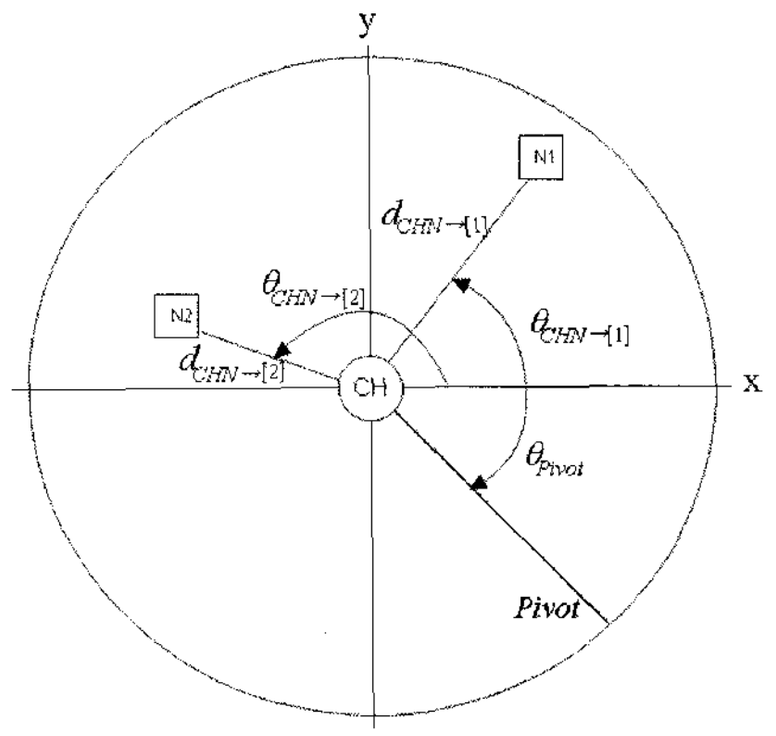

그림 2. 피벗에 의한 극좌표
Fig. 2 The polar coordinates of pivot

3.2 키 관리 구조

3.2.1 동적 키 관리 기법

데이터의 안전한 전달을 위해서는 인증과정과 암·복호화 과정이 필요하기 때문에 인증키와 비밀키가 필요하다. 이를 위해 센서 노드는 초기에 설치될 때 BS로부터 노드 번호와 전송 데이터의 암호화에 사용되는 비밀키(Ks)를 부여 받고, 센서 노드의 위치를 파악할 수 있는 위치 검출 기능을 갖도록 한다.

3.2.2 서버 독립적인 키 관리

본 논문에서 제안하는 클러스터 기반의 키 관리 구조는 "CH에서 키 관리"를 위해 별도의 인증 서버나 관리 서버를 참조하지 않고 서버 독립적으로 동작한다.

기존의 키 관리 기법들은 추가 노드를 인증하기 위해 관리 서버에 보관중인 키 정보를 확인하는 방법을 사용하고 있거나, 초기에 2차 다항식을 센서 노드에 기억시켜 공식에 의해 키를 생성하도록 하는 방법 등을 사용하지만 본 논문에서는 클러스터 내에서 CH가 클러스터 단위로 키를 생성하고 관리하는 구조를 갖는다.

3.3 세부 키 관리 절차

제안된 모델에서 세부적인 동작과정을 설명하기 위해 사용된 기호들을 정리하면 다음 표 1과 같다.

표 1. 수학적 표현 기호
Table. 1 expression symbols for mathematics

기호	설명
$\|\|$	$A\|\|B$ 이면 A와 B를 연접하는 기호
N_{CH}	클러스터 헤더(CH)
N_{BS}	베이스 스테이션(BS)
N_x	센서 노드 x
SDU_x	센서 노드 x가 전송하는 정보 (Service Data Unit)
$L_{CH,x}$	N_{CH}를 기준으로 N_x의 상대 위치
θ_P	N_{CH}를 기준으로 한 피벗
$\theta_{CH,x}$	N_{CH}를 원점으로 피벗 θ_P 와 N_x의 각도
$d_{CH,x}$	N_{CH}를 원점으로 N_x의 거리
$Session_{CH}$	CH의 세션 ID
$K_{CH.x}^{Ak}$	N_{CH}와 N_x가 공유하는 인증키
$K_{BS,x}^{Sk}$	N_{BS}와 N_x가 공유하는 비밀키
$K_{BS,CH}^{Sk}$	N_{BS}와 N_{CH}가 공유하는 비밀키
$E_{A128}(K,M)$	키K로 M을 AES128 알고리즘으로 암호화
$D_{A128}(K,M)$	키K로 M을 AES128 알고리즘으로 복호화
$L_{x,CH}=R(L_{CH,x})$	$L_{CH,x}$를 이용해서 $L_{x,CH}$를 구하는 함수
$N_{id}::z=F(x,y)$	N_{id}에서 함수 $F()$를 사용하여 z 값을 계산
$N_{id}::SDU=\{N_{id}\|\|x\}$	N_{id}에서 N_{id}와 x를 연접하여 SDU를 구성

3.3.1 센서 노드 설치

제안한 극좌표 기반 시스템의 전체적인 처리과정은 모든 센서 노드들에게 필요한 정보를 저장하는 초기화 등록 과정부터 시작된다. 초기화 단계는 처음 센서 노드들이 배치되기 전에 센서 노드들에 대한 각종 설정작업에 해당한다. 센서 노드들에게 저장되는 정보는 표 2와 같이 CH로 동작 가능한 최대 횟수, 노드 ID, 비밀키, 피벗이며, BS는 각각의 센서 노드에 할당된 정보를 표 3과 같이 비밀키 테이블에 저장한다.

표 2. 초기화 정보
Table. 2 The Initialization Information

종류	설명
$N_{id}:K_{BS,x}^{Sk}$	센서 노드의 ID 와 비밀키
θ_P	센서 노드들에게 저장되는 극좌표 θ의 피벗

표 3. 비밀키 테이블
Table. 3 Secret key Table

종류	크기	설명
id	8비트	센서 노드를 식별하기 위해 할당한 ID 번호
$K_{BS,id}^{Sk}$	128비트	센서 노드에게 할당한 비밀키

3.3.2 CH 선택

1. N_{CH} 선정

클러스터 내에서 전력 잔류량이 제일 큰 센서 노드를 CH로 설정하며, 전력 잔류량이 거의 동일한 초기 상태에서는 전력 잔류량에 의한 구분이 힘들기 때문에 미리 예측 설정한 평균 그룹원의 수를 이용하여

$$Node_{ID} \mod AvgGroupMember = 0$$

인 센서 노드가 CH가 되도록 설정한다.

2. N_{CH} 광고

CH로 선정된 노드는 ADV_MSG를 생성하여 클러스터 내의 모든 센서 노드들에게 브로드캐스트로 전송하며, 자신의 노드 ID(CH_{id})와 세션 ID($Session_{CH}$)를 연접하여 ADV_MSG를 생성한다.

세션 ID는 센서 노드에서 CH로 인증키를 암호화하여 전송하기 위해 사용되는 RC5 암호 알고리즘의 비밀키로 사용된다.

$$N_{CH}::ADV\_MSG=CH_{id}\|Session_{CH}$$
$$N_{CH}\rightarrow ※ : ADV\_MSG$$

3. 다른 노드들은 센서 노드로 결정되고 그룹에 참가하기 위한 인증 절차를 시작한다.

3.3.4 CH와 센서 노드간의 인증키 생성과 인증

1. 인증 요청

ADV_MSG 를 수신한 센서 노드(N_x)는 수신한 메시지의 세션에 참여하기 위해 자신의 노드 ID(x_{id})와 CH가 전송한 세션 ID($Session_{CH}$)를 연접하여 *REQ_Auth* 메시지를 생성하여 CH로 전송한다.

$$N_x :: REQ\_Auth = \{x_{id} \| Session_{CH}\}$$
$$N_x \rightarrow N_{CH} : REQ\_Auth$$

2. 인증 응답

*REQ_Auth*를 수신한 CH는 N_x가 인증키를 생성하기 위해 사용되는 CH의 피벗값($\theta_{CH,x}, d_{CH,x}$)을 RC5 암호 알고리즘을 이용하여 암호화하여 *REP_Auth* 메시지를 생성한 후 N_x에게 전송한다.

$$N_{CH} = F(\theta_{CH,x}, d_{CH,x}, x_{id})$$
$$REP\_Auth = E_{RC5}(N_{CH})$$
$$N_{CH} \rightarrow N_x : REP\_Auth$$

3. 인증 처리

*REP_Auth*를 수신한 N_x는 *REP_Auth*를 RC5 암호 알고리즘을 이용하여 복호화하여 자신의 ID(x_{id})를 확인하여 N_{CH}를 검증한 후 맞으면 CH가 제공한 피벗값과 세션 ID를 이용하여 자신의 인증키($K_{CH,x}^{Ak}$)를 생성하여 RC5 암호 알고리즘으로 암호화하여 ACK 신호와 함께 CH로 전송한다.

$$N_x = D_{RC5}(REP\_Auth)$$
$$N_x :: K_{CH,x}^{Ak} = F(\theta_{CH,x}, d_{CH,x}, x_{id})$$
$$REP\_Auth = E_{RC5}(K_{CH,x}^{Ak})$$
$$N_x \rightarrow N_{CH} : REP\_Auth$$
$$\begin{cases} N_x \rightarrow N_{CH} : ACK \\ \text{or} \\ N_x \rightarrow N_{CH} : NACK \\ N_{CH} \rightarrow N_x : Term \end{cases}$$

4. 인증 결과 성공

N_{CH}는 센서 노드로부터 ACK가 수신되면 성공적으로 N_x가 클러스터 내의 그룹에 합류한 것이므로 N_{CH}는 자신의 테이블에 센서 노드의 노드 ID(x_{id})와 인증키($K_{CH,x}^{Ak}$)를 저장하고 N_x로부터 데이터를 수신하도록 한다.

5. 인증 결과 실패

N_{CH}는 센서 노드로부터 $NACK$가 수신되면 N_{CH}는 N_x에게 $Term$ 메시지를 전송하고 종료하며, N_x는 처음부터 반복 수행한다.

3.3.5 센서 노드에서 CH로의 데이터 전송

1. 전송 요청

인증이 성공한 N_x는 자신의 ID를 포함한 *REQ_DATA* 메시지를 생성하여 전송한다.

$$N_x :: REQ\_DATA = \{x_{id}\}$$

2. 전송 응답

REQ_DATA 메시지를 수신한 N_{CH}는 자신의 ID (CH_{id})를 포함하는 *REP_DATA* 메시지를 생성하여 전송한다.

$$N_{CH} :: REP\_DATA = \{CH_{id}\}$$

3. 전송 준비

N_x는 수집된 데이터($DATA$)와 자신의 ID를 연접하고, 초기 설치 시 BS에서 할당받은 비밀키($K_{BS,x}^{Sk}$)를 이용하여 AES128 알고리즘으로 암호화하여 SDU를 생성한다.

$$N_x :: SDU = E_{A128}(x_{id} \| DATA)$$

4. 데이터 전송

N_x는 SDU와 자신의 ID를 연접하고, 인증키($K_{CH,x}^{Ak}$)를 포함하여 RC5 암호 알고리즘을 이용하여 암호화하여 $DATA_MSG$를 생성하여 N_{CH}에게 전송한다.

$$N_x :: DATA\_MSG = \{E_{RC5}(K_{CH,x}^{Ak}, x_{id} \| SDU)\}$$
$$N_x \rightarrow N_{CH} : \{x_{id} \| DATA\_MSG\}$$

5. 센서 노드의 인증 및 데이터 수신

N_{CH}는 수신된 $DATA_MSG$를 RC5 암호 알고리즘으로 복호화하여 N_x의 인증키($K_{CH,x}^{Ak}$)로 분해한 후 센서 노드의 ID를 추출하여 맞으면 인증하여 ACK를 전송하고, 틀리면 $NACK$와 $Term$ 메시지를 전송하고 전송을 종료한다.

$$N_{CH} :: \{x_{id} \| DATA\_MSG\} = D_{RC5}(K_{CH,x}^{Ak}, x_{id} \| SDU)$$

$$\begin{cases} N_{CH} \rightarrow N_x : ACK \\ \quad \text{or} \\ N_{CH} \rightarrow N_x : NACK \\ N_{CH} \rightarrow N_x : Term \end{cases}$$

3.3.6. BS의 데이터 수집

1. 전송 요청

N_{CH}는 자신 ID(CH_{id})를 포함하는 REQ_DATA 메시지를 생성하여 N_{BS}로 전송한다.

$$N_{CH} :: REQ\_DATA = \{CH_{id}\}$$

2. 전송 응답

데이터를 수신한 N_{BS}는 자신이 보관중인 N_{CH}의 ID를 확인하여 맞으면 N_{BS}의 ID를 포함하는 REP_DATA 메시지를 생성하여 전달한다.

$$N_{BS} :: REP\_DATA = \{BS_{id}\}$$

3. 데이터 전송

정당한 BS임을 인증한 CH는 센서 노드로부터 수신된 데이터($DATA$)를 BS로 전송하기 위해 $DATA_MSG$를 생성하여 비밀키($K_{BS,x}^{Sk}$)를 이용하여 AES128 알고리즘으로 암호화하여 전송하며, 데이터가 모두 전달될 때까지 반복한다.

$$N_{CH} :: DATA\_MSG = \{E_{A128}(K_{BS,CH}^{Sk}, DATA)\}$$

$$N_{CH} \rightarrow N_{BS} : \{N_{CH} \| DATA\_MSG\}$$

4. 전송 종료

모든 데이터가 전송되면 CH는 BS에게 $Term$ 메시지를 전송한다.

$$N_{CH} \rightarrow N_{BS} : Term$$

Ⅳ. 시스템 분석

본 논문에서 제안한 모델에 대한 성능 평가 결과는 다음과 같다.

4.1 시스템 분석 및 성능평가

본 절에서는 제안된 시스템에 대하여 TinyOS 환경에서 시뮬레이션을 통하여 제안한 시스템의 성능을 분석하였으며, 실험을 위한 대상 모델은 MICA2 노드를 대상으로 하였다. 앞장에서 제안/설계된 내용을 NesC 코드로 작성하여 키 관리 노드인 CH와 센서 노드들을 생성한 후 CH와 노드간의 키 생성 및 분배와 인증에 관련하여 정상적인 동작을 이루는지 여부를 확인하였다.

4.1.1 실험 내용 및 결과

본 연구에서는 시뮬레이션을 통하여 페어와이즈 기법과 본 논문에서 제안한 기법을 비교 평가한 결과를 나타내었다.

1. 키 생성 개수

페어와이즈 기법과 본 논문에서 제안한 기법에서의 키 생성 개수의 비교는 그림 3과 같다.

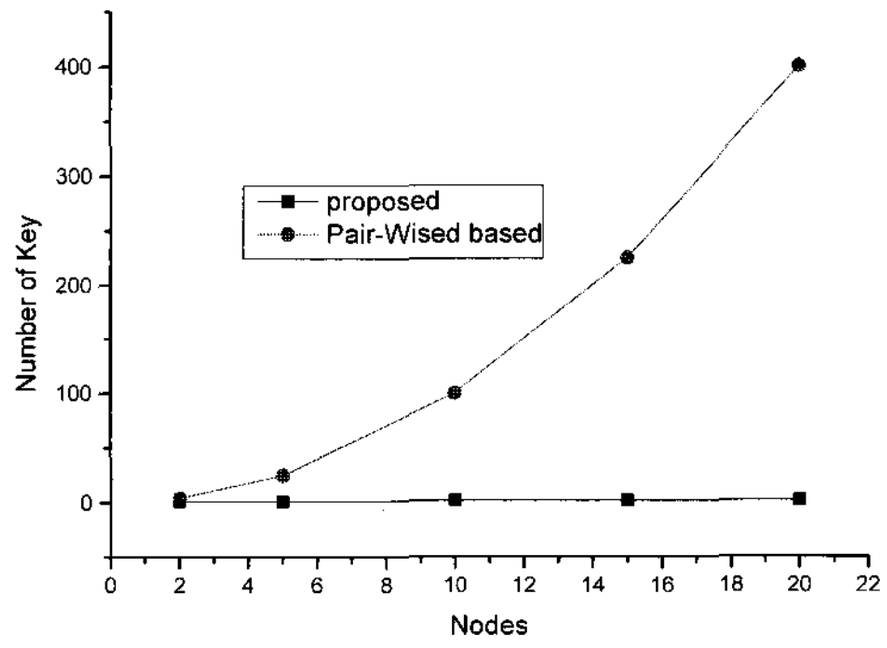

그림 3. 키의 개수
Fig. 3 The number of key

그림 3에서 센서 노드의 수가 20개 이상이 되는 경우 페어와이즈 기법의 경우 키 생성 수가 급격히 증가함을 알 수 있다. 그러나 본 논문에서 제안한 기법은 클러스터 내에서는 그룹키가 없고, 단지 센서 노드와 CH 간 상호 인증을 위한 인증키만 존재하게 되며, 인증키는 CH 노드의 수에 비례하게 된다.

또, 제안한 기법은 표 4와 같이 노드 갯수 만큼의 키만 저장하면 되므로 상대적으로 메모리나 CPU의 성능이 제약적인 센서 네트워크에 보다 더 적합함을 알 수 있다.

표 4. 키 생성 비교표
Table. 4 Key create comperative table

구분	저장할 키의 갯수	키 생성 방법
클러스터 기법	$n^*(n-1)$개 필요 그룹내 모든 노드에 대해 쌍으로 키를 관리	$r = p^*q$ $\Phi = (p-1)^*(q-1)$ $E^* D \bmod \Phi = 1$ 곱셈과 뺄셈, mod 만으로 구성된다
그리드 기법	열의 수 * 행의 수 그리드의 격자의 갯수만큼 키를 생성하여 저장한다.	난수 발생 알고리즘
제안한 방법	n개 BS1개 + 클러스터내의 센서 노드 갯수	난수발생 알고리즘 또는 사용자정의 함수 사용

2. 데이터 처리 소요 시간 측정

그림 4는 1KB의 데이터를 처리할 때 소요되는 시간을 측정한 결과이다.

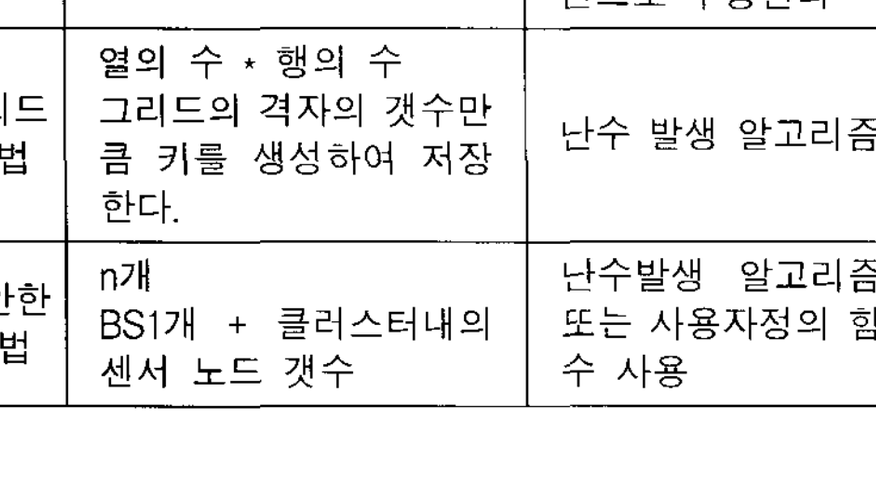

그림 4. 데이터 처리속도
Fig. 4 Data processing speed

그림 4에서 페어와이즈 기법은 초기 지연시간은 적지만 센서 노드가 증가함에 따라 점차로 느려지고 있으나, 본 논문에서 제안한 기법은 데이터 처리 시 인증키에 의한 인증과 센서 노드에서 데이터 전송 시 AES128 알고리즘을 이용한 암호화만 수행되기 때문에 처리 속도의 증가를 보이고 있다.

3. 다수개의 CH가 존재할 때 전송 시간

센서 노드들을 임의로 배치하고 다수개의 클러스터를 구성하여, 각각의 클러스터 내의 센서 노드가 데이터를 전송할 경우의 전송 시간을 측정하였으며, 측정 결과는 그림 5와 같다.

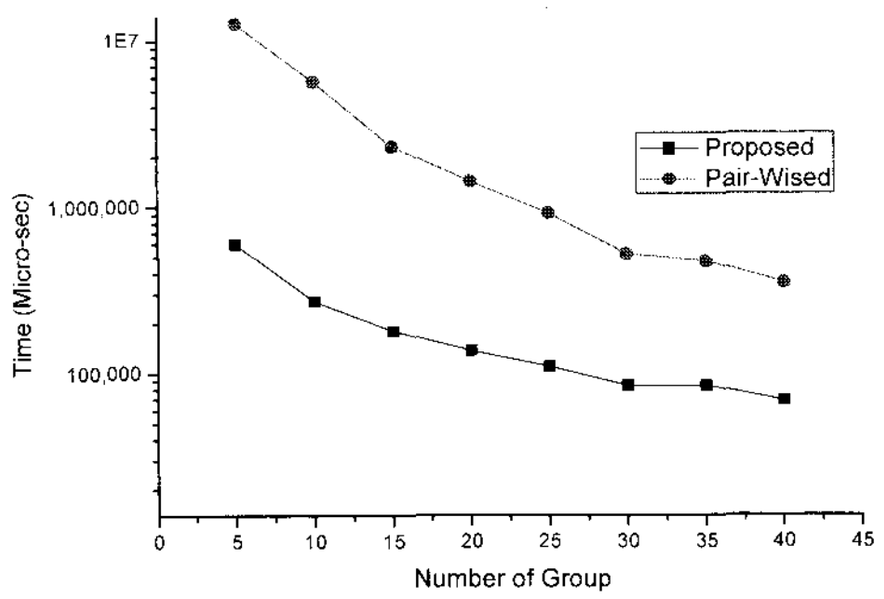

그림 5. 그룹 개수 별 전송시간
Fig. 5 A group number per transmission time

페어와이즈 기법과 본 제안 기법 모두 클러스터의 수(그룹 수)에 따른 전송 시간이 감소하고 있으나, 본 논문에서 제안한 기법은 CH는 센서 노드와의 사이에 상호 인증만을 수행하고, 센서 노드로부터 수신된 데이터는 곧바로 BS로 전송되기 때문에 페어와이즈 기법에 비하여 전송 시간이 빠름을 알 수 있다.

V. 결 론

본 논문에서는 클러스터 내에서 센서 노드가 안전하게 정보를 전송할 수 있으며 불법적인 센서 노드의 침입을 억제할 수 있는 극좌표를 이용한 클러스터 기반의 센서 네트워크의 키 관리 기법을 제안하였다. 센서 네트워크 환경에서 안전하고 효율적으로 센서 노드 관리를 위하여 일반적인 모델로 피벗값을 기반으로 하는 상대 좌표계를 이용한 방법을 제안하였고, 상대 좌표계를 이용한 방법 중에 극 좌표계를 이용한 방법을 채택하여 모델링하고 시뮬레이션을 수행하였다.

본 논문에서 제안한 극 좌표계를 이용한 방법은 클러스터 단위로 센서 노드들이 운영되는 키 관리 기법에서 상대위치를 표시하기 위한 매개변수가 2개이면서도 상대 위치 정보를 취득하는데 있어서 평면 데카르트 좌표계에 비해서도 더 간편한 구조적 특성을 갖는다.

논문의 성능 분석을 위해 생성/관리되는 키의 개수, 그룹 관리를 위한 CH의 인증키의 인증 시간, 데이터 전송 시간을 비교할 때 기존 구조적 키 관리 기법들에 비해 우수한 성능을 보였다. 또한 키 생성 구조가 효율적이며

센서 노드의 인증키를 생성 할 때에 상대 위치 정보를 이용함으로써 클러스터 그룹을 위한 초기 키 설정 과정이 없이도 운영될 수 있음을 보였으며, 초기 키 문의/전달을 위한 지연시간에 의한 성능 개선을 보였다. 향후 연구과제로서 센서 네트워크에서 보다 완벽한 보안을 위해서는 CH에 의해 보안 위협이 발생하기 전에 다시 피벗을 주기적으로 바꾸어야 하는 방법이 고려될 수 있으며, 이 경우 기존 클러스터에 포함된 센서 노드들이 CH에 의해 피벗이 재설정되기 때문에 문제가 없으나 새로 추가되는 노드들에 대한 클러스터 내의 참여 문제를 해결할 수 있는 방법에 대한 추가적인 연구가 필요할 것이다.

참고문헌

[1] J.N.Al-Karaki and A.E.Kamal, "Routing techniques in wireless sensor networks: A survey", *IEEE Personal Communications*, 11(6), pp. 6~28, Dec. 2004.

[2] Y.Hu and A.Perrig, "A survey of secure wireless ad hoc routing", *IEEE Security & Privacy, May/June,* pp. 28~39, 2004.

[3] Chris Karlof Naveen Sastry "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks", *SenSys*'04, Nov. 2004.

[4] W.Du, J.Deng, Y.Han and P.K.Varshney, "A pair-wise key pre-distribution scheme for wireless sensor networks", *ACM CCS 2003*, pp. 27~30, October 2003.

[5] R. Merkle. "Protocols for public key cryptosystems. *In Proceedings of the IEEE Symposium on Research in Security and Privacy*, pp 122~134, Apr. 1980.

[6] 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*. 2004.

[7] 나재훈, 채기준, 정교일, "센서 네트워크 보안 연구 동향", *전자통신동향분석*, 제20권, 제1호, pp. 112~122, 2005년 2월

[8] J. Deng, R. Han, and S. Mishra, "Security Support for In-Network Processing in Wireless Sensor Networks," *Proc. of the 1st ACM Workshop on the Security of Ad Hoc and Sensor Networks(SASN),* 2003.

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

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

[11] D.Liu and P.Ning, "Establishing pair-wise keys in distributed sensor networks", *ACM CCS 2003,* Oct. 2003.

[12] H. Balakrishnan, R. Baliga, D. Curtis, M. Goraczko, A. Miu, N. Priyantha, A. Smith, K. Steele, S. Steller and K. Wang, "Lesson From developing and deploying the cricket indoor location system", *preprint,* Nov. 2003.

[13] N. Priyantha, A. Miu, H. Balakrishnan and A. Steller, "The Cricket compass for context-aware mobile application.", *In Proceedings of the 7th Annual ACM/IEEE international Conference on Mobile Computing and Networking (MobiCom),* pp. 1~14, Jul. 2001.

유 황 빈(Hwang-bin Ryou)

1989년 경희대학교 전자공학과 공학박사
1981년 3월~현재 광운대학교 컴퓨터소프트웨어학과 교수
2002년 9월~2003년 6월 한국정보보호진흥원 초빙교수
※관심분야 : 정보보안,

홍 성 식(Seong-Sik Hong)

광운대학교 컴퓨터과학과 이학박사
1994년 9월~현재 혜전대학 컴퓨터과 교수
※관심분야 : 컴퓨터 네트워크 , 보안