원형 좌표계를 이용한 센서네트워크 키 관리 기법

홍성식* · 유황빈**

요 약

센서네트워크는 매우 제한된 자원을 갖는 작은 노드들로 구성되어 온도, 진동 등의 지역적 정보를 수집하고 통신망을 통하여 이들 정보를 제공한다. 센서네트워크는 일반적으로 매우 작고 제한적인 자원을 갖는 노드들로 구성되므로 이러한 노드들에 보안 서비스를 추가하는 것은 상당히 어려운 문제이다. 센서네트워크의 안정성을 위해서는 일반 네트워크 노드들에 비하여 보다 더 효율적인 키 관리 기법이 필요하다.

본 논문에서는 원형 좌표계를 이용한 위치 정보를 기반으로 키 관리하는 기법을 제시한다. 원형좌표계의 원점을 중심으로 노드와 노드간의 상대적인 위치 정보를 이용하여 키를 생성하였다. 이러한 방법은 대칭적 위치 구조에 의해 키 생성이 다른 방법에 비하여 간결하면서도 효율적임을 보였다.

Key Management Scheme of Sensor Network using Circular Coordinates

Seong-Sik Hong* · Hwangbin Ryou**

ABSTRACT

Sensor network is made from very small and restrictive-power nodes, and they collect some information of environment like as thermal and tremor, etc. And they transfer the information to each other. Generally, supporting the Security service of sensor network is a difficult work, because the nodes have very small cpu-power and low electronic-power. So, More effective Key management scheme will needed for the maintenance of stability.

In this paper, we propose the location based key management scheme with circular coordinates. We were make the key with the relative location information from one node to other. The new scheme show more simple and effective result then the other method for key management.

Key words : Sensor Network Security, Group Key Management, Polar Coordinates

* 혜전대학

** 광운대학교 컴퓨터소프트웨어학과

1. 서 론

최근 무선 센서네트워크는 매우 다양한 분야에서 활용되고 있다. 센서네트워크는 매우 제한된 자원을 갖는 작은 노드로 구성되어 지역적 정보를 수집하고 통신망을 통하여 정보를 전달한다. 그러므로 이런 자원이 제한적인 센서 네트워크를 운용하기 위해서는 기존의 암호화 방법이나 키 관리 기법들이 부적절하다. 기존 키 관리 기법들은 연산의 복잡성, 대량의 통신량 등의 문제점을 가지고 있다. 그러므로 이런 제약조건을 만족할 수 있는 키 관리 기법이 필요하다[1, 2].

전력 자원이 극히 제한된 센서네트워크에서는 잦은 데이터 통신을 통한 보안 유지가 어렵다. 그러므로 보안 기능을 제공하기 위한 트래픽 양을 줄이는 것도 센서네트워크 노드들의 생존성을 높이는데 큰 역할을 한다. 그러나 통신 트래픽 양을 줄이더라도 노드의 보안 서비스는 만족할 만한 수준을 유지해야만 한다. 네트워크 보안에서 가장 중요한 부분은 키 관리 기법이다. 타당한 키가 설정되어 있다면 다양한 보안 기술 적용에 의해 보호 받을 수 있다. 그러나 센서 네트워크는 각각의 노드들이 갖고 있는 자원의 제한성에 의해 일반 네트워크 구조에서 적용되는 키 관리 기법을 사용하는 데는 상당한 무리가 따른다. 신뢰할 수 있는 인증 서버에 의해 키를 분배 받는 방법은 구조적인 기반 구조가 없는 센서네트워크에서는 구현되기 어렵다. 또한 기존 암호화 알고리즘으로 많이 사용되는 RSA와 같은 암호화 기법은 상대적으로 낮은 처리 속도를 갖는 센서 노드에서 처리하기에는 너무 높은 연산 능력을 요구한다[3].

가장 간단한 방법으로는 모든 센서 노드들이 동일한 키를 공유하게 하는 방식이 있다. 그러나 이런 그룹 키가 외부에 노출될 경우 모든 노드들이 노출되는 위험성이 존재한다. 또한 노드 쌍마다 각각의 독립적인 키를 소유하는 방법은 위와 같은 위험성을 최소화하지만 각각의 센서 노드들은 총 개의 키를 저장하고 관리하여야 하므로 센서네트워크에는 부적합하다[1, 2].

본 논문에서는 초기 키 생성을 위한 통신 절차를 최소화하고, 키 관리를 위한 통신 절차를 줄여 센서네트워크의 생존성을 높이는 방법과 만족할 만한(Acceptable) 보안성을 제시한다. 서론에 이어 2장에서는 관련연구에 대하여 기술하고 3장에서 제안하는 기법을 설명한다. 4장에서는 기존 연구와 제안하는 방법을 비교분석하고 5장에서 결론을 기술한다.

2. 기존연구

2.1 Security Protocols in for Sensor Networks

센서 네트워크 프로토콜을 정의하고 있는 SPINS [1]방법은 BS(Base Station)가 항시 노드들 간의 데이터 통신에 관여하여 게이트웨이의 역할을 수행하도록 구성되어 있다. 그러므로 센서 노드들은 BS를 통하여 자신의 키를 전달하는 구조를 가지므로 구조적으로는 간단하지만 센서 노드들이 항시 BS와 통신 상태를 유지하여야 하므로 BS와 인접한 노드들을 과도한 자원소모를 유발하게 된다. 그러므로 SPINS는 대규모 센서네트워크에는 적합하지 않다.

2.2 Key Infection[2]

공격자가 모든 네트워크 영역을 도청하기 어렵다는 특징에 착안하여 보안 중요도가 그다지 높지 않은 분야에서 사용되는 기법이다. 암호화 키를 평문 상태로 모든 노드들에게 전송한다.

센서 노드 초기화 때에는 랜덤하게 생성된 세션키를 그대로 공개하지만 공격자가 도청할 수

있는 범위는 한정되어 있으므로 키를 평문 상태로 공개해도 대부분의 키들은 안전하다.

그러나 다수의 공격자가 협력하여 공격을 시도할 수 있으므로 보다 복잡한 보호기법이 요구된다.

2.3 TREE-BASED GROUP DIFFIE-HELLMAN[6]

그룹 멤버의 변경이 발생할 경우 스폰서로 지정되는 특정 멤버가 그룹 키 생성과 분배 역할을 담당한다. 키 트리의 최상위 노드의 값이 그룹 멤버가 사용하는 그룹 키로 사용된다. 그러나 임의로 추가, 제거가 발생하는 센서 네트워크의 경우 키 트리의 구조가 한쪽으로 편중되는 사향 트리의 경우 트리의 레벨이 깊어지므로 과도한 트래픽과 연산이 발생할 수 있다. 그러므로 트리의 깊이를 실시간으로 재조정 하여 낮추어야 하는 문제가 있다.

2.4 Aggregator 기반의 그룹 키 관리 기법[4]

BS와 클러스터 단위를 중심으로 중간에 aggregator를 두는 기본 구조를 갖는 그룹 키 관리 기법이다.

각각의 센서 노드들은 사전에 BS와 1:1로 비밀 키를 가지고 있다는 가정 하에 그룹 키 분배를 수행한다. 이때 각각의 센서 노드들은 다음과 같은 형태의 유일한 Key를 가지고 있다.

$$K_{si} = F(K_m, S_i)$$

S_i는 센서 노드 i의 ID정보이며, K_m마스터키이다. 또한 각각의 aggregator는 $K_{A_j} = F(K_m, A_j)$ 같은 형태의 유일한 Key를 소유한다.

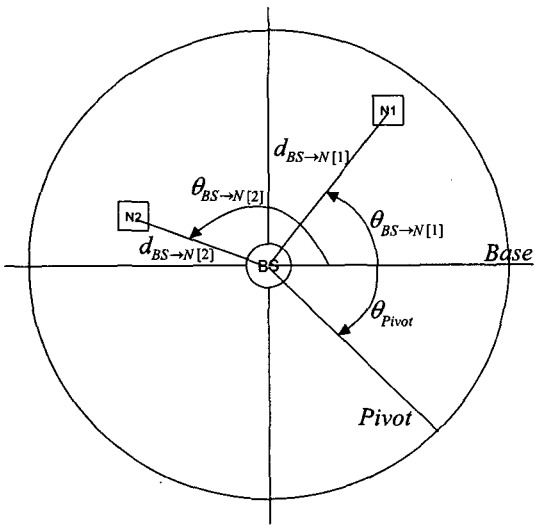
그러나 이 방법은 그룹 선언단계에서 BS는 모든 센서 노드 정보와 aggregator정보를 모든 센서노드에게 브로드캐스트 하여야 하므로 BS 및 인근 노드들에 과도한 트래픽이 발생할 수 있으며, 서브키 생성 단계에서는 BS가 각각의 센서 그룹의 aggregator에게 해당 센서 그룹에 포함되는 모든 노드의 키 값과 서브키 값을 전달해야 하는 부담을 갖고 있다. 그러므로 상당히 우월한 성능의 BS가 반드시 존재하여야 한다.

2.5 LEAP[5, 7]

통신 범위 이내에 최소한의 노드만 배치되는 것을 가정하고 있다. 초기 설정 전에 모든 노드는 BS로부터 동일한 초기키를 할당 받는다. 이후 각각의 노드들은 자신의 ID를 Broadcast 하고 이 값을 전송 받는 노드는 자신의 ID와 초기키로 인증키를 만들고 인증키로 MAC을 생성한 후에 MAC 값과 자신의 ID로 응답한다. 응답 받은 노드는 전송받은 ID와 초기키로 상대의 MAC을 확인한다. MAC이 확인된 노드에 대하여 자신의 ID, 상대 노드의 ID, 초기 키 값을 사용하여 Pair-wise 키를 생성한다[7]. 그러므로 LEAP 기법은 초기에 BS로부터 초기키를 할당 받은 노드만이 MAC을 생성하고 인증할 수 있으므로 사전 허가되지 않는 노드들에 대해 접근을 막을 수 있으나 초기 설정 시에 Pair-wise 키를 생성하는 초기 키 값이 모든 노드들에게 공개되는 문제점이 있으며 새로운 노드의 추가가 어렵다.

3. 제안시스템

본 장에서는 안전한 노드 인증을 위해서 그룹 키를 생성하도록 한다. 이때 그룹키 생성을 위하여 원형 좌표계 개념을 도입하고자 한다. 이에 제안하는 원형 좌표계를 (그림 1)에 나타내고 있다.



(그림 1) 원형좌표계

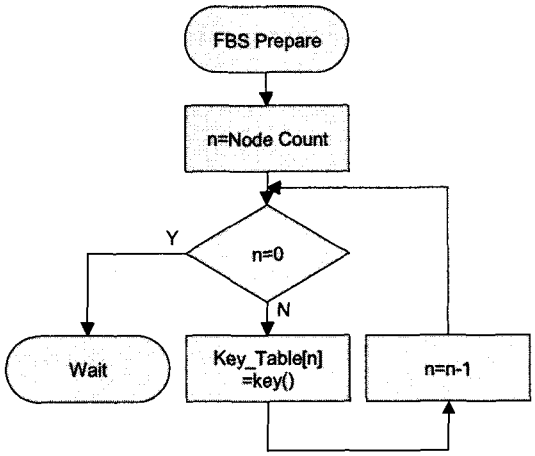
3.1 제안 시스템의 동작 절차

3.1.1 FBS(First BS)의 노드 등록 및 비밀키 전달과정

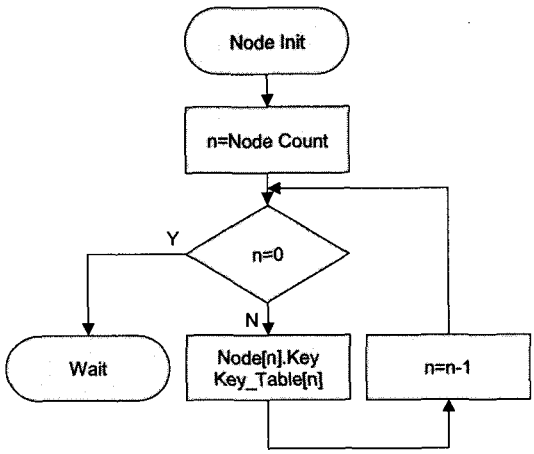
제안한 프로토콜은 통신을 수행하기 전에 다음과 같이 준비단계를 거치게 된다. 센서 노드들이 수집한 정보를 안전하게 최종 Sink 노드에게 전달할 수 있도록 암호화 할 수 있는 방안을 제공하도록 하였다. 이를 위해서 FBS는 자신의 비밀키를 센서노드들의 등록 과정에서 제공하고 해당 노드에 대한 식별 정보를 데이터베이스에 저장해 둔다. 그리고 해당하는 센서 노드의 식별 정보와 할당한 비밀키를 데이터베이스에 1:1로 사상하여 저장하여 둔다. 이의 처리과정을 (그림 2)에서 나타내고 있다.

3.1.2 센서 노드의 FBS 비밀키 등록 과정

(그림 3)에서는 FBS에 등록된 센서 노드들이 자신에게 할당된 FBS의 비밀키를 내부 저장 장치에 저장하는 과정을 흐름도로 나타내고 있다.



(그림 2) 최상위 관리 노드의 준비



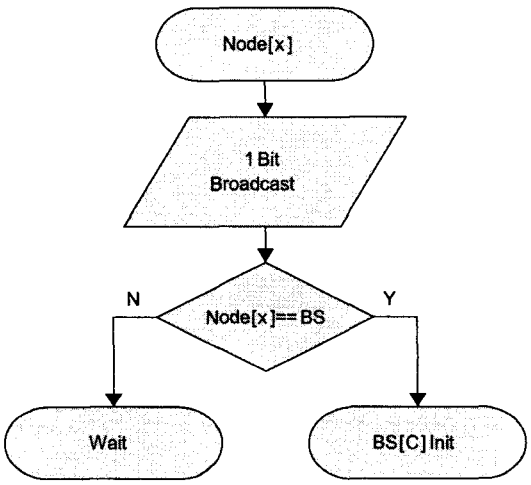
(그림 3) 노드에 키 전송

지금까지 센서 노드와 최종적으로 데이터를 전달받게 되는 FBS 노드의 사전 통신 준비 작업의 처리 절차에 대해서 살펴보았다.

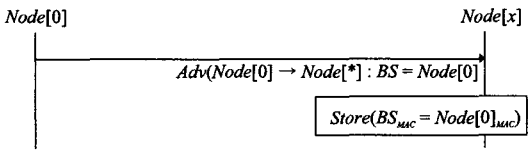
3.1.3 BS 설정 과정

노드들 중에서 잔류 전력량이 높고 처리 능력이 뛰어난 노드가 자신이 BS로 동작할 것을 주변의 노드들에게 광고를 한다. 이를 통해 BS가 설정된다. 이를 위해서 모든 노드들은 주변의 노

드에게 사전에 약속된 1bit 정보를 전송하고 이를 수신하여 자신의 잔류 전력량과 자료 송·수신 능력을 고려해 BS로 동작할 수 있는지 여부를 판별하게 된다.



(그림 4) BS 설정을 위한 통신 시도

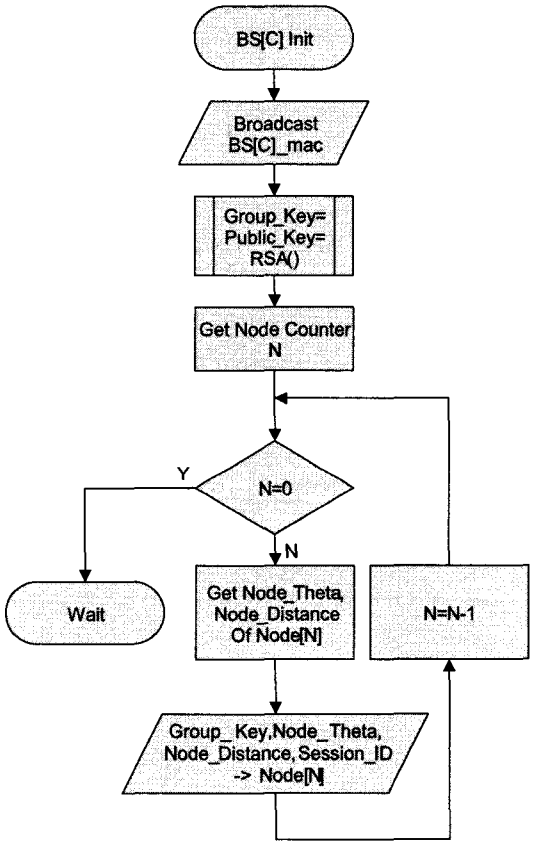


(그림 5) BS 설정 과정

(그림 4)와 (그림 5)에서 BS 설정을 위한 동작 절차와 내부적인 처리 과정을 나타내고 있다.

3.1.4 노드 그룹키 생성 및 분배 과정

그룹의 대표인 BS를 설정한 후 BS는 자신의 그룹에 속한 노드들을 등록한다. 다음으로 자신의 노드에 등록된 노드들에게 그룹 인증을 위해서 그룹키를 생성하여 전달한다. 다음으로 그룹키를 전달받은 노드들은 이를 활용하여 노드 개인의 인증을 위한 인증값을 생성한다. 이러한 과정을 통해서 노드와 BS 사이에서는 그룹의 인증과 노드에 대한 개별 인증이 가능해진다. (그림 6)에 나타내고 있다.



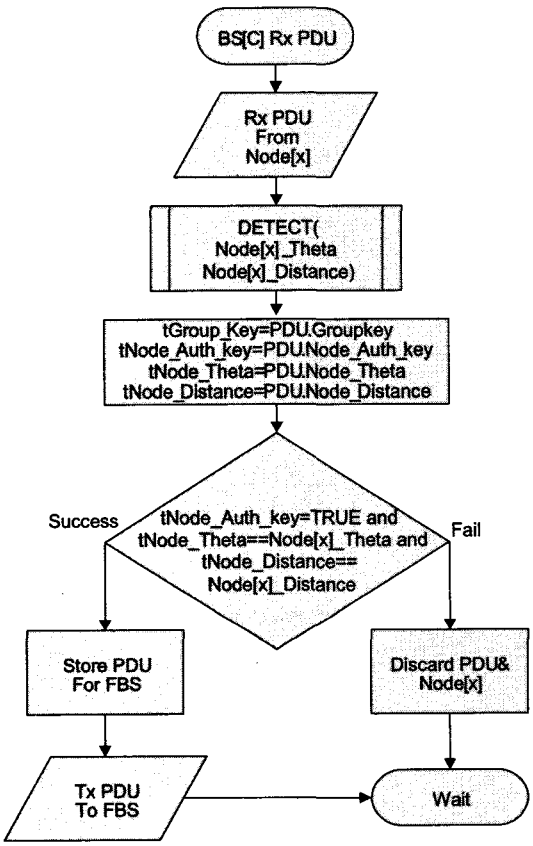
(그림 6) 그룹키 생성 및 전달

3.1.5 노드 인증 과정

BS로부터 그룹키를 전송받은 노드는 이를 토대로 자신의 인증값을 생성한다. 그리고 BS를 통해서 FBS에 데이터를 전송하고자 할 때 자신에 대한 인증을 요청하게 된다.

노드 인증값을 전달받은 BS는 노드의 위치 정보를 재차 조사하여 전달받은 노드에 대한 인증값을 스스로 생성해 본다. 생성된 노드에 관한 인증 정보와 노드로부터 전달받은 인증 정보를 확인하여 노드에 대한 인증여부를 결정하게 된

다. 이에 관해 (그림 7)에서 나타내고 있다.

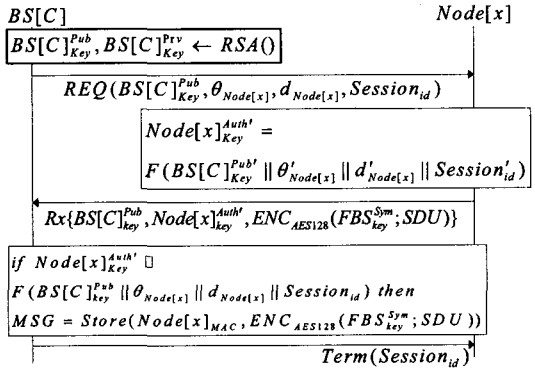


(그림 7) 노드에서 BS로 데이터 전송

3.1.7 데이터 암호화 및 전송

노드의 요청에 따라서 BS가 노드에 대한 인증 여부가 결정되어지면 인증성이 확인된 노드들은 사전에 등록과정에서 FBS로부터 할당받아 내부 메모리에 저장된 FBS의 비밀키를 사용하여 전달하고자 하는 데이터를 암호화한다.

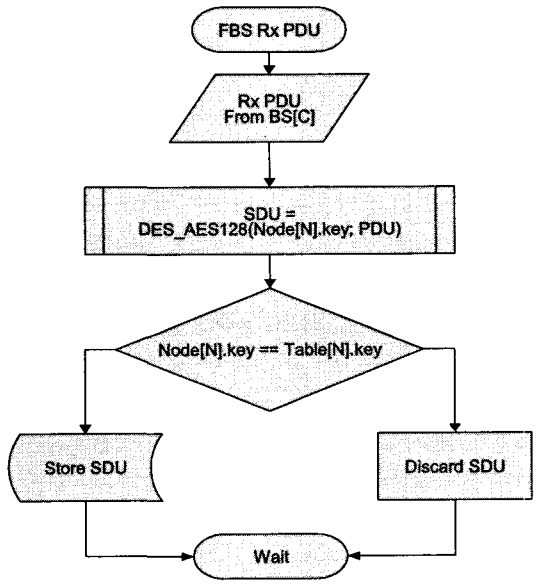
이때 암호화를 위해서 AES128 암호 알고리즘을 활용한다. 암호화를 통해서 전송 도중에 데이터에 대한 도청, 갈취에 대해서 비밀성을 보장할 수 있게 된다. 이에 관해 (그림 8)에서 세심하게 나타내고 있다.



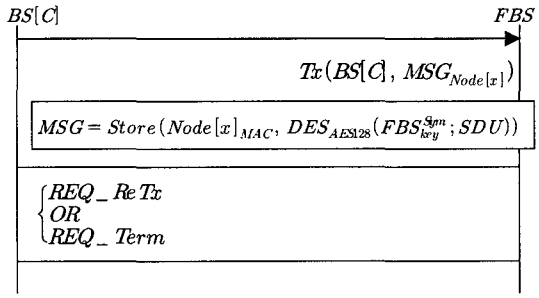
(그림 8) 인증 및 암호화된 데이터 전송

3.1.8 데이터 복호화 및 전송 종료

FBS는 노드로부터 BS를 거쳐 인증을 확인하고 인증된 노드로부터 암호화된 데이터를 안전하게 전달받을 수 있다. 전달받은 데이터를 읽기 위해서 FBS는 암호화된 데이터를 복호화하게 된다. 이를 위해서 사전에 노드에게 할당한 비밀키로 AES128 암호 알고리즘을 사용하여 복호화할 수 있다. 이에 대해서 (그림 9)와 (그림 10)에



(그림 9) 데이터 복호화 및 전송종료

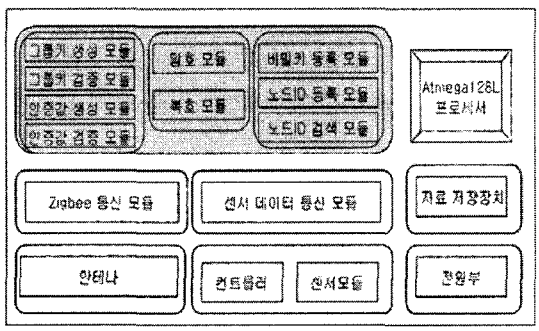


(그림 10) 데이터 전송 후 종료과정

서 나타내고 있다. 또한 데이터 전달을 마친 FBS는 데이터 통신의 종료 요청을 전달한다.

3.2 제안 시스템 설계 및 구현

제안한 시스템에 대한 내부의 논리적은 구조에 대해서 (그림 11)에서 나타내고 있다. 기존의 센서 네트워크의 프레임워크를 유지하면서 상위의 응용계층에 요구되는 시스템을 설계하였다.

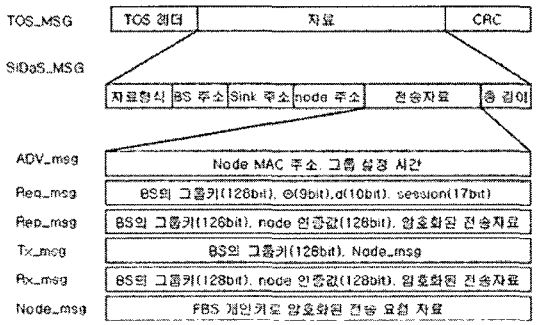


(그림 11) 제안 시스템 전체 구조

제안한 시스템은 상위 3개의 구성요소가 내부적으로 필요한 모듈들을 포함하는 구조로 설계 되었다. 또한 내부적인 처리 모듈들을 기존의 센서네트워크 상위 프레임워크에 설계함으로써 구조의 변경 없이 유기적으로 동작하도록 하였다.

3.3 내부 처리 데이터 포맷

제안한 시스템에서 노드와 BS, 그리고 FBS 간에 처리되는 데이터들의 포맷을 (그림 12)와 같이 나타낼 수 있다.



(그림 12) 데이터 포맷

기존의 TinyOS에서 사용하는 데이터 포맷을 유지하면서 제안하는 시스템에서 사용할 데이터 포맷을 확장하는 구조로 설계하였다. 각각의 노드와 BS, 그리고 FBS에서 요구되는 데이터들에 대해서 메시지 이름을 설정하고 나타내고 있다.

지금까지 제안한 시스템의 구조와 내부적으로 처리되는 데이터의 구조 및 시스템의 동작절차에 대해서 살펴보았다.

물론 제안하는 시스템에 대한 간략한 소개이다. 하지만 핵심적으로 제안하는 시스템에서 요구되는 연구의 목표를 명확하게 기술하였다. 또한 이를 위해서 처리해야 할 과정들도 명시하였다. 이제 제안한 시스템에 대한 성능을 평가하기 위해서 인증과 비밀성 보장을 위한 실험을 평가할 것이다.

4. 실 험

제안한 시스템에 대한 성능 분석을 위해서 두 가지 실험을 수행하였다. 먼저 제안한 시스템에

서 노드에 대한 인증을 위해서 필요한 그룹키 생성에 요구되는 처리시간을 측정하였다. 이를 위해서 TinyOS 운영체제에서 NesC 프로그래밍 언어를 사용해 제안하는 시스템에서 요구되는 응용 프로그램을 개발하였다. 또한, 시뮬레이션을 위해서 센서 네트워크를 데스크 탑 컴퓨터에서 시뮬레이션 할 수 있도록 도와주는 TosSim 시뮬레이터를 사용하였다.

4.1 실험환경

제안한 시스템에 대한 실험을 위한 시뮬레이션 환경은 <표 1>에서 나타내고 있다.

〈표 1〉 실험환경

구분	내용	비고
운영체제	TinyOS 2.0.4 Boomerang ver. (http://www.tinyos.net/)	
개발언어	NesC 1.2 (TinyOS 내부에 포함)	
시뮬레이터	TosSim (TinyOS Simulator)	
기타	자바 환경에서 모트 동작 모습 표현을 위한 TinyViz를 부분 적용	
하드웨어 사양	Pentium 4 3.0 Ghz 데스크 탑	

추가적으로 향후 실험에 대한 결과를 비주얼하게 나타낼 수 있도록 하기 위해서 TinyViz를 활용할 계획이다.

위에서 언급한 시뮬레이션 환경은 일반적으로 센서 네트워크에 대한 실험을 위해서 자주 모델링이 되는 MICA2 센서 노드를 모델링하도록 설정하였다. 이를 통해서 범용적인 실험 대상에 대한 시뮬레이션을 수행할 수 있다.

4.2 실험 방법 및 결과

제안한 시스템에 대한 노드 인증을 위한 그룹키 생성에 대한 처리 시간을 측정하여 나타냄으로써 시뮬레이션 결과를 얻었다. 실험 결과에 대한 공정성을 얻기 위해서 1KB 크기의 그룹키 정보를 일정 횟수 이상을 반복적으로 실험한 처리 시간을 획득하도록 하였다.

이에 대한 평균치를 획득한 결과 그룹키로 사용할 공개키와 대칭키에 대한 생성 시간은 각각 590 micro-sec과 1560 micro-sec이 소요됨을 측정하였다. 이에 관해 <표 2>로 나타낸다.

〈표 2〉 그룹키 생성시간 측정결과

실험	처리시간	비고
1회	590 ‖ 1560	
2회	600 ‖ 1430	
3회	572 ‖ 1557	
4회	589 ‖ 1556	

다음으로 노드에서 암호화 작업에 소요되는 시간을 측정한 결과를 나타내었다. 이는 위의 실험과 마찬가지로 노드에서 FBS로 전달하는 데이터에 대해 1KB 크기의 데이터를 암호화하는데 소요되는 처리시간을 Mbit per sec 단위로 측정해 보았다. 이를 <표 3>에서 나타내고 있다.

지금까지 제안한 시스템에 대한 성능 평가를 위한 실험 결과들에 대해서 살펴보았다. 주어진 조건 하에서 각각의 실험들에 대해 다음 절에서 비교 분석하고자 한다.

〈표 4〉 암호화 처리속도 측정

실험	처리속도	비고
1	0.116	
2	0.109	
3	0.115	
4	0.102	

5. 실험 결과 비교 및 분석

제안한 시스템에 대한 실험 결과에 대해서 분

석해 보기 위해서 기존에 센서 네트워크와 자주 비교되는 ARM7T 프로세서에서 동일한 실험을 수행할 수 있도록 시뮬레이션 하였다. 그리고 이에 대한 실험 결과를 참조하여 제안한 시스템에 대한 성능을 비교 분석하고자 한다.

참고로 ARM7T는 MICA2와 비교할 때 논리적으로 대략 5.5배 정도의 우월한 처리성능을 지닌 프로세서다. 이에 논리적으로 동일한 크기의 자료에 대해서 동일한 실험을 수행할 경우 대략 5.5배 정도의 우월한 성능을 보이는 경우라면 제안한 시스템에 대한 성능 평가가 신뢰할 만 한 것이라고 추정할 수 있다.

아래의 <표 4>는 각각의 프로세서에서의 동일한 1KB 데이터에 대한 공개키와 비밀키를 생성하는데 소용된 시간을 측정한 결과를 나타내고 이에 대한 비교 분석을 보여주고 있다.

〈표 4〉 ARM7T와의 성능 비교

구분	그룹키 생성 시간	비고
제안한 시뮬레이션 결과	590 ‖ 1560	ARM7T 시뮬레이션 환경이 MICA2 환경보다 약 6배 정도 빠른 처리 시간을 보임. (ARM7T 프로세서가 약 5.5배 빠름)
ARM7T 24MHz에서의 시뮬레이션 결과	98 ‖ 260	

<표 4>에서 나타내고 있는 바와 같이 공개키와 비밀키를 생성하기 위해서 소요되는 처리 시간을 동일한 실험환경에서 측정해 본 결과 약 6배 정도의 차이가 나타나고 있음을 알 수 있다. 이는 우리가 용인할 수 있을 정도의 성능을 제안한 시스템에서 발휘하고 있음을 보이는 것이다.

이를 통해서 우리의 제안한 시스템에 대한 타당성이 있음을 주장할 수 있을 것이다. 향후 이와 관련하여 보다 다양한 실험결과를 측정해 봄으로써 이를 더욱 확고히 하고자 한다.

6. 결 론

본 논문에서는 일반 네트워크 시스템에 비하여 상대적으로 제한적인 자원을 갖는 센서 네트워크에서 기존의 연구 방법들과 비교할 때 적은 연산 횟수와 데이터를 이용하여 센서 노드들에 대한 인증과 전송되는 데이터에 대한 비밀성을 보장할 수 있도록 하는 시스템을 제안하였다.

그리고 제안한 시스템이 설계 가능하도록 세부적인 동작 절차와 처리해야 할 데이터의 항목들에 대해서 정의하고 이를 시뮬레이션 환경에서 구현하였다. 구현된 시뮬레이션 프로그램을 동작시키고 이에 대한 2가지의 실험을 통해 제안한 시스템에 대한 처리 속도와 능력을 측정해 보았다. 마지막으로 논문에서 제안한 시스템에 대한 실험 결과가 현실적으로 타당한지 여부를 확인하기 위해서 기존에 대표적인 저용량 처리 장치로 인식되고 있는 ARM7T 프로세서에서 동일한 환경과 동일한 처리 동작에 대한 실험결과를 측정하고 이를 제안한 시스템의 처리결과와 비교 분석하였다.

이를 통해 최종적으로 제안한 시스템이 현실 적용 가능성이 높음을 입증하였다. 하지만 시뮬레이션 환경에서 제한된 실험을 작은 트래픽만으로 키를 생성/관리하고 만족할 만한 보안성을 제공하였다. 이에 향후 추가적으로 연구해야 할 내용은 원형좌표계에서 연산량을 줄일 수 있는 키 생성 알고리즘과 FBS와 BS의 초기 설정 이후에 노드들이 키 생성과 재 배열등을 분산처리할 수 있는 방법이 필요할 것으로 사료된다.

참 고 문 헌

[1] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. D. Tygar, "SPINS : Security Protocols for Sensor Networks", Proc. of the 7th

ACM/IEEE International Conference on MobiCom, 2001.

[2] S. Slijepcevic, M. Potkonjak, V. Tsiatsis, S. Zimbeck, and M. B. Srivastava, "On Communications Security in Wireless Ad-Hoc Sensor Networks", Proceedings of the 11th IEEE International Workshops on Enabling Technologies : Infrastructure for Collaborative Enterprises (WETICE'02), Pittsburgh, Pennsylvania, USA, June 10-12, pp. 139-144, 2002.

[3] Y. Kim, A. Perrig, and G. Tsudik, "Simple and Fault-tolerant Key Agreement for Dynamic Collaborative Groups", Proc. of 7th ACM Conference on Computer and Communications Security, pp. 235-244, November, 2000.

[4] Jing Deng, Richard Han, and Shivakant Mishra, "Security Support For In-Network Processing in Wireless Sensor Networks", First ACM Workshop on the Security of Ad Hoc and Sensor Networks (SASN 2003), held in conjunction with the ACM Conference on Computer and Communication Security (CCS'03), Fairfax, VA, USA, pp. 83-93, Oct., 2003, acceptance ratio 14/71.

[5] L. Eschenauer and V. D. Gligor, "A Key-management Scheme for Distributed Sensor Networks", In Proceedings of the 9th ACM Conference on Computer and Communications Security, ACM Press, pp. 41-47. 2002.

[6] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks", In CCS '03 : Proceedings of the 10th ACM Conference on Computer and Communications Security, ACM Press, pp. 52-61, 2003.

[7] S. Zhu, S. Setia, und S. Jajodia, "Leap : Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks", In: Proceedings of the 10th ACM Conference on Computer and Communication Security, ACM Press, S. 62-72, 2003.

홍 성 식

1989년 광운대학교 전자계산학과 (학사)
1992년 광운대학교 전자계산학과 (공학석사)
1994년~현재 혜전대학 교수

유 황 빈

1968년 인하대학교 전자공학과 (학사)
1975년 연세대학교 전자공학과 (공학석사)
1984년 경희대학교 전자공학과 (공학박사)
1981년~현재 광운대학교 컴퓨터소프트웨어학과 교수