논문 2006-43TC-7-1

계층적인 이동 센서 네트워크에서 회귀모델을 이용한 분산 키 관리

(Distributed Key Management Using Regression Model for Hierarchical Mobile Sensor Networks)

김 미 희*, 채 기 준*

(Mihui Kim and Kijoon Chae)

요 약

본 논문에서는 계층적인 이동 센서 네트워크에서 하위 센서 노드의 인증이나 센싱된 정보의 암호화를 위해 사용할 수 있는 키를 관리하기 위하여 키 선분배를 기본으로 키 재분배 방법을 제공하는 키 관리 메커니즘을 제안한다. 본 키 관리의 특징은 첫째, 중앙 관리의 약점을 극복하기 위해 키 관리를 sink 노드뿐 아니라 aggregator 노드들에 분산시켰다. 둘째, sink 노드는 회귀모델을 사용해 키를 생성 관리하여, 이미 분배된 키에 대해서는 어느 노드에게 어떤 키를 분배했는지 또는 그 키 자체를 저장하지 않고, 노드가 메시지에 첨부하여 전해주는 키 정보를 이용해 사용된 키를 간단히 계산하기 위한 정보만 저장하고 있다. 한편 기존 키 선분배에서는 키 선분배 후 키의 갱신에 대한 메커니즘이 제공되지 않았고, 네트워크 내 센서 노드가 확장되는 경우 이를 지원하도록 키 정보를 확장하기가 용이하지 않다는 단점이 있다. 이에 본 논문의 세 번째 특징으로써 기존 키 선분배 방식에서 제공되었던 센서 포획에 대한 탄력성(resilience), 즉 λ-security 특성을 제공하면서, 넷째 기존 방법의 단점을 보완하기 위해 노드 확장 시 키 풀의 확장이 용이하고, 배치된 노드에 대한 주기적인 키 재분배를 통해 키의 신규성(freshness)을 제공하며, 이동 노드에 대해 새로운 키 분배 방법을 제공하는 특징을 갖고 있다. 다섯째, 본 메커니즘은 키와 노드간의 매핑관계를 고정시키지 않음으로써 노드의 익명성 및 노드 이동 시 불추적성을 제공하고 있다. 마지막으로 본 논문에서는 기존 키 관리와의 특징 비교와 통신·계산·메모리 측면에서의 오버헤드 분석을 통해 제안된 키 관리의 성능을 분석한다.

Abstract

In this paper, we introduce a novel key management scheme that is based on the key pre-distribution but provides the key re-distribution method, in order to manage keys for message encryption and authentication of lower-layer sensor nodes on hierarchical mobile sensor networks. The characteristics of our key management are as follows: First, the role of key management is distributed to aggregator nodes as well as a sink node, to overcome the weakness of centralized management. Second, a sink node generates keys using regression model, thus it stores only the information for calculating the keys using the key information received from nodes, but does not store the relationship between a node and a key, and the keys themselves. As the disadvantage of existing key pre-distributions, they do not support the key re-distribution after the deployment of nodes, and it is hard to extend the key information in the case that sensor nodes in the network enlarge. Thirdly, our mechanism provides the resilience to node capture(λ-security), also provided by the existing key pre-distributions, and fourth offers the key freshness through key re-distribution, key distribution to mobile nodes, and scalability to make up for the weak points in the existing key pre-distributions. Fifth, our mechanism does not fix the relationship between a node and a key, thus supports the anonymity and untraceability of mobile nodes. Lastly, we compare ours with existing mechanisms, and verify our performance through the overhead analysis of communication, computation, and memory.

Keywords : 계층적인 이동 센서 네트워크, 분산 키 관리 메커니즘, 회귀모델, 키 선분배, 키 재분배.

* 정회원, 이화여자대학교 컴퓨터학과 (Ewha Womans University)
※ 본 연구는 정보통신 및 정보통신연구진흥원의 대학 IT연구센터 육성·지원사업의 연구결과로 수행되었음
접수일자: 수정완료일:

Ⅰ. 서 론

최근 유비쿼터스(ubiquitous) 컴퓨팅 구현을 위한 기반 네트워크로서 초경량, 저전력의 많은 센서들로 구성된 무선 센서 네트워크에 대한 연구가 활발히 진행되고 있다. 특히 센서 네트워크에서는 공격자가 트래픽을 쉽게 엿들을 수 있고 주변 노드에게 잘못된 정보를 제공함으로써 센서 네트워크의 노드로 흉내 낼 수 있기 때문에, 보안성을 제공하는 것이 중요하다. 이러한 서비스를 위하여 고려해야 할 사항은 노드 캡쳐에 의한 위협의 가능성, 노드의 제한적인 자원, 동적 토폴로지 변화 등이다.

게다가 센서 네트워크에서 정적인 노드뿐 아니라 이동 노드에 대한 연구와 관심이 증가되고 있다[1,2]. 이동 노드는 센서 네트워크의 가용 능력(capacity)을 증가 시킬 수 있다. 예를 들어 물리적으로 이동하며 데이터를 수집하여 전송할 수 있고, 정적 노드의 충전 및 고장을 수리할 수 있다. 그러나 노드에 대한 이동성 가정은 보안 서비스 측면에서는 더욱 그 문제를 해결하기가 복잡해지고 어려워진다.

안전한 센서 네트워크의 통신을 위한 기본적인 보안 서비스로서 키 관리 방법에 대한 다양한 연구[3-7]가 진행되어 왔다. 그러나 센서 네트워크의 응용과 구조가 다양하므로 그 모든 구조에 적합한 키 관리 방법은 존재하지 않으며, 특정 네트워크 특성에 맞는 최적화된 키 관리가 제공되어야 한다.

센서 네트워크는 응용에 따라 다양한 구조가 가능한데, 대표적인 배치로서 무작위 배포에 의한 평평한(flat) 구조[6]와 중앙 노드(SINK 혹은 BS)를 중심으로 퍼져있는 계층적인 구조[8,9,10]가 있다. 본 논문에서 가정하는 센서 네트워크 구조는 sink 노드와 aggregator 노드(AN), 센서 노드(SN)들이 계층적으로 구성된 네트워크를 가정한다. 이러한 구조는 센서들이 센싱한 정보를 유용하게 사용하기 위하여 유선 네트워크에 전달해 주기 위해 멀티홉 무선 구간을 거쳐 SINK로 전달해 주는 구조이다. 이때 에너지 효율적인 운용을 위해 많은 노드들이 센싱한 같은 이벤트에 대한 정보는 중간 노드인 AN에 의해 종합되어 전달될 수 있다.

이러한 구조의 트래픽 특성에 맞추고 기존 키 관리 기법들의 문제점을 보완하기 위하여, 본 논문에서는 센서들이 센싱한 정보의 암호화나 인증을 위하여 사용할 수 있는 키를 관리하되 SINK 뿐 아니라 AN에 분산 관리하는 키 관리 기법을 제안한다. 본 키 관리의 특징은 회귀모델을 사용하여 키를 생성 관리하고, 키 관리 노드에서는 키의 재분배를 위한 키 스페이스를 제외하고, 이미 분배된 키에 대해서는 어느 노드에게 어떤 키를 분배했는지 또는 그 키 자체를 저장하지 않고, 노드가 메시지에 첨부한 키 정보를 사용하여 사용된 키를 계산하기 위한 정보만 저장하고 있어, 관리 노드의 위협에 위한 피해를 최소화 하였다. 이를 통해 키를 계산할 수 있는 노드를 소수의 노드(SINK, AN)로 한정하였지만 이를 계층적으로 분산화 하였고, 노드 캡쳐에 대해 λ-security 특성 제공한다. 또한, 용이한 키 스페이스의 확장 및 키 풀 확장을 통해 센서 노드 수 확장에 따른 키의 확장성을 제공하며, 주기적으로 키를 재분배함으로써 키의 신규성(freshness)을 제공하고, 이동 노드에 대한 새로운 키 분배 방법을 제공한다. 마지막으로 키와 노드간의 매핑관계를 고정시키지 않음으로써 노드의 익명성 및 이동 노드의 불추적성을 제공하고 있다.

본 논문의 구성은 다음과 같다. Ⅰ장의 서론에 이어, Ⅱ장에서는 관련 연구로서 본 논문의 키 관리에서 기본으로 사용하고 있는 회귀모델을 설명하고, 기존에 제안된 키 관리 메커니즘의 특징을 비교 설명하고자 한다. Ⅲ장에서는 본 논문에서 제안하고 있는 계층적인 센서 네트워크에서의 분산 키 관리 메커니즘을 기술한다. Ⅳ장에서는 제안된 메커니즘의 통신·계산·메모리 측면의 오버헤드를 분석하여 기존 키 관리 메커니즘과 비교하고, 마지막으로 결론으로써 본 논문을 마치고자 한다.

Ⅱ. 관련 연구

1. 회귀모델

회귀분석이란 이미 알려진 독립변수(independent variable)들로부터 하나의 종속변수(dependent variable)의 값을 예측하는데 사용되는 통계 방법 중의 하나이며 이를 위해 회귀모델이 사용된다. 예를 들어 어떤 플라스틱 제품의 견고도가 이 제품을 만드는 기계의 온도와 만드는 시간에 어떤 연관성이 있다면, 이들 변수간의 함수관계를 규명하기 위해 사용되는 방법이다. 이 예에서 견고도는 종속변수라 부르고, 온도와 시간과 같이 종속변수에 영향을 주는 변수를 독립변수라고 한다[11].

대표적인 회귀분석 방법인 최소 제곱법(least square)은 종속변수의 측정치 y와 행렬 b에 의해 계산된 값인 y_k와의 차이에 대한 제곱 값이 최소($\min\sum(y-y_k)^2$)가 되도록 하는 행렬 b를 추정하며 구하는 공식은 다음과 같다.

$b = (X'X)^{-1}X'y$, X'는 X의 전치행렬

회귀분석에서는 각 변수의 표현 방법으로 행렬을 많이 사용하는데, 이는 표현이 간결할 뿐만 아니라 독립변수의 수에 상관없이 회귀분석을 다루는 행렬 표현 방법이 일정하기 때문이다.

그림 1은 독립변수의 계수가 $(\lambda+1)$, 표본 수가 n일 때의 중회귀(Multiple Regression) 모형으로 본 논문에서는 다음과 같은 중회귀 모형을 사용하는데, 이는 종속변수를 예측하기 위함이 아니라 노드가 송신해 주는 $(\lambda+1)$의 키 정보 X_i를 이용하여 사용된 키 y_{ki}를 계산해 내는 새로운 응용으로써 사용한다. 이에 대한 상세한 내용은 Ⅲ장에서 설명한다.

$$y_k = X(X'X)^{-1}X'y = X \times b$$

$$y_k = \begin{Bmatrix} y_{k1} \\ y_{k2} \\ y_{k3} \\ \vdots \\ \cdot \\ y_{kn} \end{Bmatrix} = \begin{Bmatrix} 1 & x_{11} & x_{21} & \cdots & x_{\lambda 1} \\ 1 & x_{12} & x_{22} & \cdots & x_{\lambda 2} \\ 1 & x_{13} & x_{23} & \cdots & x_{\lambda 3} \\ \vdots & \vdots & \vdots & & \vdots \\ \cdot & \cdot & \cdot & & \cdot \\ 1 & x_{1n} & x_{2n} & \cdots & x_{\lambda n} \end{Bmatrix} \times \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ \cdot \\ b_\lambda \end{Bmatrix} = \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ \cdot \\ X_n \end{Bmatrix} \times \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ \cdot \\ b_\lambda \end{Bmatrix} = X \times b$$

그림 1. 일반 중회귀 모형
Fig. 1. General multiple regression model.

2. 기존 키 관리 메커니즘

센서 네트워크에서 안전한 통신 인프라를 구축하기 위한 가장 기본적인 보안 프로토콜로서 키 관리 프로토콜을 꼽을 수 있다. 이는 배치 전 각 노드에 저장해 둔 비밀정보를 이용해 설치된 후, 센서 노드들 간에 안전한 통신을 제공하기 위한 키 분배 및 갱신 등의 총체적 관리를 제공하는 것으로 센서 네트워크에서는 다음과 같은 물리적인 한계 및 이에 따른 요구조건들이 제시되고 있다[12].

- 노드 캡쳐에 의한 위협 가능성: 센서 네트워크 환경 자체가 물리적 공격에 취약한 경우가 많기 때문에 이러한 공격의 피해를 국지화(localization) 해야 한다.
- 노드의 제한적인 자원의 한계: 많은 통신량이나 계산량 또는 정보 저장량을 요구하는 메커니즘은 현실적이지 않다.
- 동적인 토폴로지 변화: 초기 노드 설치 후, 에너지 고갈에 의한 노드 제거 및 노드 추가 설치 또는 비정상적인 노드의 제거에 의한 토폴로지 변화에 적응할 수 있는 키 관리 메커니즘이어야 한다.

이러한 요구사항에 맞춰 표 1과 같은 다양한 키 관리 방법들이 제안되어 왔다. 표 1은 제안된 대표적인 키 관리 방법들의 특징과 각 방법의 취약점을 정리한 표이다.

Ⅲ. 제안하는 분산 키 관리 메커니즘

제안하는 분산 키 관리 메커니즘은 계층적인 이동 센서 네트워크의 특성에 맞춰 다음과 같은 장점을 제공하고 있다.

- 키 관리 노드가 키와 할당된 노드와의 관계 정보를 저장하지 않음으로써 키 관리 노드의 위협에 대한 안전성 제공
- 노드 캡쳐에 대해 λ-security 특성 제공
- 용이한 키 스페이스의 확장과 다수의 키 스페이스로 구성된 키 풀 사용 및 키 스페이스의 추가/갱신을 통한 키 관리의 확장성(scalability) 제공
- 간단한 키 스페이스의 변형 및 주기적인 노드의 키 재분배를 통한 키 신규성 제공
- 노드가 랜덤하게 키를 선택하여 사용함으로써 노드의 익명성(anonymity) 및 이동에 의한 노드의 불추적성(untraceability) 제공
- 키 관리의 분산화를 통해 로드 분산 및 키의 노출에 대한 피해의 국지화(localization)

이에 대해 1)키 관리의 기본, 2)키 추가/삭제, 3)키 재분배, 4)키 스페이스 추가/삭제, 5)이동 노드에 새로운 키 분배로 나누어 설명하고자 한다.

1. 키 관리의 기본

본 논문에서 가정하는 계층적인 센서 네트워크의 트래픽 흐름은 첫째, SINK에서 관심 있는 이벤트를 등록하기 위해 SN들에게 이벤트 등록 메시지를 브로드캐스팅하거나 특정 SN에게 제어메시지 전달을 위한 하향 트래픽 흐름과, 둘째 이벤트 발생시 SN에서 감지된 센싱 정보를 상향 노드(SINK 방향의 노드)에게 전달해 주면 해당 지역의 AN 노드가 전달받아 적절한 처리를 통한 총체적 데이터를 SINK에게 전달해 주는 상향 트래픽 흐름으로 구성된다. 이러한 "요청과 데이터 포워딩(query and data forwarding)" 통신 구조는 가장 기본적인 센서 통신 구조로서 많은 기존 논문에서 가정하고 있다[5,13,14]. 또한 노드의 배치 방법은 SINK와 AN은 고

표 1. 기존 키 관리 메커니즘의 특징과 취약점 비교
Table 1. Comparison of existing key managements.

방식	특징	취약점
공유키 기반[3]	- 전 네트워크에서 공유된 키로 암호화 및 인증에 사용되어 효율성 및 단순성 제공	- 노드 캡쳐에 의한 피해가 전 네트워크에 미침 - 별도의 키 갱신 프로토콜이 필요
중앙 집중적 키관리[4,5]	- BS가 안전하다는 가정 하에 각 노드는 BS와의 공유키를 갖고 설치된 후, BS를 통해 다른 노드와의 링크키 설정	- BS가 장애나 공격에 노출 시 피해가 전 네트워크에 미침 - BS를 통한 링크키 설정에 높은 통신 오버헤드
랜덤 키 선분배[16]	- 설치 전, 이미 생성된 키 풀에서 임의의 개수의 키를 분배 받고, 이웃 노드와 공통키 존재하면 이를 사용하고, 존재하지 않으면 다른 연결된 이웃노드를 통해 링크키 설정	- 센서 네트워크의 밀도에 따라 전체 네트워크가 완전히 연결되지 않은 상태가 될 수 있음 - 해당 키는 소수 노드의 전복에 매우 취약함 - 선분배 방식이므로 키의 신규성 제공 불가
랜덤 pairwise 키 선분배[6,7]	- 랜덤키 선분배를 기본으로 Blom 스킴[6]과 다항식에 의해[7] 각 노드 사이에 유일한 키가 간단한 계산에 의해 공유되어 노드 인증이 가능함	- 각 노드 사이에 공통적인 키 스페이스 또는 다항식을 갖고 있지 않다면 다른 중간 노드를 통해 패스키를 설정하는데, 한 노드의 전복이 공격자가 위장하고 있는 노드들을 합법화 할 수 있음[17] - 선분배 방식이므로 키의 신규성 제공되지 않고, 네트워크 내 센서 노드의 수 증가, 노드의 이동으로 토폴로지가 변하는 경우 관리하는 키 정보를 확장하기 용이하지 않음
공개키 기반[18,19]	- 최근 제약된 환경으로 키 분배 메커니즘으로 배제되었던 공개키 방식에 대해 적절한 알고리즘과 파라미터 선택 및 최적화 구현을 통한 하드웨어 기반 공개키 암호 가능성 제시[18] - Mica2 플랫폼에서 공개키 암호 구현 결과를 제시함으로 사용 가능성을 보임[19]	- 수행 빈도가 높지 않은 키 분배를 위한 공개키 기술의 적용을 제시했지만, 이를 위해서는 공개키 기반구조가 확립되어야 한다는 기본 가정이 따름
계층적인 키관리[20,21]	- 계층적인 통신 구조의 특징에 맞춰 노드 간 선분배된 키에 의해 홉바이홉 메시지 암호화/복호화 과정을 포함하고 있고, 센서 추가/삭제에 대한 키 처리 과정을 제시[20] - 각 클러스터헤드가 클러스터 내에 사용될 그룹 키를 선분배된 공유키를 사용하여 계산[21]	- [20]에서의 중간 노드는 하위 노드 수에 비례하여 키를 저장해야 하며, 키를 생성할 수 있는 능력이 요구되고, 홉바이홉 메시지 암호화/복호화가 과부하게 될 수 있음 - [21]에서는 센서 포획에 대한 탄력성이 제공되지 않고, 특히 키 설정 시 노드 포획에 대한 위협에 취약함

정 배치를 가정하고 있고, SN은 해당 AN 주위에 랜덤 또는 고정 배치를 가정한다. 노드 배치 후 계층적인 센서 토폴로지 구성은 [15]에서 나온 방법 등을 활용할 수 있다고 가정한다.

이러한 계층적인 센서 네트워크 구조에서 사용하는 키는 크게 1)상향 트래픽 흐름을 위한 키 K_{up}와 2)하향 트래픽 흐름을 위한 키 K_{down}가 있다. 후자의 키는 한 키 관리 노드에서 센서 노드들로의 브로드캐스팅이나 유니캐스팅을 위해 사용되는 키로 한 구역 내에서 유일하며 구역 내 모든 노드가 공유하고 있다. 이 키는 키 관리 노드에 의해 주기적으로 재분배되며, 새로운 키 K_{down}''를 재분배할 때 기존의 키 K_{down}'를 사용하여 안전하게 전달한다. 이 후, 본 논문에서는 센서 노드가 제공하는 정보의 무결성과 기밀성 및 하단 계층에 있는 노드에 대한 인증에 사용되는 상향키 K_{up} 관리에 대하여 집중하여 설명하고자 한다. 본 논문에서 제공되는 상향키는 회귀모델을 사용하여 생성되고 관리된다. 즉, 그림 1에서 하나의 키 스페이스 y_k는 $(n \times (\lambda+1))$의 키 정보 행렬(Key Information Matrix) X 와 $(n \times 1)$의 키 생성 행렬(Key Generation Matrix) y로부터 계

산된다. 여기에서 행렬 X와 y는 유한체(Finite Field)인 $GF(q)$ 상에서 정의되며, q는 하나의 키 y_{ki}의 길이가 64 비트일 때, 2^{64}보다 큰 소수 중 가장 작은 소수로써 선택하면 된다[6]. 앞서 설명했듯이, 본 논문에서는 회귀 분석의 원래 목적인 종속변수의 값 y_k를 예측하기 위하여 회귀모델을 사용하는 것이 아니라 y_k를 키들의 집합인 키 스페이스로 사용하고 이 키는 키 정보 행렬 X와 키 생성 행렬 y로부터 최소 제곱법에 의해 계산되어 사용한다. 이렇게 생성한 키와 키 정보 값을 노드에게 같이 분배해 주고, 키 관리 노드는 이에 대한 관계 정보를 전혀 저장하지 않고, 노드가 키 사용 시 키 정보도 함께 제공함으로써 키 관리 노드는 센서 노드로부터 수신한 키 정보와 키 생성 시 생성된 행렬 b를 가지고 간단한 행렬 곱셈 연산을 통해(λ번의 모듈러 곱셈(Modular Multiplication)) 키를 계산하여 사용하게 된다. 키 관리라는 새로운 응용을 위한 회귀모델의 적용 가능성은 첫째, 공격자(adversary)에게 키 정보 행렬 X에서 $(\lambda+1)$개의 행, 즉 $(\lambda+1)$개의 키 정보가 노출되지 않는 한 다른 키를 계산해 내기 위한 행렬 b를 알아낼 수 없다는 키 관리의 안정성을 제공할 수 있고, 둘째 키 관리 노드는 분배된 키와 노드와의 관계 정보를 저장하지 않고 노드가 제공하는 키 정보를 가지고 간단히 키를 계산함으로써 키 관리 노드의 위협에 의한 전체 키의 즉각적인 노출이라는 취약점을 보완할 수 있다. 이러한 키 스페이스는 노드 캡쳐에 의한 키 스페이스의 노출 정도를 낮추어 안전한 키 관리를 행할 수 있도록 다수의 행렬 y로부터 계산된 다수의 키 스페이스를 키 풀로 생성하여 관리할 수 있다.

본 논문에서 제공되는 키 관리 메커니즘은 회귀모델을 통해 생성된 키 풀에서 랜덤하게 선택한 키를 각 노드에 저장하고 배치하는 키 선분배 방법을 기본으로 하며, 키의 신규성을 제공하기 위해 주기적으로 키를 재분배하는 방법을 제공한다. 초기 노드 배치 시, SINK(별도의 키 관리 노드가 존재할 수도 있음)는 모든 키 스페이스의 행렬 b를 저장한다. 또한 일정 계층(layer) 및 구역에서 센싱된 정보의 통합 및 전송 역할을 맡은 AN에게는 키 풀에서 μ개의 키 스페이스를 랜덤하게 선택하여, 해당 키 스페이스에 대한 행렬 b를 저장한다. SN인 경우에는 SN이 배치될 곳의 AN에 저장된 μ개의 키 스페이스 중에서 키 스페이스를 랜덤하게 선택하여, 키 스페이스 번호 j 와 해당 키 스페이스에서 랜덤하게 선택된 키 y_{ki}^{j}와 해당 키 정보 X_i, 즉$(1\ x_{1i}\ x_{2i} \cdots x_{\lambda i})$

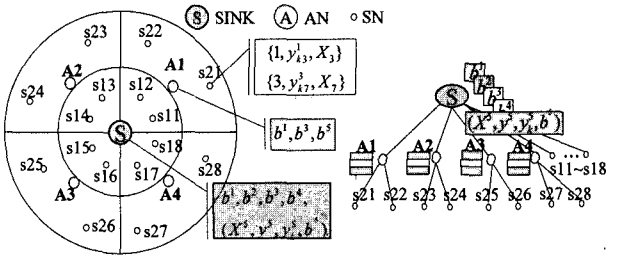


그림 2. 계층적인 센서 네트워크의 예와 SINK, AN, SN이 갖는 정보의 예

Fig. 2. Example of hierarchical sensor network and information example of SINK, AN, and SN.

를 저장시킨 후(이하, 이 3개의 정보의 한 세트를 키셋이라고 칭함), SN을 배치한다. 이 3개의 정보는 노드의 저장 공간에 따라 다수 개가 랜덤하게 선택되어 저장되고, 노드가 키 사용 시마다 랜덤하게 선택하여 사용할 수 있다. 즉, 한 노드가 전송하는 메시지에서 그 구역의 하향키 K_{down}로 암호화해서 보여 지는 키 정보가 변하기 때문에 노드의 익명성을 제공하여 공격자에 의한 트래픽 분석을 어렵게 만들고, 노드가 이동하는 환경에서는 노드의 불추적성을 제공할 수 있다. 그림 2는 1개의 SINK와 4개의 AN, 16개의 SN이 2계층으로 구성된 계층적인 센서 네트워크에서 5개의 키 스페이스로 구성된 키 풀이 관리되는 예시이다. 각 AN에게는 2개의 키 스페이스를 랜덤하게 선택하여 각 스페이스에 대한 행렬 b와 재분배를 위한 키 스페이스의 행렬 b^5를 저장하여 배치하고, 각 SN에게는 2개의 키셋 {키 스페이스 번호, 키, 키 정보}가 선택되어 저장된다고 할 때, SINK와 AN, SN에 저장된 키에 관련된 정보를 예시하고 있다.

각 SN은 센싱한 정보를 암호화하거나 정보의 무결성 제공을 위해 인증코드(MAC, Message Authentication Code)를 작성하거나, 상위 노드에게 자신을 인증받고자 할 때, 자신이 갖고 있는 키 중에 랜덤하게 선택하여 선택된 키 y_{ki}^{j}를 사용하고, 관련 정보를 메시지로 전송할 때 사용된 키에 대한 키 스페이스 번호와 키 정보 $\{j, X_i\}$를 함께 전송하게 된다. 다음은 SN이 선택한 키를 통해 암호화 및 인증코드를 첨부한 메시지의 예이다.

$$E_{K_{down}}(j|X_i)|E_{y_{ki}^{j}}(\text{센싱된 정보})|MAC_{y_{ki}^{j}}(j|X_i|\text{센싱된 정보})$$

$E_{key}(M)$: key를 가지고 메시지 M의 암호화

$MAC_{key}(M)$: key를 가지고 메시지 M에 대한 인증코드 작성

이러한 정보를 AN이 수신한다면, 수신된 정보에서

키 스페이스 번호를 추출하여 해당 키 스페이스 번호에 대한 행렬 b를 찾아 행렬 모듈로 곱셈(Modular Multiplication) 연산 $(X_i \times b^j)$을 수행하여 사용된 키 y_{ki}^j를 계산하고, 정보에 대한 인증이나 복호화 작업을 수행한다. 또한 AN이 같은 이벤트에 대해 다수의 SN으로부터 정보를 수신했다면, 적당한 인증 및 복호화 작업 후, 정보를 통합(aggregation)하여 이를 상위 AN이나 SINK 노드에게 전달한다. 그 때, 수신된 메시지에서 사용된 키 중에 랜덤하게 선택하여 통합된 정보를 암호화하거나 인증코드를 만들 때 사용하고, 생성된 메시지와 함께 사용된 키에 대한 키 스페이스 번호와 키 정보 $\{j, X_i\}$를 함께 전송한다. SINK도 마찬가지로 하단 노드로부터 메시지 수신 시, 수신된 메시지에 포함된 키 스페이스 번호와 키 정보를 가지고 키를 계산하여 메시지 복호화 및 인증 등에 사용하게 된다. 그림 3은 수신된 메시지에 담긴 키 정보가 $\{1, X_3\}$인 경우, AN 또는 SINK에서 키 y_{k3}^1를 계산하는 연산 과정을 나타낸다.

본 논문의 키 관리 메커니즘에서 키 계산을 위한 통신 오버헤드는 키를 사용한 메시지에 첨부되는 키 정보 $\{j, X_i\}$가 되는데, 전체 키 스페이스의 개수가 ω이고 키 정보에서 첫 번째 열의 값인 1은 전송하지 않는다고 했을 때, 키 정보는 $(x_{1i}\ x_{2i}\ x_{3i} \cdots x_{\lambda i})$가 되므로 $(\log_2\omega + \sum_{q=1}^{\lambda}|x_{qi}|)$ 비트가 된다. 여기에서 하나의 키 y_{ki}의 길이가 64 비트라고 했을 때, 오버헤드는 $(\log_2\omega + 64\lambda)$ 비트가 되므로 통신 오버헤드는 ω와 λ 값에 비례하여 커지게 될 수 있다. 이러한 오버헤드를 줄이기 위해 키 정보 행렬 X를 구성할 때 그림 4와 같이 구성하면 보안적 요소를 그대로 유지하면서 첨부해야 되는 키 정보의 오버헤드를 $\log_2\omega + |x_{qi}|$비트로 줄일 수 있다. 즉, 키 정보 X_i의 시드(seed) 값인 s^i 값만 전달해 주면 키 계산 시 전체 키 정보인 $(1\ s^i\ (s^i)^2\ (s^i)^3 \cdots (s^i)^\lambda)$을 계산하여 사용할 수 있다. 이 때 키 정보 행렬의 시드인 s가 $GF(q)$의 생성자(primitive element)라면 $s^i (0 < i \le q-1)$는 $GF(q)$의 한 요소가 되며, i와 j가 같지 않다면 s^i와 s^j도 같지 않다고 알려져 있다[6].

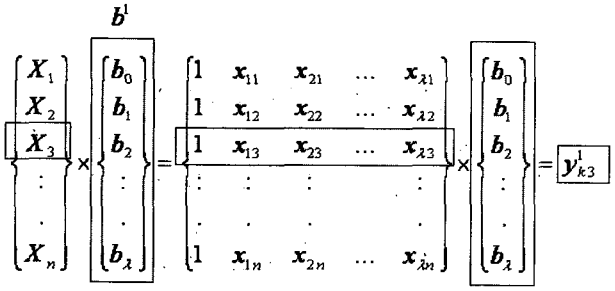


그림 3. 수신된 메시지에 담긴 키 정보가 $\{1, X_3\}$인 경우, 키 관리 노드에서 계산되는 키 y_{k3}^1

Fig. 3. A calculated key y_{k3}^1 at key management node, with key information $\{1, X_3\}$ of received message.

SINK나 키 관리 센터(KDC, Key Distribution Center)와 같이 최상위에서 키를 관리하는 노드에서는 키 풀을 관리할 때, 관리 노드의 노출에 대한 위협을 줄이기 위해 이미 분배된 키에 대한 스페이스(*AD-Key Space*, Already Distributed-Key Space)와 새로 네트워크에 추가된 노드나 이동해 온 노드의 키 분배와 키 갱신을 목적으로 사용되는 키 스페이스(*D-Key Space*, Distributing-Key Space)로 나누어 관리할 수 있다. *AD-Key Space*에 대해서는 해당 키 스페이스에 포함된 키가 사용되었을 때, 그 키의 계산을 위해 행렬 b만을 저장하고 있고, 해당 키 스페이스의 행렬 X, y 및 어느 노드에게 어떤 키셋을 분배하였는지에 관한 관계 정보는 저장하지 않는다. 이에 반해 *D-Key Space*에 대해서는 분배하고 있는 키 스페이스 y_k및 키 정보 행렬 X, 키 생성 행렬 y, 키의 용이한 계산을 위한 행렬 b를 저장하고 있고, 그러나 어느 노드에게 어떤 키셋이 분배되

$$\begin{Bmatrix} 1 & s & s^2 & \cdots & s^\lambda \\ 1 & s^2 & (s^2)^2 & \cdots & (s^2)^\lambda \\ 1 & s^3 & (s^3)^2 & \cdots & (s^3)^\lambda \\ & & & \vdots & \\ 1 & s^n & (s^n)^2 & \cdots & (s^n)^\lambda \end{Bmatrix}$$

그림 4. 키 정보 행렬의 예제

Fig. 4. Example of key information matrix.

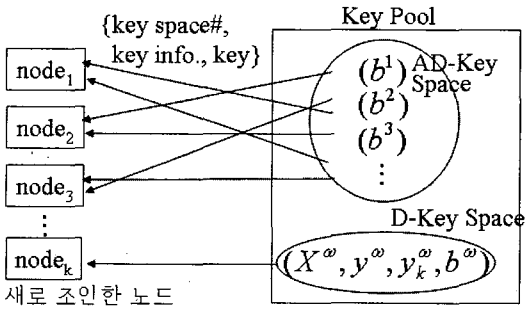


그림 5. 키 풀 관리

Fig. 5. Key pool management.

었는지는 저장하지 않는다. 그림 5는 키 풀 관리에 대한 예시이다.

2. 키 추가/삭제

본 논문의 키 관리 방법에서는 센서 네트워크의 노드 수가 많아지거나 노드에게 새로운 키를 분배하기 위한 한 가지 방법으로서 추가적인 노드에 대한 키 분배 목적으로 사용하고 있는 키 스페이스(D-Key Space)의 키를 추가하거나 갱신하는 방법을 제공한다. 이를 위한 기본 아이디어는 이전 절에서 설명하였듯이 키를 만들 때 사용한 최소 제곱법을 이용한다. 즉 키를 생성할 때, 키 생성 행렬 y와 키 스페이스 행렬 y_k의 각 원소의 차이의 제곱이 최소($\min\sum(y-y_k)^2$)가 되도록 하는 최소 제곱법에 의해 y_k가 계산되는데, 이미 키 스페이스가 계산된 이후, 키를 추가할 때에는 추가할 키 y_{ki}와 대응되는 키 정보 y_i를 같은 값으로 사용하면($y_i = y_{ki}$), 키를 계산하는데 사용되는 행렬 b의 값은 변화가 없게 되므로 이러한 특성을 이용하여 키를 추가하게 된다. 키를 추가하는 방법은 그림 6에서처럼 우선 $GF(q)$를 따르는 새로운 키 정보 X_{n+1}를 생성하여 행렬 X에 추가한다. X_{n+1}을 기존의 행렬 b에 대응시켜 새로운 키 y_{kn+1}을 생성해 낸다. 그런 다음, 키 스페이스 행렬 y_k에 y_{kn+1}을 추가하고, 행렬 y의 $n+1$번째 행인 y_{n+1}로서 y_{kn+1}을 추가한다(즉 $y_{n+1} = y_{kn+1}$). 같은 방식으로 추가된 키에 한하여 삭제가 가능하며, 주기적인 키 추가, 삭제에 의해 키 스페이스의 신규성을 증가시킬 수 있다.

$$\begin{Bmatrix} 1 & x_{11} & x_{21} & x_{31} & \cdots & x_{i1} \\ 1 & x_{12} & x_{22} & x_{32} & \cdots & x_{i2} \\ 1 & x_{13} & x_{23} & x_{33} & \cdots & x_{i3} \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ \cdot & \cdot & \cdot & \cdot & & \cdot \\ 1 & x_{1n} & x_{2n} & x_{3n} & \cdots & x_{in} \end{Bmatrix} \times \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ \cdot \\ b_i \end{Bmatrix} = \begin{Bmatrix} X_1 \\ X_2 \\ X_3 \\ \vdots \\ \cdot \\ X_n \end{Bmatrix} \times \begin{Bmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ \cdot \\ b_i \end{Bmatrix} = X \times b = y_k = \begin{Bmatrix} y_{k1} \\ y_{k2} \\ y_{k3} \\ \vdots \\ \cdot \\ y_{kn} \end{Bmatrix} y = \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \cdot \\ y_n \end{Bmatrix} \Rightarrow \begin{Bmatrix} y_1 \\ y_2 \\ y_3 \\ \vdots \\ \cdot \\ y_n \\ y_{kn+1} \end{Bmatrix}$$

$$\{1 \;\; x_{in+1} \;\; x_{2n+1} \;\; x_{3n+1} \;\; \cdots \;\; x_{in+1}\} \qquad \{X_{n+1}\} \qquad \{y_{kn+1}\}$$

그림 6. 키 스페이스에서 새로운 키 추가
Fig. 6. Addition of a new key in a key space.

3. 키 재분배

본 논문의 키 관리의 기본은 우선 노드 배치 전, 키를 저장하는 키 선분배를 기본으로 하고 있지만, 기존의 공유키 기반의 키 선분배의 단점인 키 갱신에 의한 신규성을 제공하지 못하고 센서 노드가 이동하는 환경에서는 부적합하다는 단점을 개선하기 위하여 키 재분배 방법을 제공한다.

전체 키 풀을 관리하는 최상위 노드(그림 2에서는 SINK)에서는 주기적으로 키를 갱신시키기 위한 타이머를 갖고 있고, 해당 타이머가 타임아웃 되면 일정 시간(T_{redist}) 동안 키 재분배 기능을 수행하게 된다. 그림 2를 가지고 예를 들어 설명하면, SINK가 키 재분배 시간 동안 센서 노드 SN으로부터 직접 센싱된 정보를 수신하면, 수신된 정보에 포함된 키 스페이스 번호와 키 정보를 가지고 키를 계산하여 정보를 처리하고, 새로운 키셋 정보(복수 개 키셋도 가능)를 *D-Key Space*인 b^5에서 랜덤하게 선택하여 수신한 정보에 사용했던 기존 키를 가지고 암호화 및 인증코드를 작성하여 전송한다. 이를 수신한 SN에서는 복호화 및 인증 과정을 거친 후 정당한 키 재분배 메시지이면, 자신이 갖고 있던 다수의 키셋 중에서 새로 받은 키셋의 수만큼 랜덤하게 제거하고 새로운 것으로 대치하여 관리하게 된다.

이와는 달리 SINK가 키 재분배 시간 동안 AN으로부터 통합된 센싱 정보를 수신하면, 이에 대한 복호화, 인증 및 데이터 처리 후, 새로운 키를 분배하기 위해 다수의 키셋 정보를 D-Key Space인 b^5에서 랜덤하게 선택하고 남은 키 재분배 시간(T_{redist})과 함께 수신한 정보에 사용했던 기존 키를 가지고 암호화 및 인증코드를 작성하여 전송한다. 이를 수신한 AN에서는 남은 키 재분배 시간 동안, 하단 AN이나 SN들에게 새로이 수신한 다수의 키셋 정보를 이용하여 SINK와 유사한 키 재분배 과정을 수행하고, 키 재분배 시간 T_{redist}가 끝나면, AN은 재분배를 위한 키셋 정보를 삭제하게 된다. 이로써 키 재분배 시간 동안 센서 네트워크의 노드들의 키가 랜덤하게 갱신되게 된다. 그림 7은 이러한 키 재분배 과정을 예시하였고, 다음은 키 재분배를 위한 메시지의 예시이다.

$E_{K_{down}}(n|X_m)|E_{y^n_{km}}(\{j, X_i, y^j_{ki}\}^p) \mid MAC_{y^n_{km}}(n|X_m|\{j, X_i, y^j_{ki}\}^p)$

$\{j, X_i, y^j_{ki}\}$: 새로운 키셋

$\{n, X_m, y^n_{km}\}$: 수신된 메시지에서 사용된 키셋

$\{j, X_i, y^j_{ki}\}^p$: p개의 키셋을 붙인(concatenation) 메시지

$E_{key}(M)$: key를 가지고 메시지 M의 암호화

$MAC_{key}(M)$: key를 가지고 메시지 M에 대한 인증코드 작성

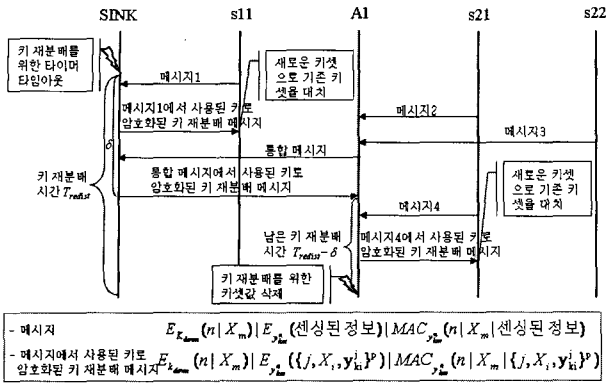


그림 7. 키 재분배 과정 예시
Fig. 7. Illustration of key redistribution process.

4. 키 스페이스 추가/삭제

본 논문의 키 관리 방법에서 키 신규성을 높이기 위한 또 하나의 방법으로 최상위 키 관리 노드(그림 2에서는 SINK)가 관리하고 있는 키 풀의 키 스페이스를 추가하거나 삭제를 통해서 새로 네트워크에 조인한 노드에 키를 분배하거나 기존 노드에 키 재분배 시, 새로운 키를 사용해서 분배할 수 있는 방법을 제공한다. 그림 2의 예를 통해 설명하면, 키 스페이스 추가는 Ⅲ.1에서 설명한 키 스페이스 생성과 마찬가지로 새로운 키 스페이스 (X^6, y^6, y_k^6, b^6)를 생성한 후, 기존의 *D-Key Space* (X^5, y^5, y_k^5, b^5)를 대신하여 이후의 키 분배 및 재분배에 사용되고, 기존에 *D-Key Space*는 *AD-Key Space*로 포함되어 키 계산을 위한 행렬 b^5만 남기고 나머지 X^5, y^5, y_k^5 행렬은 삭제하게 된다. 그리고 새로 생성된 키 스페이스의 행렬 b^6을 AN들에게 전달하되, 각 AN들로부터 최근에 수신된 메시지에 사용된 키셋을 가지고 암호화하여 안전하게 전달하고, 이를 수신한 AN은 자신의 하단에 또 다른 AN이 존재하면 같은 방법으로 안전하게 행렬 b^6을 전달한다.

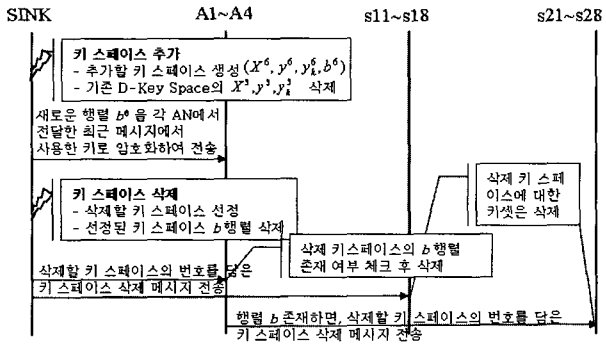


그림 8. 키 스페이스 추가/삭제 과정 예시
Fig. 8. Illustration of key space addition/deletion process.

또한 키 관리 노드의 저장 공간의 유한성으로 인해 일부 키 스페이스를 제거하고자 하는 경우, *AD-Key Space* 중에 가장 오래된 순서대로 혹은 랜덤하게 선택하여 삭제할 키 스페이스를 선정한 후, 키 관리 노드는 삭제할 키 스페이스의 번호를 담은 키 스페이스 삭제에 관한 메시지를 한 홉 이웃 노드에게 전달하고, 해당 키 스페이스에 대한 행렬을 저장 공간에서 삭제하게 된다. 이러한 키 스페이스 삭제 메시지를 받은 SN은 자신이 저장하고 있는 키셋 정보에서 삭제된 키 스페이스에 대한 키셋 정보를 삭제하게 되고, 또한 AN은 키 스페이스 삭제 메시지에 담긴 키 스페이스의 행렬을 갖고 있으면 이를 삭제하고 자신의 하단에 위치한 AN 또는 SN에게 해당 메시지를 전달하며, 그러한 행렬을 갖고 있지 않으면 이 메시지를 무시하게 된다. 그림 8은 이러한 키 스페이스 추가/삭제 과정을 예시하였다.

5. 이동 노드에 새로운 키 분배

본 논문에서는 노드의 이동성을 가정하였고, 이에 대해 키 관리 노드는 이동해 온 노드가 가지고 있는 키셋이 새로운 구역에서 사용할 수 없는 경우 새로운 키셋을 분배하는 메커니즘을 포함하고 있다. 그림 9에서 센서 노드 s21이 A1이 관할하는 구역에서 A5가 관할하는 구역으로 이동하는 예제를 가지고 설명한다. s21이 이동 후(1), 자신이 갖고 있던 키 (y_{k3}^1 또는 y_{k7}^3)를 가지고 자신이 센싱한 정보를 전송(2)한다. 이 정보를 전달 받은 첫 번째 키 관리 노드 A5는 자신이 갖고 있는 키 스페이스에 대한 키가 아니므로 이를 최상위 키 관리 노드 SINK S2에게 s21에서 전송한 메시지를 전달(3)해 준다. 이를 수신한 키 관리 노드 S2는 메시지의 키 정보를 가지고 키를 생성하여 MAC 코드를 생성해 봄으로써 노드를 인증하고, *D-Key Space* y_k^6에서 새로

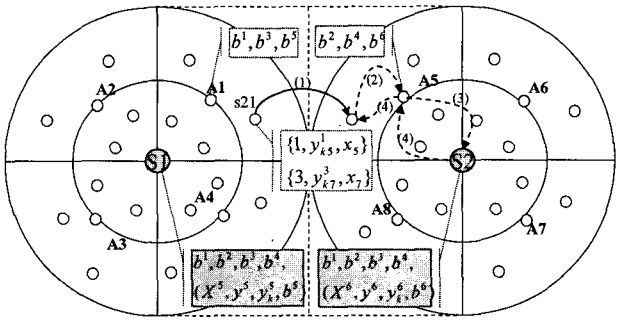


그림 9. 센서 노드 s21의 이동에 따른 키 분배
Fig. 9. Key distribution by the mobility of sensor node s21.

운 키셋을 랜덤하게 선택하여 수신된 메시지에서 사용되었던 키를 가지고 암호화를 하여 s21에게 안전하게 전달(4)해 준다. 이 예제와 달리 최상위 키 관리 노드도 이동 노드의 인증을 위한 키 스페이스 정보가 없으면, 이웃 최상의 키 관리 노드(그림 9에서는 S1)에게 인증을 요청한 후 인증 결과에 따라 키 분배 과정의 여부를 결정하게 된다.

Ⅳ. 성능 분석

본 장에서는 제안한 키 관리와 기존에 제안되었던 키 선분배 방법[6], 중앙 키 관리 기법[4], 안전한 계층적인 센서 네트워크 모델[5]과 비교를 통해 본 논문의 키 관리의 특징 및 장점을 살펴보고, 통신·계산·메모리 측면의 오버헤드를 분석하고자 한다.

본 논문의 키 관리 기법은 pairwise 키 선분배 기법[6]에서처럼 키 선분배를 가정하고 있지만, 배치 후 키 신규성을 제공하기 위한 키 재분배 방법을 제공하고 있다. 또한 [6] 기법에서는 pairwise 키를 만들기 위해 사용되는 $(\lambda+1)\times N$ 크기의 행렬 G에 대해 각 센서 노드에게 해당하는 열(column)을 분배하기 때문에, G 또는 A 행렬의 크기가 결정된 후에, 이동 환경 또는 노드의 재배치 및 추가 배치에 의한 노드 수의 증가에 유동적으로 대처하지 못한다. 또한 행렬 G의 열 값이 노드에 각각 매핑되기 때문에 트래픽 분석에 의한 공격에 대해 노드의 익명성 및 불추적성을 제공하지 못한다.

중앙 키 관리의 대표적인 예인 SPINS[4]의 SNEP 프로토콜에서 BS는 각 센서 노드 M_i와 사전에 마스터 키 K_i를 공유하고, 만약 노드 A와 B가 안전한 키를 설정한 후 통신하고자 하면, BS를 통해 4번의 메시지 교환 후, SK_{AB} 를 설정하여 사용한다. 그러나 두 노드의 안전한 메시지 교환을 위해 꼭 BS가 관여해야 하므로 중앙관리에 따른 여러 단점, 즉 BS 노드의 장애나 공격자에 대한 노출에 의한 피해가 전체 키의 노출로 확산될 수 있고, 트래픽 로드가 중앙으로 밀집된다는 단점을 포함하고 있다. 그러나 본 논문의 키 관리는 SINK에서 총체적인 키 관리를 담당하되 키의 계산에 의한 메시지 인증 및 복호화가 각 센서 노드의 배치 지역의 AN으로 분산되어 제공되므로 중앙 관리에 의한 단점을 극복할 수 있다.

또한 본 논문의 키 관리에서처럼 계층적인 센서 네트워크를 가정한 [5]에서는 BS를 중심으로 모든 노드가 두 홉만에 통신 가능한 2계층 네트워크를 가정하였고, 각 센서 노드 j는 자신의 고유키 Key_j와 공통키 Key_{BS}를 BS와 공유한다는 가정 하에 단대단으로(end-to-end) 안전한 통신이 가능한 보안 모델을 제안하였다. 그러나 가정하고 있는 네트워크가 너무 제한적이고, 데이터 취합이나 인증 포인트가 BS 하나로 구성되어 있기 때문에 키 관리의 중앙 집중화에 따른 취약점을 갖고 있다. 그러나 본 논문의 키 관리 메커니즘은 멀티홉 네트워크를 가정하여 키 관리를 분산시키고, 관리 노드에서도 키와 노드와의 관계를 저장하지 않음으로써 관리 노드의 노출이 각 노드의 키 노출과 연관되지 않도록 하였다.

1. 통신 오버헤드

본 논문에서 제안한 키 관리 메커니즘에서 기본적으로 키는 선분배 된 후 배치되므로 초기 키 분배를 위한 통신 오버헤드는 없으나, 키를 사용할 때 키 정보를 함께 보내야 하는 오버헤드가 발생하게 된다. 그러나 이러한 정보는 Ⅲ.1에서 설명했듯이 키 정보 행렬을 그림 4처럼 구성하면 전체 키 스페이스의 개수가 ω이고 키 정보 행렬의 한 원소가 x_{qi}라고 했을 때, $\log_2\omega+|x_{qi}|$ 정도로 사용되는 키 길이 정도의 오버헤드만 발생하게 된다. 그러나 이 오버헤드는 별도의 패킷으로 전송되는 것이 아니라 데이터 패킷 안에 첨부되어 정보이므로 추

표 2. 통신 오버헤드 비교 (bits)
Table 2. Comparison of communication overhead (bits).

	pairwise 키 선분배 기법[6]	제안하는 키 관리
키 사용을 위한 통신 오버헤드	$\log_2 N+\tau\cdot\log_2\omega+\alpha$	$\log_2\omega+\alpha$
	(N:전체 노드 수 , τ: 노드 당 분배 받는 키 스페이스의 정보 수, ω: 전체 키 스페이스 수, α: 시드값 길이)	
추가 메시지	- 해당 정보를 추가적으로 모든 노드가 브로드캐스트 함 - 두 노드 간에 공유되는 키 스페이스가 없는 경우, 다른 노드를 거쳐 공유키를 설정해야 함	- 데이터 패킷 안에 키 정보가 추가되어 추가 메시지 필요없음 - 상위 AN/SINK는 하위 노드가 사용하는 키 스페이스에 대한 정보를 항상 갖고 있어 추가적인 키 설정을 위한 메시지 교환 필요없음

가 메시지에 따른 오버헤드는 존재하지 않는다. 이를 pairwise 키 선분배 기법[6]에서의 키 설정 단계(Key Agreement Phase) 동안의 오버헤드((1)노드 ID, (2)노드가 갖고 있는 키 스페이스들의 인덱스, (3)노드가 갖고 있는 행렬 G의 행에 대한 시드 값)와 비교하면 표 2와 같다.

그리고 노드의 키 갱신을 위해 전송되는 메시지의 길이는 Ⅲ.3에서와 같은 메시지가 전송되고 그림 4와 같이 키 정보 행렬이 구성된다고 할 때, α가 키 정보의 한 요소나 키의 길이라고 하고 β가 MAC 코드의 길이라면, $(\log_2\omega+\alpha)+p(\log_2\omega+2\alpha)+\beta$ 비트가 된다.

2. 계산 오버헤드

Pairwise 키 선분배 기법[6]에서는 키를 계산하기 위해서 2λ 모듈로 곱셈(Modular Multiplication)의 계산 오버헤드가 필요하다고 언급하고 있고, 이는 RSA 공개키 기반의 암호화 알고리즘에 비해 훨씬 작은 계산양이라고 분석하고 있다[6]. 본 논문에서 키를 계산하기 위한 오버헤드는 수신된 키 정보의 시드 값에서 키 정보를 생성하는데 필요한 $(\lambda-1)$ 번의 모듈로 곱셈과 사용된 키의 계산을 위해 생성된 키 정보와 행렬 b와의 $(\lambda+1)$번의 모듈로 곱셈이 필요하므로 총 2λ 모듈로 곱셈의 계산 오버헤드가 필요하여, 이는 pairwise 키 선분배 기법[6]과 동일한 양이다.

3. 메모리 오버헤드

본 논문에서 제공하는 키 관리 메커니즘에서 각 노드가 저장해야 할 정보에 따른 메모리 오버헤드는 표 3과 같다. 표 5에서처럼 하드웨어 리소스가 제한적인 SN에서는 사용하는 키에 관한 최소한의 값만을 저장하고, 이보다 리소스가 더 많은 AN에서는 자신의 영역에 있는 SN에서 사용하는 키만을 계산할 수 있는 최소의 정보를 저장하고 있으며, 가장 많은 리소스를 갖고 있는 SINK에서는 전체 네트워크에서 사용될 수 있는 키를 계산할 수 있는 정보와 키 갱신을 위해 사용될 키 정보만을 저장하고 있다. 특히 다른 중앙 관리와는 달리 SINK에서는 키와 노드와의 관계를 전혀 저장하지 않기 때문에 보안상 유리한 점을 제공할 뿐 아니라, 메모리의 오버헤드도 그만큼 줄일 수 있고, 노드와 키 사이에 매핑관계가 없으므로 노드 수에 따라 메모리 오버헤드가 정비례하지 않는다는 장점을 제공한다.

이에 비해 pairwise 키 선분배 기법[6]에서 각 노드는 키를 만들기 위한 비밀정보 τ개를 갖는데, 즉 각 비밀정보는 행렬 $A(=(DG)^T)$의 한 행(사이즈 $(\lambda+1)$)이 되고 각 비밀정보에 대한 키 스페이스 번호와 G 행렬의 시드도 저장해야 하므로, 전체 노드의 수가 N이고 행렬 A의 각 요소의 크기가 α인 경우, 필요한 메모리양의 총 합은 $((((\lambda+1)\alpha+\log_2\omega)\tau)+\alpha)N$이 된다. 이는 N에 비례하는 양으로써, 그림 10, 그림 11은 본 논문에서 제안한 메커니즘과 메모리 오버헤드 측면에서 비교하여 도시하였다. 그림 10은 각 노드의 이웃 노드 수(d, density)를 50으로 고정한 경우, 네트워크 사이즈(N)를 증가시켰을 때 [6] 스킴과 비교한 그림이고, 그림 11은 N이 5000인 경우, 각 노드의 이웃 노드 수(d)를 증가시켰을 때 비교한 그림이다. 그림 10, 그림 11에서 사용한 파라미터 값은 각각 표 4, 표 5와 같다.

표 3. 제안하는 키 관리에서 메모리 오버헤드
Table 3. Memory overhead of proposed key management.

노드	메모리 오버헤드	참고
SN	$\tau(\log_2\omega+2\alpha)$ (ω: 전체 키 스페이스 수, α: 시드 값 혹은 키 길이, τ: 각 SN이 갖고 있는 키셋 수)	τ개의 키셋 저장
AN	$\mu\alpha(\lambda+1)$ (μ:각 AN이 갖고 있는 행렬 b의 수, α:행렬b의 한 원소 길이)	μ개의 행렬 b저장
SINK	$\omega\alpha(\lambda+1)+n\alpha(\lambda+1)+2n\alpha$ $=\alpha((\omega+n)(\lambda+1)+2n)$ (α: 행렬 X,y,y_k,b의 한 원소 길이, n: X,y,y_k행렬의 크기)	ω개의 행렬 b 및 *D-Key Space*에 대한 행렬 X,y,y_k저장

그림 10에서는 노드의 수를 2000, 5000, 8000으로 늘이되, SINK의 수는 하나로, AN의 수는 각 이웃 노드의 수 d를 50으로 고정시켰으므로, 한 AN이 SN 50개를 처리한다고 가정하여 수를 결정하였다. [6]에서 사용되는 ω와 τ값은 두 노드가 적어도 하나의 키를 공유할 확률 계산에 사용되었던 수식 1([6]에서 수식 4)을 이용하여 τ 값을 4로 하고, P_c(그래프가 연결될 확률)을 0.9999로 하여(PIKE[22]에서 실험에 사용된 값) ω값을 결정하였다. 키의 길이나 키 정보에 해당하는 행렬([6]의 행렬 A, G, 본 논문에서의 행렬 X,y,y_k,b)의 각 원소의 길이 α를 64 비트로 하였으며, 키 정보에 해당하는 행렬의 크기를 결정하는 λ는 수식 2([6]에서 수식 5,11)에 의해 결정하였다. 즉, 공격자가 x개의 노드보다 더 많은 노드를 공격하면 관리되는 키에 대한 정보가 급격하게 노출되는데, 이러한 노출에 대한 한계점을 결정하는

표 4. 그림 10을 위해 사용한 파라미터
Table 4. Used parameters for figure 10.

N 또는 n	2000	5000	8000
SINK	1	1	1
AN	40	100	160
SN	1959	4899	7839
ω 또는 μ	42	40	38
τ	4	4	4
α	64	64	64
λ	9	25	42

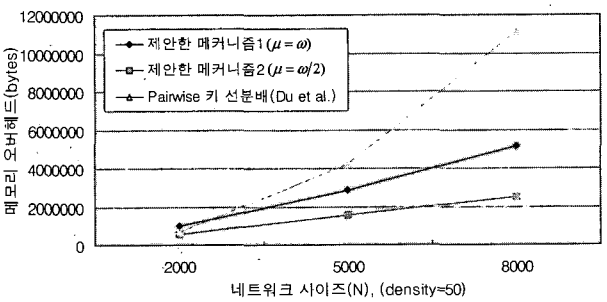


그림 10. 네트워크 사이즈(N)에 따른 메모리 오버헤드
Fig. 10. Network size(N) vs. Memory overhead.

$P_{compromise}$ 확률 값을 5%(PIKE[22]에서 실험에 사용된 값)로 하고, 한 노드가 갖는 메모리양을 나타내는 m값에 따라 λ 값을 결정하였다. 또한, 본 논문의 AN에서 갖는 키 스페이스 수 μ 값을 전체 키 스페이스 수인 ω와 동일하게 한 경우가 그림 10에서 "제안한 메커니즘1"이고(메모리 측면에서 최악의 경우), ω의 절반의 값을 μ의 값으로 한 경우가 "제안한 메커니즘2"의 결과이다. 마지막으로, 본 논문의 행렬 X, y, y_k의 크기를 나타내는 n값은 전체 노드 수 N과 동일하게 하였으나, 실제로는 더 작은 값을 사용하여도 키 관리가 가능하다. 결과적으로 [6]의 메모리 오버헤드는 네트워크 사이즈에 비례하여 증가하는 반면, 제안한 메커니즘에서는 네트워크 사이즈 증가에 덜 영향을 받을 뿐 아니라, AN이 SN보다 메시지 송수신 거리가 더 큰 경우, AN의 수가 감소할 수 있어 메모리 오버헤드가 줄어들 수 있고, 또한 "제안한 메커니즘2"에서처럼 AN이 갖고 있는 키 스페이스 수가 적은 경우 메모리 오버헤드는 더 줄어들 수 있다.

$$P_{actual} \geq P_{required},$$
$$1-\frac{((\omega-\tau)!)^2}{(\omega-2\tau)!\omega!} \geq \frac{(N-1)}{dN}[\ln(N)-\ln(-\ln(P_c))] \quad (1)$$

표 5. 그림 11을 위해 사용한 파라미터
Table 5. Used parameters for figure 11.

d	20	50	80
N 또는 n	5000	5000	5000
SINK	1	1	1
AN	250	100	63
SN	4749	4899	4936
ω 또는 μ	11	40	67
τ	4	4	4
α	64	64	64
λ	90	25	14

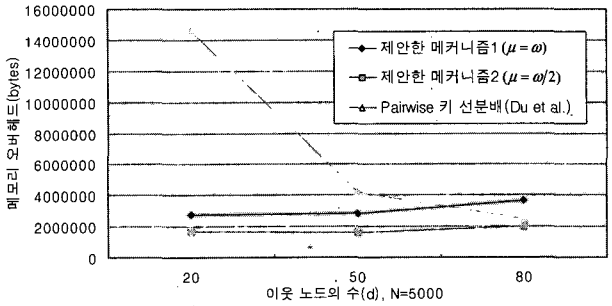


그림 11. 이웃 노드의 수(d)에 따른 메모리 오버헤드
Fig. 11. Density(d) vs. Memory overhead.

$$x < \frac{m\omega}{\tau^2},\ m = \lambda\tau(\lambda+1 \approx \lambda),$$
$$P_{compromise}N < \frac{\lambda\tau\omega}{\tau^2} = \frac{\lambda\omega}{\tau} \quad (2)$$

그림 11은 각 노드의 이웃 노드 수(d)를 증가시켰을 경우 비교한 그림인데, d 값에 따라 한 AN이 처리하는 SN의 수를 정하여 AN의 수를 결정하였으며, 나머지 파라미터 값은 표 6에서와 동일하게 결정하였다. 그림 11의 결과는 [6]의 경우, 밀집한 네트워크에서 두 노드가 적어도 하나의 키 스페이스를 공유할 확률이 높아지기 때문에 d 값이 증가하면 더 낮은 메모리 오버헤드를 갖는다. 이에 비해 본 제안한 메커니즘은 d 값에 따라 AN이 처리하는 SN의 수를 결정하여 AN의 총 수를 결정하였기 때문에 d 값이 증가하면 AN의 메모리 오버헤드가 증가하므로 약간의 메모리 오버헤드 증가 결과를 보이고 있다. 그러나 이 증가는 크지 않을 뿐 아니라 AN이 SN보다 메시지 송수신 거리가 더 큰 경우, AN의 수가 감소할 수 있어 메모리 오버헤드가 줄어들 수 있고, 또한 "제안한 메커니즘2"에서처럼 AN이 갖고 있는 키 스페이스 수가 적은 경우 메모리 오버헤드는 더 줄어들 수 있다.

V. 결 론

본 논문에서는 안전한 센서 네트워크를 위한 기본적인 메커니즘인 키 관리에 대해 기존에 제안되었던 메커니즘의 비교 분석을 통해 이에 대한 요구사항을 도출하고, 계층적인 이동 센서 네트워크에 특성에 맞는 분산 키 관리 메커니즘을 제안하였다. 계층적인 센서 네트워크에서 대부분의 트래픽을 구성하고 있는 이벤트 정보를 담은 상향 트래픽(SINK 노드로의 트래픽)의 암호화 및 하단 노드에 대한 인증을 위해 사용될 키를 생성하고, 사용된 키를 계산하며, 관리하는 키 스페이스의 용이한 확장 및 추가를 위해 회귀모델을 이용하였다. 또한, 노드 캡처에 따른 위협에 대해 λ-security 특성을 제공해 주며, 키 관리를 SINK 및 AN으로 분산시키고, 관리 노드에서 키 자체에 대한 정보를 저장하지 않고 사용된 키를 간단히 계산함으로써 중앙 키 관리에 대한 단점 및 위협에 따른 피해를 완화시켜 주었다. 또한 키와 노드에 대한 매핑관계를 고정시키지 않음으로써 노드의 익명성 및 불추적성을 제공하고, 주기적으로 키를 재분배함으로써 키의 신규성을 제공하는 특징을 갖고 있다.

마지막으로 제안된 메커니즘을 기존 메커니즘과 특징을 비교하고, 통신·계산·메모리 측면의 오버헤드 분석을 통해 제안된 키 관리의 효율성을 입증하였다.

참 고 문 헌

[1] S. Schmidt, H. Krahn, S. Fischer and D. Watjen, "A Security Architecture for Mobile Wireless Sensor Networks," ESAS, LNCS 3313, pp. 166-177, 2005.

[2] F. Hu and X. Cao, "Security in Wireless Actor & Sensor Networks (WASN): Towards A Hierarchical Re-Keying Desig,." International Conference on Information Technology: Coding and Computing, pp. 528-533, 2005.

[3] C. Karlof, N. Sastry and D. Wagner, "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks," 2nd ACM Conference on Embedded Networked Sensor Systems, Nov.2004.

[4] A. Perrig, R. Szewczyk, V. Wen, D. Culler and J. D. Tygar, "SPINS: Security Protocols for Sensor Networks," *Wireless Networks Journal (WINE)*, September 2002.

[5] S. Avancha, J. Undercoffer, A. Joshi and J. Pinkston, "Secure sensor networks for perimeter protection," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Vol. 43, Issue 4, pp. 421-435, November 2003.

[6] W. Du, J. Deng, Y. S. Han and P. Varshney, "A Pairwise Key Pre-distribution Scheme for Wireless Sensor Networks," 10th ACM Conference on Computer and Communications Security (CCS), October 2003.

[7] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," 10th ACM Conference on Computer and Communications Security (CCS), pp. 52-61, October 2003.

[8] J. D. Richard and S. Mishra, "Security Support for In-Network Processing in Wireless Sensor Networks," ACM Workshop on the Security of Ad hoc and Sensor Network, pp. 83-93, 2003.

[9] M. Bohge and W. Trappe, "An Authentication Framework for Hierarchical Ad-hoc Sensor Networks," ACM Workshop on Wireless Security (WiSe), 2003.

[10] Y. T. Hou, Y. Shi and H. D. Sherali, "Rate Allocation in Wireless Sensor Networks with Network Lifetime Requirement," MobiHoc, pp. 67-77, 2004.

[11] 박성현, "제3판 회귀분석," 민영사, 1998년 9월

[12] 임채훈, "유비쿼터스 센서 네트워크 보안," *한국통신학회지*, 정보통신 Vol. 22, no. 8, August 2005

[13] F. Ye, H. Luo, J. Cheng, S. Lu and L. Zhang, "A Two-tier Data Dissemination Model for Large-scale Wireless Sensor Networks," ACM/IEEE Mobicom, 2002.

[14] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann, and F. Silva. "Directed Diffusion for Wireless Sensor Networking," *ACM/IEEE Transactions on Networking*, 11(1), pp. 2-16, February 2002.

[15] J. Ding, K. Sivalingam and B. Li, "Design and Analysis of an Integrated MAC and Routing Protocol Framework for Wireless Sensor Networks," *Int. Journal on Ad Hoc & Sensor Wireless Networks*, March 2005.

[16] H. Chan, A. Perrig and D. Song, "Random Key Predistribution Schemes for Sensor Networks," 2003 IEEE Symposium on Security and Privacy, pp. 197-213, 2003.

[17] S. Schmidt, H. Krahn, S. Fischer and D. Watjen, "A Security Architecture for Mobile Wireless Sensor Networks," 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS), LNCS 3313, August 2004.

[18] G. Gaubatz, J. P. Kaps and B. Sunar, "Public

Key Cryptography in Sensor Networks-Revisited," 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS), LNCS 3313, pp. 2-18, August 2004.

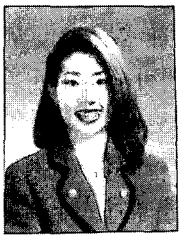
[19] D. J. Malan, M. Welsh and M. D. Smith, "A Public-Key Infrastructure for Key Distribution in TinyOS Based on Elliptic Curve Cryptography," 1st IEEE International Conference on Sensor and Ad Hoc Communications and Networks (SECON), October 2004.

[20] X. Chen and J. Drissi, "An Efficient Key Management Scheme in Hierarchical Sensor Networks," MASS 2005.

[21] M. Tubaishat, J. Yin, B. Panja and S. Madria, "A Secure Hierarchical Model for Sensor Networ," SIGMOD Record, Vol. 33, No. 1, pp. 7-13, March 2004.

[22] H. Chan and A. Perrig, "PIKE: Peer Intermediaries for Key Establishment," IEEE INFOCOM, March 2005.

저 자 소 개



김 미 희(정회원)

1997년 이화여자대학교 전자계산학과 학사 졸업.

1999년 이화여자대학교 컴퓨터학과 석사 졸업.

1999년 (주)인티 연구원.

1999년~2003년 한국전자통신연구원 네트워크연구소 연구원.

2003년~현재 이화여자대학교 컴퓨터학과 박사과정 재학중.

<주관심분야 : 센서 네트워크 보안, 이동 네트워크 보안, 통합망 보안>

채 기 준(정회원)

1982년 연세대학교 수학과 학사 졸업.

1984년 미국 Syracuse University 컴퓨터학과 석사 졸업.

1990년 미국 North Carolina State University 컴퓨터공학과 박사 졸업.

1990년~1992년 미국 해군사관학교 컴퓨터학과 조교수

1992년~현재 이화여자대학교 컴퓨터학과 교수

<주관심분야 : 네트워크 보안, 액티브 네트워크 보안 및 관리, 인터넷/무선통신망/고속통신망 프로토콜 설계 및 성능분석>