논문 2007-44TC-11-10

오염에 취약한 센서노드들을 위한 주기적인 키갱신 방안

(A Periodical Key Refreshment Scheme for Compromise-prone Sensor Nodes)

왕 기 철*, 김 기 영**, 박 원 주**, 조 기 환***

(Gi Cheol Wang, Ki Young Kim, Won Ju Park, and Gi Hwan Cho)

요 약

센서 네트워크에서 센서 들의 통신키를 주기적으로 혹은 필요에 의해 갱신하는 것은 매우 중요한 문제이다. 동적인 키관리의 효율성을 기하기 위해, 센서 네트워크는 클러스터 구조를 도입하고 각 CH(Cluster Head)가 클러스터 내의 키를 관리하도록 한다. 그러나, 이러한 클러스터 기반 센서 네트워크에서 CH는 공격의 목표가 되기 쉽고, CH들의 오염은 네트워크 전체에 큰 위협을 가져온다. 본 논문에서, 우리는 CH들의 오염에 강건한 주기적인 키 갱신 방안을 제안하였다. 먼저, 제안방법은 CH가 관리하는 센서들의 수를 줄이고 CH들이 주기적으로 변경되게 함으로써, CH의 오염에 따른 영향을 최소화 한다. 둘째, 제안방법은 임의의 CH와 BS(Base Station)간의 키 설정에 다른 센서노드들을 참가시켜 공격자들을 혼란에 빠뜨린다. 우리는 수치적인 분석을 통해 제안방법이 다른 키 관리 방법들에 비해 안전하고 CH들의 오염에 강건함을 증명하였다.

Abstract

In sensor networks, it is very important to refresh communication keys of sensors in a periodic or on-demand manner. To perform a dynamic key management efficiently, sensor networks usually employ cluster architecture and each CH (Cluster Head) is responsible for key management within its cluster. In cluster-based sensor networks, CHs are likely to be targets of capture attacks, and capture of CHs threatens the survival of network significantly. In this paper, we propose a periodical key refreshment scheme which counteracts against capture of CHs. First, the proposed scheme reduces the threat caused by compromise of CHs by forcing each CH to manage a small number of sensors and changing CH role nodes periodically. Second, the proposed scheme flings attackers into confusion by involving other nodes in a key establishment between BS (Base Station) and a CH. Our numerical analyses showed that the proposed scheme is more secure than other schemes and robust against compromise of CHs.

Keywords : Periodical Key Refreshment, WSN(Wireless Sensor Network), Cluster Formation

I. 서 론

센서 네트워크는 일반적으로 많은 수의 극히 작은 센서로 구성되는 네트워크이다. 이러한 네트워크에서 각 센서 노드는 제한된 전원을 가지고 동작하며 다양한 센서들을 통해 데이터를 감지하고 이를 수집하여 사용자에게 전달한다. 따라서 센서 네트워크는 군사목적의 탐지 및 추적, 환경감시, 환자감시 및 추적 등과 같은 다양한 분야에 활용될 것으로 보인다[1].

센서 네트워크가 널리 사용되기 위해서는 네트워크 상의 각 노드들에게 안전한 통신을 보장할 수 있어야 한다. 이는 일반적으로 센서 노드들이 무방비 상태의

* 정회원, 전북대학교 영상정보신기술 연구소
(Center for Advanced Image and Information Technology, Chonbuk National University)
** 정회원, 한국전자통신연구원 정보보호연구단 임베디드보안기술연구팀
(Division of Information Security Research, ETRI)
*** 정회원, 전북대학교 전자정보공학부
(Division of Electronics and Information Engineering)
※ 본 연구는 정보통신부 및 전보통신연구진흥원의 IT 신성장 동력 핵심기술 개발사업의 일환으로 수행하였음. [2007-S-023-01, 복합단말용 침해방지 기술]
접수일자: 2007년8월14일, 수정완료일: 2007년11월16일

환경이나 적대적인 환경에 배치되기 때문이다. 즉, 이러한 적대적인 환경하에서 임의의 공격자는 손쉽게 데이터 트래픽을 엿듣거나 수정할 수 있으며, 정당한 사용자인 것처럼 위장할 수 있다. 그러므로 센서노드들 간의 통신은 기밀성과 인증을 보장할 수 있어야 한다. 즉, 센서 네트워크에서 노드들 간에 통신키를 설정하는 것은 매우 중요하다[2~5].

센서 네트워크에서 통신키를 설정하기 위한 기존의 연구들은 대부분 정적인 키관리[1~4, 6~7]에 집중되어 왔다. 정적인 키관리에서는, 임의의 키 서버가 여러 개의 키를 가진 키풀에서 특정한 개수만큼 키들을 랜덤하게 뽑아 각 센서에게 미리 분배한다. 센서들이 임무현장에 배치된 후에, 각 센서들은 미리 분배받은 키들을 이용하여, 이웃 센서들과 통신키를 설정한다. 그러나 이러한 키들은 네트워크가 소멸될 때까지 변경되지 않으므로, 정적인 키들이라고 불린다. 따라서 공격자들에 의해 포획된 센서들의 키는 네트워크 소멸 시까지 유효하게 된다. 반면에, 동적인 키관리에서는 센서들이 가지는 키들이 주기적으로 혹은 필요에 의해서 변경된다[8~13]. 이러한 동적인 키관리는 공격자의 부담을 가중시키는 효과를 발휘하므로, 정적인 키관리에 비해 바람직하다. 기존의 동적인 키관리 기법들은 하나의 네트워크를 여러 개의 그룹(클러스터)로 분할하고 각 클러스터 헤드(CH:Cluster Head)가 키의 분배 및 관리를 담당하고 있다. 이는 하나의 중앙화된 서버가 네트워크 전체를 담당하는 부하를 줄이고 키관리의 효율성을 높이는 효과를 유발한다.

클러스터 기반의 동적인 키관리에서 문제점은 CH들이 집중공격의 대상이 된다는 것이다. 즉, 공격자들은 데이터의 집약점인 CH를 쉽게 파악할 수 있고, 이들을 포획하기 위해 힘을 집중할 것이다. 이는 CH들을 포획함으로써 얻는 이점이 일반센서들의 그것 보다 훨씬 크기 때문이다. 즉, CH에서 BS(Base Station)으로 전송되는 데이터는 그 클러스터 내에서 수집된 모든 데이터가 들어있다. 만일 모든 CH가 포획되면, 네트워크의 모든 정보가 공격자들에게 노출 된다. 그러나 기존의 기법들은 센서들의 포획에 따르는 위험을 최소화 시키는 데만 집중해 왔다. 즉, 그들은 오염된 센서들이 더 이상 클러스터 내의 통신에 참여하지 못하도록 방지하는 데에 초점을 맞추었다.

본 논문에서, 우리는 CH의 오염에 따르는 공격자의 이점을 최소화하는 주기적인 키 갱신기법을 제안한다. 제안하는 방법에서, CH들은 기존의 방법들 보다 훨씬 작은 수의 센서들만을 관리한다. 따라서 공격자의 CH 포획으로 인한 이점은 크게 감소된다. 또한 제안하는 방법에서 CH들은 주기적으로 변경되므로, 공격자들의 CH 포획으로 인한 정보획득은 시간에 따라 제한된다. 제안된 방법에서, CH와 BS간의 통신키는 여러 노드들의 부분키 전달에 의해 설정되므로, 공격자들에게 많은 부담을 유발한다.

본 논문의 구성은 다음과 같다. Ⅱ장에서 우리는 기존의 기법들에 대해 간략히 살펴본다. Ⅲ장에서는 본 논문에서 사용하는 네트워크의 시스템 모델을 정립한다. Ⅳ장에서는 제안된 방법에 대해 자세히 기술한다. Ⅴ장은 제안된 방법의 안전성과 효율성을 분석한다. Ⅵ장은 본 논문의 결론을 내린다.

Ⅱ. 관련연구

Eschnauer등은 최초로 센서 네트워크를 위한 키 선행 분배 방법을 제안하였다[6]. 이 방법에서, 각노드는 네트워크 배치 이전에 임의의 키풀로부터 정해진 수만큼의 키를 분배 받는다. 네트워크 배치 후에, 각노드는 이웃노드들과 공유된 키들을 이용하여 통신키들을 설정한다. 만일, 공유된 키가 존재하지 않는 경우에는, 공유된 키를 가지는 이웃노드들을 이용하여 간접적으로 통신키를 설정한다. Liu등은 선행 비밀 정보 분배 기반의 통신 키 설정방법에 관한 프레임워크를 제안하였다[14]. 이 방법에서는 키 setup서버가 이변수 t차 다항식들을 담고 있는 다항식 pool에서 임의의 개수 만큼의 다항식들을 랜덤하게 뽑고 그 다항식들의 부분키들을 각 노드들에게 분배한다. 각 노드는 이 부분키들을 이용하여 같은 부분키들을 가지는 노드와 직접 키를 설정한다. 만일 임의의 두 노드가 같은 부분키들을 가지지 못하는 경우에는, 두 노드와 공유된 부분키들을 가지거나 직접 키를 설정한 노드들을 검색한다. 이후에 검색된 노드들을 이용하여 간접적으로 키를 설정한다.

Eschnauer방법의 문제점은 두 노드간에 공유된 키의 수가 1인 경우에도, 통신키 설정이 가능하다는 것이다. Chan등은 이러한 문제점을 해결하기 위해 통신키 설정이 가능한 공유키의 수를 q개 이상이 되도록 한 방법이다[1]. Du 등은 센서노드들이 임무현장에 배치될 대략의 위치를 안다면, 각 센서에게 분배되는 키의 수를 크게 감소시킬 수 있음을 보였다[3]. 즉, 키 선행 분배시간에 키 분배서버는 임의의 영역의 센서들이 인접영역의 센서들과 다수의 키들을 공유하도록 조정한다. 따라서, 임

의의 두 센서는 분배 받는 키의 수가 작다 하더라도, 대부분의 이웃노드들과 통신키를 설정할 수 있다. Liu등은 노드들이 그룹으로 배치되는 환경에서의 키 선행 분배 기법을 제안하였다[4]. 이 방법에서, 동일 그룹에 속하는 노드들은 그들간의 거리가 매우 가까우므로, 공통의 키를 가질 확률이 매우 높도록 키를 분배받는다. 만일, 인접한 노드들이 동일한 그룹에 속하지 않는 경우에는 그 인접한 노드들이 다중 그룹들에 속하는 노드들간의 간접키 설정을 용이하게 하기 위한 그룹간 게이트웨이들로 동작한다. Gu등은 2홉 전송범위 내에서 노드들간의 키 공유관계를 표현하는 논리적 그래프를 이용하여 이웃노드간의 통신키 설정 비율을 높이는 방법을 제안하였다[7]. Traynor등은 다른 능력을 가진 센서노드들로 구성된 네트워크에서의 키 선행 분배방법을 제안하였다[2]. 이 방법에서, 센서들은 높은 능력을 가진 센서들과 낮은 능력의 센서들로 구분되며, 높은 능력의 센서들은 보다 많은 키들을 분배 받는다. 또한, 이 방법은 높은 능력의 센서들이 낮은 능력의 센서들간의 통신을 도와주는 계층형의 통신모델을 적용한다. 결과적으로, 이 방법은 낮은 능력의 센서들에게 작은 수의 키들만을 분배하여 저장공간을 절약시키고 노드 오염에 따르는 영향을 최소화 시킨다.

Eltoweissy등은 임의의 통신그룹내에서 그룹키를 제거된(evicted) 노드들로부터 보호하기 위한 EBS (Exclusion Basis System)시스템을 제안하였다[10]. EBS에서 각 멤버가 k+m개의 관리키들 중에서 k개씩을 분배받는다. 만일 그룹의 멤버가 제거(evicted)되면, 단지 m개의 메시지만으로 제거된 노드가 알 수 없는 기존의 관리키들을 수정할 수 있다. 따라서 제거된 노드들은 원래의 그룹내의 통신에 참여할 수 없게 된다. 참고문헌 [11]은 무선 센서 네트워크에 EBS기반의 그룹키 관리 방법을 적용한 기법을 제안하였다. 이 기법에서, 각 센서들은 BS이 방송하는 훈련내용에 의해 자신이 속하는 클러스터를 결정한다. 각 클러스터 마다 하나의 그룹키를 가지며, 이 그룹키를 관리하기 위해 전체 네트워크에 EBS시스템을 적용한다. Jolly등은 센서 네트워크에서 EBS에 기반하지 않은 동적인 키관리 기법을 제안하였다[9]. 이 기법에서, BS은 키들을 생성하고 이들을 미리 노드들에게 분배하고, 게이트웨이들은 이웃 게이트웨이들과의 통신을 통해서 공유되지 않은 키들을 분배받는다. 따라서 이 방법은 센서들의 기억공간을 크게 절약시킨다. 반면에, rekeying은 클러스터들의 재구성과 키들의 재분배를 유발하므로, 통신 오버헤드가 크다.

Younis등은 Eltoweissy가 제안한 EBS기법이 노드들간의 협력공격(collusion)에 취약함을 지적하였다[13]. 노드들의 협력공격을 해결하기 위해, Younis등은 SHELL (Scalable, Hierarchical, Efficient, Location-aware, and Lightweight)기법을 제안하였다. SHELL은 인접 노드을 포획하는 것에 의해 얻어지는 위협을 최소화하기 위해 클러스터 내에서 위치에 기반한 키 배정을 수행한다. 즉, SHELL 에서 인접한 노드들은 그렇지 않은 노드들에 비해 많은 키들을 공유한다. SHELL은 Jolly의 방법처럼 중앙화된 서버가 rekeying을 수행한다. Eltoweissy는 클러스터와 센서간은 물론 BS과 클러스터간에도 EBS를 적용시키는 LOCK(LOcalized Combinatorial Keying)을 제안하였다[8].

III. 네트워크 및 위협 모델

1. 네트워크 모델

본 논문에서의 네트워크는 하나의 BS과 몇몇의 CH들, 그리고 CH들의 지배하에 있는 센서들로 구성된다. 센서들은 모두 고정되어 있으며 모두 CH의 역할을 수행할 수 있다. 즉, 시간에 따라 CH역할을 수행하는 노드들은 변경된다. 일반센서는 오직 하나의 CH에만 속하며 자신이 감지한 정보를 CH노드에게 전송한다. CH센서는 일반센서로부터 수신한 정보를 집약하며 집약된 데이터를 BS에게 전송한다. 센서노드들은 모두 고정되어 있으며, 정확히 하나의 CH에 의해 지배를 받는다. 네트워크의 수명을 증가시키기 위해, 각 일반센서는 자신의 전송이 허용된 시간에만 데이터를 전송하고 나머지 시간에는 sleep상태로 존재한다. 이를 위해 각 CH는 센서와 직접통신을 수행하며 주기적으로 그 클러스터의 TDMA스케쥴을 방송한다. BS은 많은 양의 가용자원을 보유하고 있으며, 안전한 곳에 위치하여 공격들에 대해 자유롭다. 반면에, 센서들은 가용자원이 빈약하고 언제든지 공격을 받을 수 있는 환경에 배치되어 있다. 따라서, 보호되지 않는 환경에서 동작하는 센서들로부터의 정보를 보호하는 것이 필요하다.

2. 위협모델

사실, 센서 네트워크에서 가장 강력한 공격은 DoS (Denial of Service)공격이다. 예를 들어, 공격자는 임의의 강력한 방해전파를 통해 센서들의 통신을 방해할 수 있다. 또한, 공격자들은 대량의 데이터를 지속적으로 전송하여 CH와 주변센서들의 정당한 정보전송을 방해할

수 있다. 그러나 이러한 형태의 공격을 완벽하게 퇴치할 수 있는 해결책은 아직까지는 없는 것으로 알려져 있다.

위의 공격을 제외하면, 공격자들은 물리적으로 취약한 센서노드들을 포획함으로써 많은 이득을 얻는다. 이는 포획된 센서 내에 저장된 정보 및 키들이 모두 공격자에게 노출되기 때문이다. 특히, CH의 포획은 그 클러스터내의 모든 센싱정보를 노출시키므로 센서노드들의 포획에 비해 더 위협적이다. CH들은 일반센서들과 마찬가지로 보호되지 않은 환경에 배치된다. 또한, CH들은 센싱 데이터들이 모이는 집중점이므로 공격자들은 손쉽게 CH노드들을 파악할 수 있다. 따라서 공격자들은 센서들보다도 오히려 CH들을 포획타깃으로 정할 것이다. 이는 작은 수의 CH들을 포획함으로써 전체 네트워크를 그들의 수중하에 놓을 수 있기 때문이다.

본 논문에서 가정하는 공격자들은 다른 수준의 능력을 가진다. 높은 수준의 공격자들은 트래픽 분석을 통해 CH역할을 수행하는 센서를 식별하고, CH들을 포획한다. 반면에, 낮은 수준의 공격자들은 일반센서들을 포획하여 그들의 키들과 정보를 획득한다.

또한, 공격자들의 목적은 시스템에서 사용되는 모든 키들을 획득하여 시스템을 지배하는 것이다. 여기서, 지배란 공격자가 네트워크내의 모든 트래픽의 내용을 획득할 수 있음을 의미한다. 이를 위해, 공격자들은 획득한 키들을 다른 공격자들과 빠르고 손쉽게 공유한다.

본 논문에서, 우리는 이러한 노드 포획에 따르는 공격자의 이익을 최소화하기 위해 다음의 방법들을 이용한다. 먼저, 우리는 CH와 BS간의 키설정을 위해 임의의 비밀키 공유기법을 이용한다. 두 번째, 그 비밀키 공유기법에서 전송되는 부분키들을 보호하기 위해, 우리는 비중첩성을 가지는 키들을 이용한다. 세 번째, CH는 공격타깃이 되므로, 우리는 주기적으로 CH역할 노드들을 변경한다. 마지막으로, CH포획에 따르는 위협을 최소화 하기 위해 하나의 CH가 관리하는 노드들의 수는 낮은 전송범위를 통해 제한된다.

Ⅳ. 주기적인 키 변경 방안

임의의 비밀키 공유기법에서, 비밀키는 부분키 전달자들의 수만큼 분할되고, 각각의 부분키는 대응하는 전달자들에게 건네진다. 부분키들의 전달자들은 각각의 부분키를 combiner (제안방법에서, combiner는 BS와 통신키 설정을 원하는 CH를 의미한다)와 미리 공유된 키로 암호화한 뒤에 combiner에게 전송한다. 따라서 combiner와 부분키 전달자들간에 미리 임의의 공유키들을 설정하여야 한다. 노들간에 공유키를 설정하는 방법은 크게 두가지로 나뉜다. 하나는 중첩 키 설정 방법[1~4, 6~7]이고, 다른 하나는 비중첩 키 설정 방법이다[5, 18]. 중첩키 설정방법들의 가장 심각한 문제점은 포획된 노드들의 증가에 따라 통신키의 안전성이 크게 훼손된다는 것이다. 이는 임의의 키가 다른 여러 개의 센서들 내에도 존재하기 때문이다. 따라서 우리는 CH와 BS간의 안전한 키설정을 위해 후자를 선택한다. Ⅳ장 1절에서 우리는 노드들간의 비중첩 키 설정에 대해 살펴본다. 비중첩 키 설정이 완료되면, 각 노드는 자신의 식별자 및 에너지 잔량을 이웃노드들과 교환한다. 즉, 그 값들의 비교를 통해, 각 노드는 자신이 CH가 될지를 결정한다. 일반적으로, 에너지 잔량이 많은 노드가 CH가 되고, 식별자는 tie breaker역할을 한다. Ⅳ장 2절은 네트워크의 클러스터구성에 관한 절차를 기술한다. 클러스터 구성이 완료되면, 각 CH는 두 종류의 통신키들을 생성한다. 하나는 자신과 BS과의 통신키이고, 다른 하나는 자신과 멤버센서들과의 통신키들이다. Ⅳ장 3절에서, 우리는 이 두 종류의 키 설정에 관해 자세히 기술한다.

1. 노드들간의 비중첩키 설정

네트워크 구성시간에 각 노드는 자신의 주변에 있는 노드들과 비중첩 키들을 설정한다. 나중에, 이 키들은 BS와 CH간의 통신키 생성을 위한 부분키들을 안전하게 전달하는 데 이용된다. 여기서, 비중첩 키 설정의 범위를 정할 필요가 발생한다. 만일, 비중첩 키 설정의 범위가 크다면, 비밀정보를 전달할 수 있는 노드의 수가 증가하게 된다. 즉, 이를 통해 통신키의 안전성을 향상시킬 수 있다. 그러나 이는 비중첩 키 설정 동안의 통신 오버헤드를 크게 증가시키게 될 것이다.

제안방법에서, CH는 미리 설정된 범위내에 있는 노드들만 관리한다. 이는 이전의 클러스터 기반 방법들[8~9, 13]에 비해 많은 수의 클러스터를 생성하게 된다. 또한 멤버들은 오직 하나의 클러스터에만 가입하므로, CH들간의 거리는 최대 3홉이 된다. 따라서 비중첩 키 설정의 범위가 3이면, 우리는 이웃 CH들을 부분키 전달에 참여시킬 수 있다. 이웃 CH들이 다른 CH들의 부분키 전송에 연루되므로, 공격자들은 많은 CH들을 오염시켜야 되는 부담을 떠안게 된다. 이런 이유로, 각 노드는 네트워크 구성 시간에 자신의 3홉 이내에 있는 노드들

과 비중첩 키를 설정한다. 다음은 본 논문에서 사용되는 표기법들이다.

- K_I : 네트워크 전역 키
- F_K : 키 K를 이용하는 의사 랜덤 함수
- $\{M\}_K$: 메시지 M을 키 K를 사용하여 암호화
- $MAC(K,M)$: 키 K를 사용한 메시지 M의 메시지 인증 코드
- $nonce_A$: 노드 A에 의해 생성된 랜덤 수
- K_A : 노드 A의 마스터 키
- K_{AB} : 노드 A와 B의 비중첩 키

먼저, 각 노드는 자신의 이웃 노드들과 비중첩 키들을 설정한다. 이 비중첩 키는 1홉 이웃사이에서 설정되므로, 1홉 비중첩 키라 칭하기로 한다. 그림 1은 노드 1과 7이 비중첩 키를 설정하는 예를 보여준다. 노드 1과 7은 각각 nonce값과 식별자를 담고 있는 Hello 메시지를 방송한다. 각 노드는 식별자가 더 큰 노드로부터의 Hello메시지는 무시한다. 노드 7은 노드 1의 식별자가 자신의 것보다 작으므로, 네트워크 전역키를 이용하여 자신의 마스터 키($K_7 = F_{K_I}(7)$)를 생성하고 이 키와 노드 1의 식별자를 이용하여 비중첩 키를 생성한다 ($K_{17} = F_{K_7}(1)$). 이후에 노드 7은 K_7을 이용하여 노드 1의 nonce에 대한 응답을 전송한다. 이후 노드 1은 노드 7의 마스터 키($K_7 = F_{K_I}(7)$)를 생성하고, 노드 7의 응답을 검증한 후에 노드 7과의 비중첩 키 ($K_{17} = F_{K_7}(1)$)를 생성한다. 같은 방법으로 각 노드는 모든 이웃노드들과 비중첩 키들을 설정한다.

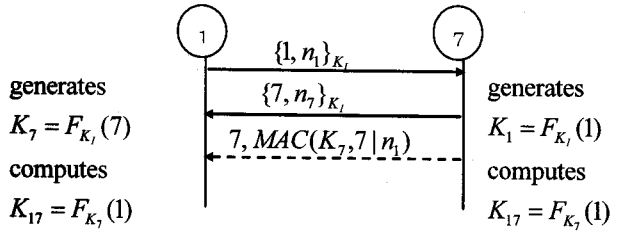


그림 1. 1홉 비 중첩키 설정 과정
Fig. 1. Establishment of one hop non-overlap keys.

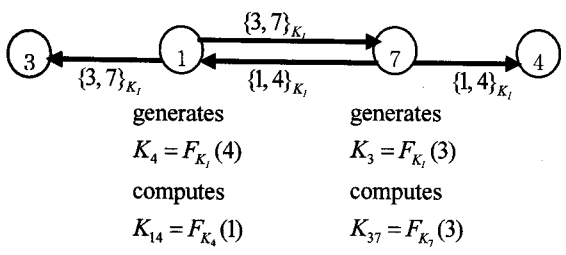


그림 2. 2홉 비중첩키 설정 과정
Fig. 2. Establishment of two hop non-overlap keys.

1홉 비중첩 키를 설정한 후에, 각 노드는 2홉 거리에 있는 노드들과의 비중첩 키 설정을 시작한다. 그림 2에서, 노드 1과 7은 각각 2홉 비중첩 키를 설정한다. 이때, 노드 1은 노드 3, 7과 이미 1홉 비중첩 키를 설정했다. 또한, 노드 7은 노드 1, 4와 이미 1홉 비중첩 키를 설정했다. 노드 1은 노드 3과 7의 식별자들을 네트워크 전역키로 암호화 한 후에 방송한다. 이 메시지를 수신한 노드 7은 노드 3에 대한 마스터 키(즉, $K_3 = F_{K_I}(3)$)를 생성한다. 이후에 노드 3의 식별자가 자신의 것보다 더 작으므로, 노드 7은 자신의 마스터키를 이용하여 3과의 2홉 비중첩 키($K_{37} = F_{K_7}(3)$)를 계산한다. 한편, 노드 1은 노드 7로부터 노드 4에 대한 식별자를 수신하므로, 노드 4에 대한 마스터 키($K_4 = F_{K_I}(4)$)를 생성한다. 그러나 노드 4의 식별자가 자신의 것보다 더 크므로, 노드 1은 노드 4의 마스터 키를 이용하여 2홉 비중첩 키($K_{14} = F_{K_4}(1)$)를 생성한다.

2홉 비중첩 키가 설정된 후에 각 노드는 자신과 2홉 비중첩 키를 설정했던 노드들을 네트워크 전역키로 암호화 한 후에 다시 방송한다. 이후의 3홉 비중첩 키 설정과정은 2홉 비중첩 키 설정과정과 동일하다. 비중첩 키 설정이 완료되면, 클러스터 구성 및 통신키 설정이 이어진다. 비중첩 키 설정은 네트워크 구성 시간에 단 한번만 실행되는 반면에 클러스터 구성 및 통신키 설정은 네트워크가 소멸될 때 까지 일정주기에 의해 반복된다.

2. 클러스터 구성

일반적으로, 센서 네트워크에서 클러스터 구조를 채용하면, 데이터 수신률 향상, 에너지 절약, 네트워크 수명 증가와 같은 이점들을 얻을 수 있다[8, 15~17]. 이러한 이점들을 활용하기 위해, 여러 문헌들은 클러스터 기반의 센서 네트워크를 위한 키 관리 방안들을 제안하였다.

우리는 기존의 클러스터 기반 방법들과는 다른 클러스터 구조를 채용한다. 먼저, 제안방법은 하나의 CH가 관리하는 멤버센서의 수가 작다. 이는 CH가 공격자의 공격목표가 되기 쉽기 때문이다. 임의의 CH 관리내에

있는 노드들의 수를 감소는 각 센서가 클러스터 구성시에 제한된 전송범위 내에 있는 노드들과만 경합하도록 하는 것에 의해 실현될 수 있다. 둘째, 제안방법에서는 모든 센서가 CH가 될 수 있다. 만일 CH의 자격을 높은 능력 (많은 에너지, 높은 컴퓨팅 파워)의 노드로 한정한다면, 높은 능력의 노드들은 일반센서들에 앞서서 임무환경에 배치될 것이다. 이러한 이종 노드 환경에서, 만일 임의의 CH가 오염되면 다른 CH들이 오염 클러스터의 멤버들을 떠맡아야 한다. 이는 다시 CH들이 관리하는 노드의 수를 증가시켜서 보안성을 약화시킨다. 셋째, 제안방법은 주기적으로 새로운 CH들을 선정한다. 따라서 공격자가 임의의 오염된 CH를 통해 그 클러스터 내의 데이터를 취득한다 하더라도, 그 불법 데이터 획득은 다음 클러스터 구성 시간까지만 유효하다. 공격자가 불법적인 데이터 획득을 연장시키기 위해서는 가능한 많은 노드들을 지속적으로 오염시킬 필요가 있다. 매 클러스터 구성 시간마다 다음의 절차를 통해 네트워크는 새로운 클러스터 구조를 가지게 된다.

1. 각 노드는 자신의 식별자와 에너지 잔량을 담은 메시지를 방송한다. 이 메시지를 통해, 각 노드는 자신이 CH가 될 자격이 있는지를 판단한다.
   A. 먼저, 이웃노드들 중에서 에너지 잔량이 가장 많은 노드가 CH가 된다.
   B. 동일한 에너지 잔량의 조건하에서, 식별자는 tie breaker역할을 수행한다. 즉, 더 작은 식별자를 가진 노드가 CH가 된다.
2. CH 노드는 그 역할을 방송하고, 이를 수신하는 노드들은 그 CH의 멤버가 된다. 모든 센서들은 하나의 CH만 선택할 수 있다. 이 선택은 "선착자 우선 규칙"을 따른다. 즉, 임의의 센서는 첫 번째 CH 선언메시지의 전송자를 자신의 CH로 정한다. 이후에 그 센서는 다른 노드로 부터의 CH 선언메시지들은 모두 무시한다.
3. 임의의 센서가 임의의 CH에 가입한 후에, 그 센서는 이 사실을 클러스터 가입 메시지 방송을 통해 그 CH에게 알린다.
   A. 만일 우선순위를 가진 노드로부터 클러스터 가입 메시지를 수신하면, 전송자를 제외한 이웃노드들 중에서 자신의 CH자격을 판단한다. 만일, 자신의 우선순위가 가장 높으면, CH가 된다.
   B. 그렇지 않다면, 그 클러스터 가입 메시지를 통해 센서들은 자신이 가입한 CH외에 주변의 다른 CH들을 알게 된다. 센서들은 주변 CH들을 자신의 CH에게 알린다.

3. 통신키 설정

클러스터가 구성된 후에, 각 CH는 자신과 BS간의 통신에 이용되는 통신키를 설정해야 한다. 또한 각 CH는 자신과 멤버센서 들간의 통신키들도 설정해야 한다. 일반적으로, CH와 BS간의 키설정이 CH와 센서간의 키설정에 비해 훨씬 중요하다. 이는 전자가 노드들의 오염에 훨씬 민감하기 때문이다. 즉, 후자의 오염은 그 센서의 데이터를 노출시키는 반면에, 전자의 오염은 그 클러스터내의 모든 센서의 데이터를 노출시키게 된다. 전자의 안전성을 향상시키기 위해, 우리는 비밀키 공유기법과 비중첩키를 이용한다. 다음은 제안방법에서 BS과 CH간의 통신키 설정 및 CH와 센서들간의 통신키 설정 절차를 기술한 것이다.

1. 각 CH는 부분키 전달자들의 리스트를 BS에게 알린다.
   A. 만일 주변 CH들의 수가 2개 이상이면, 자신의 주변 CH들의 리스트를 BS에게 알린다.
   B. 그렇지 않으면, 자신의 이웃노드들을 BS에게 알린다.
2. BS는 전송 CH와의 통신키를 생성하고, 이를 부분키 전달자의 수만큼 분할한다. BS은 분할된 부분키를 하나씩 부분키 전달자에게 전송한다. 이때 부분키들은 BS과 전달자간의 미리 분배된 암호화키로 암호화되어 전송된다.
3. 부분키를 수신한 부분키 전달자들은 비중첩 키를 이용하여 목적 CH에게 부분키들을 전송한다.
4. 목적 CH는 수신한 부분키들을 병합하여 원래의 통신키를 복원한다. 그 CH는 자신의 멤버들을 위한 통신키들을 생성하고, 멤버들과의 비중첩키들을 이용하여 이들을 멤버들에게 전송한다.
5. 단계 2부터 4까지가 모든 CH들에 대해 반복된다.

예를 들어, 그림 3에서 보이는 것 처럼, CH 7이 BS과 통신키를 설정한다고 가정하자. 먼저, CH 7은 자신의 이웃 CH들인 12, 34, 36, 79가 부분키 전달자들임을 BS에게 알린다. BS는 CH 7과의 통신키 ck_{BS-7}을 생성한다. CH 7의 인접 CH수가 4이므로, BS은 생성된 통신키를 임의의 키분할 기법을 이용하여 4개의 부분키 (즉, ck^{12}_{BS-7}, ck^{34}_{BS-7}, ck^{36}_{BS-7}, 그리고 ck^{79}_{BS-7})로 분

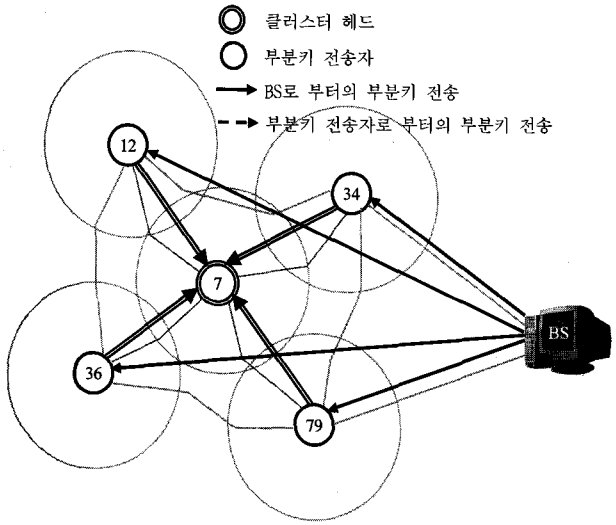


그림 3. BS와 CH간의 통신키 생성
Fig. 3. Generation of a communication key between BS and CH.

할한다. BS은 분할된 부분키들을 인접 CH 12, 34, 36, 그리고 79에게 각각 전송한다. 그들은 CH 7에게 그들의 부분키를 전송한다. CH 7은 수신한 부분키들 ck_{BS-7}^{12}, ck_{BS-7}^{34}, ck_{BS-7}^{36}, 그리고 ck_{BS-7}^{79}를 병합하여 원래의 통신키를 복원한다. 이후에 CH 7은 자신의 멤버센서들을 위한 통신키들을 생성하고 그들을 비중첩 키들을 이용하여 분배한다.

V. 분 석

본 장에서, 우리는 세 가지 다른 키 관리 방법의 안전성과 효율성에 대해 분석하고자 한다. 5.1절에서, 우리는 네트워크 내에 오염된 노드들이 다수 포함되어 있다는 가정 하에서, 오염된 노드들이 순결한 노드들의 통신에 영향을 미치는 정도를 분석한다. 5.2절에서는 키 관리 기법의 동작에 의한 통신 오버헤드를 분석한다.

1 안전성 분석

이 절에서는 네트워크 내에서 오염된 노드의 수를 n_c로 놓고, 오염노드 수에 따른 센서 및 네트워크의 통신이 노출되는 확률을 구한다. 먼저, 정적인 키관리 방법을 분석하기 위해 우리는 Chan 등이 제안한 방법[1]을 대표로 선출하였다. 동적인 키관리 방법의 대표로는 Younis등이 제안한 방법[13]이 선택되었다. 마지막으로, 우리는 본 논문에서 제안하는 주기적인 키갱신 방법을 분석하였다. 분석을 위해 우리는 먼저 실험에 사용된 파라미터들을 기술한다. 표 1은 분석에서 사용된 파라미터들과 그 의미, 그리고 값의 범위들을 기술하고 있다.

Chan의 방법에서 각 노드는 S개의 키를 가진 키풀에서 r개를 분배받으므로, 각 센서가 이웃노드와 q개의

표 1. 분석을 위한 파라미터들과 그 의미
Table 1. Parameters for analyses and their meaning.

파라미터	의미	값	적용방법
n	노드 수	100	공통
n_c	오염 노드 수	10~60	
n'	예상 이웃노드 수	22	
n_{cc}	클러스터내 오염노드 수	2~12 (n_c/c)	Younis의 방법
c	클러스터의 수	5	제안방법
c_{CH}	오염된 CH 수	2	
l_m	센서와 BS간의 최대 경로길이	20	Chan 의 방법
r	센서의 키링 크기	50	
$\|S\|$	키풀의 크기	500	
q	안전링크 설정을 위한 공유키의 수	3	
k	분배받은 관리키의 수	7	Younis의 방법
m	분배받지 않은 관리키의 수	3	
n_{ca}	오염된 관리키의 수	2	
n_{ka}	키일치 과정에 참여하는 노드의 수	3	제안방법
n_s	클러스터내의 센서노드 수	20	
s	부분키를 전송하는 노드의 수	6 ($\lceil c/2 \rceil \times 2$)	

공통키를 가질 확률 $p(q)$은 식 (1)과 같다.

$$p(q)=\frac{\binom{|S|}{q}\binom{|S|-q}{2(r-q)}\binom{2(r-q)}{r-q}}{\binom{|S|}{r}^2} \quad (1)$$

각 노드는 임의의 이웃노드와 q개 이상의 키를 공유할 경우에만 안전링크를 설정할 수 있으므로, 임의의 노드가 이웃노드와 안전링크를 설정할 확률은 $p_s=p(q)+p(q-1)+\ldots+p(r)$이 된다. 따라서, n_c개의 오염노드로 인해 임의의 두 노드 사이의 안전링크가 파괴될 확률 p_b는 식 (2)로 표현된다.

$$p_b=\sum_{i=q}^{r}\left(1-\left(1-\frac{r}{|S|}\right)^{n_c}\right)^i\frac{p(i)}{p_s} \quad (2)$$

Chan의 방법에서 각 센서는 MTE(Minimum Trans mission Energy)라우팅을 사용하여 데이터를 전달한다고 가정하자. 이때, n_c개의 오염노드에 의해 전체 네트워크가 오염될 확률은 식 (3)에 의해 결정되고, l_m은 센서와 BS간의 최대 경로길이 이다.

$$p_w=\frac{p_b}{n'}+\frac{p_b(l_m-1)}{n'-1} \quad (3)$$

Younis의 방법에서, 센싱한 데이터의 암호화에 사용되는 키는 통신키일지라도, 이 키는 관리키들에 의해 보호된다. 임의의 클러스터 내에서 각 키생성 게이트웨이는 $(k+m)/2$개씩의 관리키들을 생성하며, 전체 $k+m$개의 관리키들 중에서 k개의 키들이 랜덤하게 각 센서에게 분배된다. 만일, 임의의 센서가 오염되면 키생성 게이트웨이는 오염된 센서가 알지 못하는 키들을 이용하여 오염된 관리키들을 수정한다. 즉, 오염된 노드들의 공유키집합이 임의의 키생성 게이트웨이의 관리키들을 모두 포함하면, 오염된 노드들의 관리키들은 다시 공격자에게 노출된다. 만일 클러스터 내의 오염된 노드들의 수를 n_{cc}라 하면, 오염된 노드들의 공유키 집합이 키 생성 게이트웨이들의 관리키들을 모두 포함할 확률은 식 (4)에 의해 결정된다.

$$p_{kg}=\left(1-\left(1-\frac{k}{k+m}\right)^{n_{cc}}\right)^{k+m} \quad (4)$$

결과적으로, n_c 개의 오염노드에 의해 전체 네트워크가 오염될 확률은 식 (5)에 의해 결정된다. 이 식에서, c는 클러스터 수이고 c_{CH}는 오염된 CH의 수이다.

$$p_w=\left(p_{kg}+(1-p_{kg})\times\frac{n_c}{n}\right)^{c-c_{CH}} \quad (5)$$

제안하는 방법에서 통신키의 안전성은 노드들이 보유하는 키들 보다는 통신키 생성에 참여하는 노드들의 순결성에 의존한다. 즉, n_c개의 오염노드가 클러스터 내의 모든 센서의 통신을 획득할 확률은 식 (6)으로 표현된다. 이식에서 s는 부분키를 전송하는 노드의 수이고 n_s는 클러스터내의 센서의 수이다.

$$p_{cluster}=\left(\frac{n_c}{n}\right)^{n_s}+\frac{n_c}{n}+\left(\frac{n_c}{n}\right)^{s} \quad (6)$$

또한, n_c개의 오염노드에 의해 전체 네트워크가 오염될 확률은 식 (7)로 표현된다. 이식에서, c는 클러스터 수이고 c_{CH}는 오염된 CH의 수이다.

$$p_w=\left(\left(\frac{n_c}{n}\right)^{n_s}+\frac{n_c}{n}+\left(\frac{n_c}{n}\right)^{s}\right)^{c-c_{CH}} \quad (7)$$

그림 4는 위의 분석결과를 토대로, 표 1의 값을 대입한 결과를 보인다. 제안방법과 Younis의 방법에서는 클러스터 구조를 채용하므로, 오염된 CH의 수가 2인 경우의 결과도 나타내었다. 즉, 본 논문에서 가정하는 높은 능력의 공격자들은 최대 2개까지의 CH를 오염시킬 수 있다. 그림 4에서 보는 것처럼, Chan의 방법과 Younis의 방법모두 오염노드의 수가 증가함에 따라, 네트워크의 오염확률이 크게 증가한다. 반면에, 제안방법은 오염노드수의 증가가 네트워크 오염확률에 그다지

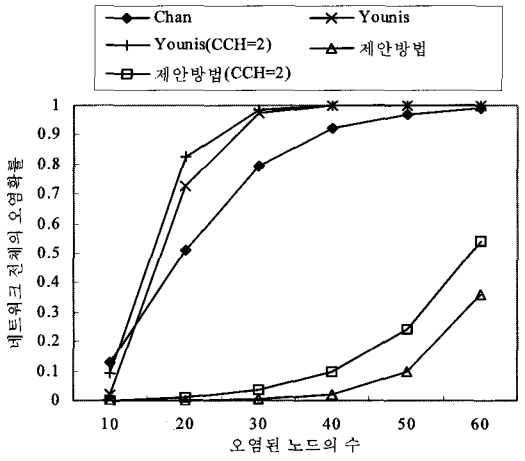


그림 4. 오염노드수에 따른 네트워크 전체의 오염 확률
Fig. 4. Compromise probability of network vs. the number of compromised nodes.

영향을 미치지 못한다. 즉, 제안방법은 다른 방법들에 비해 노드들의 오염에 매우 강건하며 네트워크의 생존 능력을 증가시킨다.

2 통신 오버헤드 분석

본 절에서, 우리는 키설정 동안 모든 센서들에 떠안게 되는 통신 오버헤드를 센서들로부터 전송되는 비트의 수를 이용하여 분석한다. 여기서, 우리는 BS의 통신 오버헤드는 고려하지 않는다. Chan의 방법에서, 임의의 두 노드간에 공유된 키가 최소 q개 이상이면 직접키를 설정할 수 있다. 그러나 공유된 키가 q개 미만이면, 두 노드는 공통의 이웃노드들을 이용한 간접키 설정을 시도할 수 있다. 그러나 간접키 설정은 공통의 이웃노드들을 찾기 위한 많은 양의 통신을 요구하므로, 본 논문에서는 배제되었다. 결과적으로, Chan의 방법에서 통신 오버헤드는 식 (8)로 표현된다.

$$co = n(\log_2 n + r \times \log_2 |S|) \tag{8}$$

Younis의 방법에서, 통신 오버헤드는 클러스터 구성, 관리키 분배, 그리고 관리키 갱신으로 구성된다. 이후에, 만일 CH가 오염된 경우는 세 단계가 모두 다시 수행되어야 한다. 반면에, 센서들만 오염된 경우에는 관리키 갱신 단계만 다시 수행되면 된다. 따라서, 네트워크가 살아있는 동안에 CH의 오염이 t_c번 발생하고, 센서노드들만 오염된 경우가 t_s번 발생한다면, 전체 통신 오버헤드는 식 (9)에 의해 계산된다.

$$yo = t_c(yo_1 + yo_2) + t_s yo_3 \tag{9}$$

이때, $yo_1 = c\left((3+n_s)\log_2 n + \left(\frac{c+1}{2}\right)\log_2 ck\right)$,

$$yo_2 = c((2+n_s)\log_2 n + 3n_s \log_2 ck + (5k+4m+2n_s)\log_2(k+m)),$$

$$yo_3 = c((n_{cc}\log_2 n + 2\log_2 ck + (6n_{ca}+2k+2m)\log_2(k+m))$$

이다. 여기서 ck는 통신키의 길이를 그리고 n_{ca}는 오염된 관리키의 수를 의미한다.

제안방법에서, 키관리를 위한 통신 오버헤드는 네트워크구성 초기의 초기키 설정, 클러스터 구성, CH와 BS간의 키설정, 그리고 센서노드와 CH간의 키설정, 으로 구성된다. 이때, 클러스터 구성부터 두 가지 키 설정은 네트워크가 살아있는 동안에 계속 반복된다는 것에 유의해야 한다. 즉, 만일 네트워크가 살아있는 동안에 클러스터 구성 및 두 가지 키설정이 t번 반복된다면 전체 통신 오버헤드는 식 (10)에 의해 계산된다.

$$po = po_1 + t(po_2 + po_3 + po_4) \tag{10}$$

이때, $po_1 = (2n + 3n')\log_2 n$,

$$po_2 = (c + cn_s)(\log_2 n + 1) + \frac{n_s c^2}{4}\log_2 n,$$

$$po_3 = n_s \log_2 ck,$$

$po_4 = 2cs \log_2 ck$ 이다.

그림 5는 위의 분석결과를 바탕으로, 표 1의 값을 대입시킨 결과를 보인다. Younis의 방법에서 CH의 오염횟수는 2 부터 3까지의 범위에 있다. 또한, 센서들의 오염횟수는 각각 30과 60으로 설정하여 결과를 추출하였다. 제안방법은 네트워크가 살아있는 동안의 클러스터 구성 및 키설정 반복횟수를 10에서 20까지 증가시켜 가면서 결과를 추출하였다.

그림 5에서 보는 것처럼, 제안방법과 Younis의 방법 모두 노드수가 증가함에 따라 통신오버헤드는 증가한다. 그림 5에서 제안 방법은 Chan의 방법과 거의 유사한 통신 오버헤드를 유발한다. 제안방법은 Younis의 방법에 비해서 노드수의 증가에 민감하게 반응한다. 이는 제안방법에서 클러스터 구성 및 키설정이 주기적으로 반복되고 모든 노드들이 그 과정에 참여하기 때문이다. 따라서, 제안방법에서 키갱신의 주기가 길면 길수록 통신 오버헤드는 감소한다. 그림 5에서 보는 것처럼, 키갱신 반복주기가 긴 경우(반복횟수 10)에는 Younis의 방법(오염 CH수: 2, 오염노드수: 60)에 비해 통신 오버헤드가 작다.

Younis의 방법은 BS이나 CH들이 그들의 관리하에 있는 CH나 멤버센서의 오염을 인식한 경우에만 클러스터 구성 및 관리키 갱신을 수행한다. 특히, 관리키 갱신의 경우는 각 CH가 자신의 클러스터 내에 오염된 센서가 있을 경우에만 수행하므로, 깨끗한 센서들만을 가지는 클러스터는 관리키 갱신을 수행하지 않는다.

사실, 이러한 반응형 키갱신 시스템이 성공적으로 동작하기 위해서는 시스템의 감시 및 보고체계가 잘 갖추어져 있어야 한다. 그러나 센서 네트워크에서의 잘 성숙된 감시체계의 존재는 비현실적인 가정이다. 만일, 이러한 시스템이 존재한다면, 제안하는 방법에도 유용하게 이용될 수 있다. 즉, BS는 현재 오염된 센서노드들을 각 CH에게 알리고, CH들은 불법적인 데이터의 수신을 차단할 수 있다. 또한, BS는 오염된 센서노드들

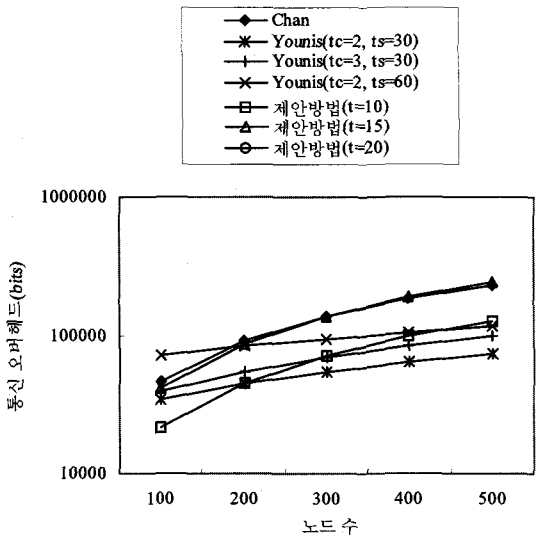


그림 5. 노드수에 따른 통신 오버헤드

Fig. 5. Communication overhead vs. the number of nodes.

및 CH들의 수에 따른 네트워크의 오염확률을 계산할 수 있다. 만일 이 확률이 미리 정해진 임계치를 넘지 않는다면, BS는 클러스터 재구성 및 키 갱신 주기를 길게 할 수 있다. 이는 센서들의 통신 오버헤드를 크게 감소시키고 따라서 네트워크의 수명이 연장된다.

그림 5에서 눈여겨 볼 또 하나는 Younis의 방법에서 CH의 오염은 센서들의 오염에 비해 많은 통신 오버헤드를 유발한다는 것이다. 이는 Younis의 방법에서 CH의 오염은 클러스터 구조의 재구성 및 관리키 갱신을 모두 수행해야 하기 때문이다.

VI. 결론 및 향후 연구

클러스터 구조를 이용하는 센서 네트워크는 센서들의 오염 보다는 CH의 오염이 네트워크에 훨씬 위협적이다. 더구나, 지능적인 공격자들은 네트워크 전체를 지배하기 위해 작은 수의 CH들을 표적으로 삼을 것이다. 그럼에도 불구하고, 기존의 연구들은 대부분 센서들의 오염에 따르는 효과를 완화하는데 집중하여 왔다. 본 논문에서는 CH들의 오염에 강건한 동적인 키 관리 방법을 제안하였고, 이 방법의 안전성 및 통신 오버헤드를 분석하였다. 분석결과를 통해, 제안방법은 다른 키 관리 방법 등에 비해 노드들의 오염은 물론 클러스터 헤드의 오염에도 매우 안전함을 보였다.

향후 연구로서 우리는 제안방법의 안전성 과 효율성을 정확히 분석하기 위해 실험환경을 구축하고, 다양한 실험을 통해 제안방법을 다른 방법들과 비교하고 평가할 계획이다.

참 고 문 헌

[1] H. Chan, A. Perrig, and D. Song, "Random Key Predistribution Schemes for Distributed Sensor Networks," in Proc. of IEEE Symp. Security and Privacy, May. 2003.

[2] P. Traynor et al., "Establishing Pair-wise Keys in Heterogeneous Sensor Networks," in Proc. of IEEE Infocom '06, 2006.

[3] W. Du et al., "A Key Management Scheme for Wireless Sensor Networks Using Deployment Knowledge," in Proc. of IEEE Infocom '04, Mar. 2004.

[4] D. Liu, P. Ning, and W. Du, "Group-Based Key Pre-Disribution in Wireless Sensor Networks," in Proc. of 2005 ACM Wksp. Wireless Security(WiSe 2005), pp. 11-20, Sep. 2005.

[5] S. Zhu, S. Setia, and S. Jajodia, "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks," in Proc. 10th ACM Conf. Computer and Comm. Security (CCS '03), Oct. 2003.

[6] L. Eschenauer and V. D. Gilgor, "A Key Management Scheme for Distributed Sensor Networks," in Proc. of 9th ACM Conf. Comp. and Comm. Sec., pp. 41-47, Nov. 2002.

[7] W. Gu et al., in Proc. of 14th IEEE Wksp on Quality of Service(IWQoS 2006), pp. 189-198, Jun. 2006.

[8] M. Eltoweissy, M. Moharrum, and R. Mukkamala, "Dynamic Key Management in Sensor Networks," IEEE Communications Magazine, vol. 44, issue 4, pp. 122-130, Apr. 2006.

[9] G. Jolly et al., "A Low-Energy Key Management Protocol for Wireless Sensor Networks," in Proc. IEEE Int'l Symp. Comp. and Comm. (ISCC '03), pp. 335-340, Jun. 2003.

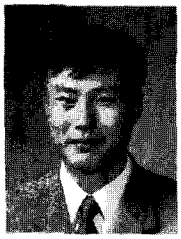
[10] M. Eltoweissy et al., "Combinatorial Optimization of Group Key Management," Journal of Network and Systems Management, vol. 12, no. 1, pp. 33-44, Mar. 2004.

[11] M. Eltoweissy et al., "Group Key Management Scheme for Large-Scale Sensor Networks," Ad Hoc Networks, vol .3, issue 5, pp. 668-688, Sep. 2005.

[12] A. Perrig et al., "SPINS: Security Protocols for

Sensor Networks," Wireless Networks, vol. 8, no. 5, pp. 521-534, Sep. 2002.
[13] M. Younis, K. Ghumman, and M. Eltoweissy, "Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks," IEEE Tans. on Parallel and Distributed Systems, vol. 17, no. 8, pp. 865-882, Aug. 2006.
[14] D. Liu and P. Ning, "Establishing Pairwise Keys in Distributed Sensor Networks," in Proc. of the 10th ACM Conference on Computer and Communications Security(CCS '03), pp. 52-61, 2003.
[15] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," IEEE Trans. Mobile Computing, vol. 3, no. 4, pp. 366-379, Oct.-Dec. 2004.
[16] W. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," IEEE Trans. on Wireless Communications, vol. 1, no. 4, pp. 660-670, Oct. 2002.
[17] G. Gupta and M. Younis, "Performance Evaluation of Load-Balanced Clustering of Wireless Sensor Networks," in Proc. Int'l Conf. Telecomm. (ICT '03), pp. 1577-1583, Mar. 2003.
[18] G. Wang and G. Cho, "Pairwise Key Establishments without Key Pre-distribution for Mobile Ad hoc Network Environment," IEE Proceedings Communications, vol. 153, no. 6, pp. 822-827, Dec. 2006.

저 자 소 개



왕 기 철(정회원)
1997년 광주대학교 전자계산학과 학사 졸업
2000년 목포대학교 전산통계학과 석사 졸업
2005년 전북대학교 컴퓨터 통계정보학과 박사 졸업
2005년 12월～현재 전북대학교 영상정보신기술연구소 연구원
<주관심분야 : Ad hoc 네트워크, 센서 네트워크, 무선네트워크 보안, 이동 컴퓨팅>

박 원 주(정회원)
1998년 충남대학교 정보통신공학과 학사 졸업
2000년 충남대학교 정보통신공학과 석사 졸업
2005년 충남대학교 정보통신공학 박사 수료
2000년 2월～현재 한국전자통신연구원 정보보호연구단 임베디드보안기술연구팀 선임연구원
<주관심분야 : 네트워크보안, 임베디드시스템보안기술>

김 기 영(정회원)
1988년 전남대학교 전산통계학과 학사 졸업
1993년 전남대학교 전산통계학과 석사 졸업
2002년 충북대학교 전자계산학과 박사 졸업
1988년 2월～현재 한국전자통신연구원 정보보호연구단 임베디드보안기술연구팀 팀장 /책임연구원
<주관심분야 : 네트워크보안, IPv6 보안 기술, 임베디드시스템 보안기술>

조 기 환(정회원)
1985년 전남대학교 계산통계학과 학사 졸업
1987년 서울대학교 계산통계학과 석사 졸업
1996년 영국 Newcastle 대학교 전산학과 박사 졸업
1987년～1997년 한국전자통신연구원 선임연구원
1997년～1999년 목포대학교 컴퓨터과학과 전임강사
1999년～현재 전북대학교 전자정보공학부 부교수
<주관심분야 : 이동컴퓨팅, 컴퓨터통신, 무선네트워크 보안, 센서네트워크, 분산처리시스템>