무선 센서 네트워크에서 안전한 클러스터링 프로토콜들의 안전성 분석*

최임성, 김진, 김광조

한국정보통신대학교

Security analysis of secure clustering protocols in wireless sensor networks

Im-sung Choi, Zeen Kim, Kwangjo Kim

Information and Communications University

요 약

무선 센서 네트워크에서 효율적으로 에너지를 사용하기 위해서 클러스터링 프로토콜들이 제안되었다. 하지만 현재까지 안전한 클러스터링에 대한 연구는 아직 미미하다. 본 논문에서는 기존의 안전한 클러스터링 프로토콜들에 대해 소개하고, 분석하였다. 또한 기존의 연구들의 문제점들을 지적하고 향후의 연구를 위하여 안전한 클러스터링을 위한 연구 방향을 제시한다.

ABSTRACT

Clustering protocols are proposed for efficient energy consumption in Wireless Sensor Networks. So far, the research of secure clustering is not activated. In this paper, We introduce existing secure clustering protocols and analyze them. We also indicate the weakness of existing researches and present research direction for future work.

Ⅰ. 서 론

최근 컴퓨터 기술과 무선 네트워크 기술의 급속한 발전으로 인해 국방 경계, 산불 감시, 환경 탐사 등 다양한 분야에서 무선 센서 네트워크가 주목받고 있다. 이런 무선 센서 네트워크는 낮은 계산 능력, 한정된 배터리 용량 등 제한된 자원을 가진 센서 노드들로 구성된다. 특히 센서 노드들은 배터리를 교환하거나, 충전하는 것이 용이하지 않기 때문에, 센서 노드들의 에너지를 효율적으로 사용하여 네트워크의 수명을 늘리는 것은 중요한 문제이다.

* 본 과제는 국가보안기술연구소의 연구과제(생존성을 보장하는 안전한 클러스터링 기법 연구,08032)의 지원으로 수행하였습니다.

이런 문제를 해결하기 위해서 클러스터링 프로토콜들이 제안되고 있다. 클러스터링 프로토콜들은 일반적으로 다음과 같이 동작한다. 먼저 특정 노드들이 스스로 자신을 클러스터 헤드로 뽑고, 주변의 노드들을 자신의 멤버노드들로 모집한다. 클러스터 헤드들은 멤버노드들로부터 센싱 데이터를 전송받고, 그것들을 집합하여 베이스 스테이션에 전송한다.

이런 클러스터 기반의 무선 센서 네트워크는 클러스터링을 형성할 때 서비스 거부 (Denial of Service) 공격 [6] 에 취약점을 가진다, 클러스터 헤드가 공격당하면 클러스터의 모든 노드들이 제대로 동작할 수 없기 때문에, 기존의 센서 네트워크보다 많은 안전상의 취약점을 가진다고 볼 수 있다. 하지만 아직까지 클러스터링 프로토콜의 안전성에 대한 연구는 아직 미미하다. 본 논문에서는 현재까지 제안된 안전한 클러스터링을 위한 프로토콜들을 소개하면서 분석하고, 향후 연구 방향을 제시한다.

II. 클러스터링 프로토콜 소개

본 장에서는 안전한 클러스터링 프로토콜들을 소개한다. 지금까지 제안된 모든 안전한 클러스터링 프로토콜들은 Low Energy Adaptive Clustering Hierarchy (LEACH) 프로토콜 [1] 에 기반으로 해서 설계되었다. 그래서 LEACH 프로토콜을 먼저 소개하고, LEACH 프로토콜에 안전성을 고려한 Secure LEACH (SecLEACH) 프로토콜 [2] 과 Grid-based Secure LEACH (GS-LEACH) 프로토콜 [3] 에 대해 소개한다.

2.1 LEACH 프로토콜

Heinzelman 등은 에너지를 효율적으로 쓰기위한 클러스터링 프로토콜인 LEACH 프로토콜을 제안하였다. LEACH 프로토콜은 전체 노드들의 에너지의 균등한 사용을 그 목적으로 한다. LEACH 프로토콜은 센서 노드들이 전체 센서 네트워크에 고르게 뿌려져 있고, 모든 노드들이 전송 파워를 높임으로써 베이스 스테이션에 직접적으로 데이터를 전송할 수 있다고 가정한다.

LEACH 프로토콜은 그림 1과 같이 형성 (setup) 단계와 지속(steady-state) 단계로 나뉘어 동작한다. 시간 동기화를 통해 모든 노드들은 각각의 라운드가 언제 시작할지 알 수 있다고 가정한다.

형성 단계는 세 가지 스텝들로 구성되어 있다. 첫 번째 스텝에서 노드들은 확률 함수를 사용하는 자가 선출 알고리즘을 통해 스스로 이번 라운드에 클러스터 헤드가 될지를 결정한다. 각 라운드에 선출되는 클러스터 헤드의 수는 배포 전에 미리 결정되어 선출 알고리즘에 반영된다. 스스로 선출된 클러스터 헤드 노드들은 광고 (advertisement) 메시지를 브로드캐스트한다. 이때 메시지간의 충동을 막기 위해 CSMA (Collision Sense Multiple Access) 프로토콜이 쓰인다. 두 번째 스텝에서 광고 메시지를 받은 노드들은 가장 가까운 클러스터 헤드를 선택하기 위해서 가장 강한 신호의 세기로 보낸 클러스터 헤드에 참여 (join) 메시지를 보낸다. 세 번째 스텝에서 각 클러스터 헤드는 참여 메시지를 보낸 노드들을 자신의 멤버노드로 뽑고, 확인 (acknowledgement) 메시지와 함께 지속 단계 동안에 쓰일 시간대 일정 (time slot schedule)을 브로드캐스트한다.

이렇게 클러스터 형성되고 난 다음에는 지속 단계로 들어간다. 지속 단계에서는 각각의 노드들이 패킷 충돌을

형성 (setup) 단계

1. $H \Rightarrow \varsigma : id_H$, adv
2. $A_i \rightarrow H : id_A, id_H$, join_req
3. $H \Rightarrow \varsigma : id_H, (\ldots, <id_A, t_A>, \ldots)$, sched

지속 (steady) 단계

4. $A_i \rightarrow H : id_A, id_H, d_A$
5. $H \rightarrow BS : id_H, id_{BS}, F(\ldots, d_A, \ldots)$

기호 설명

A_i, H, BS : 일반 노드, 클러스터 헤더, 베이스 스테이션
ς : 네트워크의 모든 노드들
$\Rightarrow$, $\rightarrow$: 브로드캐스트 전송, 유니캐스트 전송
id_x : 노드 x의 id
d_x : 노드 x로부터의 센싱 데이터
$<id_x, t_x>$: 노드 x의 id와 보고 시간대
adv(공고), join_req(참여), sched(일정) : 메시지 타입
F : 데이터 퓨전 함수

그림 1 LEACH 프로토콜

막기 위해 세 번째 스텝에서 받은 시간대 일정을 사용하여 Time Division Multiple Access (TDMA) 방식의 통신을 사용한다. 멤버 노드들은 자신의 전송 시간대가 오면 자신이 센싱한 데이터를 클러스터 헤더에게 전송한다. 다섯 번째 스텝에서 각 클러스터 헤더는 전송할 양을 줄이기 위해 멤버 노드들로부터 받은 데이터들을 퓨전 함수를 이용해 융합하고, 그 결과 값을 베이스 스테이션에 전송한다. 보통 한번 클러스터링이 형성되고 나면, 효율적인 에너지 사용을 위해 위와 같은 전송이 여러 번 반복된다.

2.2 SecLEACH 프로토콜

LEACH 프로토콜은 설계 단계에서 안전성을 고려하지 않았기 때문에, 여러 가지 보안 문제점을 가지고 있다. Oliveira 등은 안전한 클러스터링을 위해 LEACH에 Eschenauer 등이 제안한 임의의 키 사전 분배 방식 [4] 을 적용한 SecLEACH 프로토콜을 제안하였다.

SecLEACH 프로토콜은 배포 전에 임의의 키 사전 분배 방식처럼 S개의 키들로 이루어진 키 풀(pool)과 키 id를 만든다. 노드들은 m개의 키들을 할당받는다. 이를 위해서 각각의 노드 x에 대해서 의사 난수 함수(pesudorandom function)을 이용하여 각 노드에 id_x를 정하고, 이 id_x를 의사 난수 발생기의 시드 값으로 이용하여 m 번만큼의 수열을 생산한다. 이 m개의 수열에 mod S 연산을 함으로써 0에서 S-1 사이의 값들로 사상(mapping)하고, 그 값들에 해당하는 m개의 키 집합을 할당받는다. 또한 모든 노드들은 베이스 스테이션과의 통신을 위한 하나의 비밀 키를 할당받는다.

SecLEACH 프로토콜은 그림 2와 같이 LEACH 같은 다섯 단계의 스텝들로 동작한다. 첫 번째 스텝에서 자가 선출된 클러스터 헤드는 자신의 id와 하나의 랜덤 넘버를 브로드캐스트한다. 두 번째 스텝에서 이 광고 메시지를 받은 노드들은 받은 아이디를 이용하여 가장 가까운 클러스터 헤드의 키 아이디들을 계산하고, 자신이 가지고 있

형성 (setup) 단계

1. $H \Rightarrow \varsigma : id_H$, nonce, adv

   $A_i : r \in (R_H \cap R_A)$한 r을 선택

2. $A_i \rightarrow H : id_A, id_H, r$, join_req, $mac_{k_r}(id_A | id_H | r | nonce)$

3. $H \Rightarrow \varsigma : id_H, (..., <id_A, t_A>, ...)$, sched

지속 (steady) 단계

4. $A_i \rightarrow H : id_A, id_H, d_A, mac_{k_r}(id_A | id_H | d_A | nonce+j)$

5. $H \rightarrow BS : id_H, id_{BS}, F(..., d_A, ...), mac_{k_r}(F(..., d_A, ...) | c_H)$

기호 설명

r : 키 링에서 키들의 id

k_{H}: 키 id r의 대칭키

R_x: 노드 x의 키링의 키 아이디들의 집합

j : 현재 라운드의 보고 사이클(cycle)

그림 2 SecLEACH 프로토콜

는 키 링 중에서 중복되는 키 아이디가 있는지 찾고, 하나의 키를 선택한다. 이 키를 이용하여 메시지 인증 코드(Message Authentication Code)를 만들고 참여 메시지와 함께 그 클러스터 헤드에게 보낸다. 셋 번째 스텝에서는 LEACH와 마찬가지로 멤버 노드들에게 지속 단계 동안에 쓰일 시간대 일정을 브로드캐스트한다. 네 번째 스텝에서 멤버 노드들은 참여 메시지를 보낼 때와 마찬가지로 메시지 인증 코드를 만들어서 클러스터 헤드에게 센싱한 데이터를 전송하고, 다섯 번째 단계에서 클러스터 헤드는 데이터를 융합한 결과 값을 베이스 스테이션과 공유하고 있는 키를 사용하여 만든 메시지 인증 코드와 함께 베이스 스테이션에 전송한다.

2.3 GS-LEACH

SecLEACH에서 일반적인 노드는 자신과 키를 공유하고 있는 클러스터 헤드들중에서 가장 가까운 노드를 자신의 클러스터 헤드로 뽑는다. 만약 가장 가까운 노드가 자신과 키를 공유하고 있지 않다면, 다른 노드를 클러스터 헤드로 선택해야하는 단점이 있다. 또한 어떤 노드들은 모든 클러스터 헤드들이 자신과 키를 공유하고 있지 않아서, 많은 에너지를 소모하면서 베이스 스테이션에 직접 센싱한 데이터를 전송할 수도 있다.

Banerjee 등은 SecLEACH의 이런 문제점을 보완한 GS-LEACH를 제안한였다. GS-LEACH에서는 그리드 기반의 센서 노드들의 배포를 가정한다. 센싱 지역을 k개의 정사각형 모양의 그리드들로 나누고, n 개의 센서 노드들을 각각의 그리드에 배포한다. 그리고 각각의 그리드마다 S 개의 키들로 이루어진 풀을 만든다. 그리고 그 그리드에 배포되는 센서 노드들은 대응하는 키 풀에서 m 개의 키들을 뽑아서 할당한다. 모든 센서 노드들은 SecLEACH와 마찬가지로 베이스 스테이션과 하나의 비밀 키를 공유한다.

GS-LEACH는 SecLEACH와 똑같이 동작한다. 단지 클러스터링 형성은 각각의 그리드 안에서만 이루어지고, 보통 그리드 안에서는 하나의 클러스터 헤드만이 존재한다고 가정한다. 만약 클러스터 헤드와 키를 공유하지 않

은 노드들은 그 라운드를 쉬게 된다.

GS-LEACH는 Sec-LEACH와 비슷한 보안 레벨을 유지하면서, Sec-LEACH보다 평균적으로 클러스터 헤드와 멤버 노드간의 통신거리가 짧고, 클러스터 헤드와 키를 공유하지 않은 노드들은 직접적으로 베이스 스테이션과 통신하지 않고 그 라운드를 쉰다고 가정하기 때문에 에너지 소모가 적다. 또한 Sec-LEACH보다 적은 수의 키 링만으로도 비슷한 레벨의 연결성을 유지할 수 있다는 장점이 있다.

III. 클러스터링 프로토콜 분석

SecLEACH 와 GS-LEACH는 사실상 거의 비슷한 성능과 안전성을 지니고 있다. 차이점은 GS-LEACH는 그리드 기반의 노드 배포를 가정하기 때문에, 같은 크기의 키 링을 사용할 경우에 좀 더 향상된 노드간의 연결성을 제공한다. 또한 GS-LEACH는 그리드 안의 노드들과만 클러스터를 형성하기 때문에, SecLEACH보다 에너지 소모가 좀 더 작다.

두 프로토콜들은 공통적으로 세 가지 취약점들을 가지고 있다. 첫째로, 클러스터 헤드들이 멤버 노드들에게 브로드캐스트할 때 인증을 하지 않기 때문에 웜홀 공격이나 헬로우 플루드 공격같은 서비스 거부 공격들 [6] 에 취약하다는 점이다. 안전한 클러스터링 프로토콜들은 광고 메시지를 보낼 때 등 클러스터 헤드에서 멤버 노드들에게 패킷을 전송할 때 브로드캐스트를 주로 사용한다. 하지만, 이때 클러스터 헤드는 멤버 노드들에게 자신이 적법한 클러스터 헤드인지 인증하지 않기 때문에, 서비스 거부 공격들에 약점을 가진다. 만약 공격자가 많은 에너지를 사용하여 광고 메시지를 보내면, 근처의 많은 노드들이 그 노드를 자신의 클러스터 헤드로 뽑고, 잘못된 클러스터 링을 형성할 수 있다. 또한 공격자 노드가 먼 곳에 있는 클러스터 헤드가 보낸 광고 메시지를 반복함으로써 공격자 주위의 노드들이 이 노드가 가까이 있다고 생각하고 먼 거리에 있는 클러스터에 포함되게 할 수 있다.

둘째로, 임의의 키 사전분배 방식을 쓰기 때문에 노드 탈취 공격에 취약하다 [5] . 기존의 프로토콜들은 키 관리 방법으로 임의의 키 사전 분배 방식을 사용하고 있다. 이런 임의의 키 사전 분배 방식은 미리 정해진 키 풀에서 임의로 어느 정도 수의 키들을 뽑아서 노드들에 할당하는 방법이기 때문에, 적은 수의 탈취된 노드만으로도 키 풀의 많은 키들을 복구할 수 있다.

마지막으로, 주변의 통신 가능한 노드들과 완전한 연결성을 제공하기 힘들다는 것이다. 확률적인 키 관리 방식에 기반 하기 때문에, 자신의 주변에 통신 가능한 노드들과 완전한 연결성을 제공하기 힘들다. 주변의 모든 노드들과의 연결성을 유지하기 위해서는 네트워크 전체의 노드들과 연결성을 유지할 만큼의 키들을 가지고 있어야 한다. 어떤 노드들은 자신의 주위에 가장 가까운 클러스터 헤드와 키를 공유하지 않아서, 멀리 있는 다른 클러스터 헤드에 종속되거나, 모든 클러스터 헤드와 키를 공유하지 않아서, 많은 에너지를 소비하면서 베이스 스테이션과 통신해거나 그 라운드를 쉬어야 한다. 위의 내용들은 표 1에 정리되어 있다.

IV. 향후 연구 방향

표 1. 안전한 클러스터링 프로토콜들의 분석

클러스터링 프로토콜	클러스터 헤드와 멤버 노드간의 인증성	클러스터 헤드와 멤버 노드간의 기밀성	통신 가능한 노드간의 연결성	웜홀 공격 (wormhole attack)	헬로우 플루드 공격 (HELLO flood attack)
SecLEACH	부분적으로 가능	가능	완전하지 않음	취약함	취약함
GS-LEACH	부분적으로 가능	가능	완전하지 않음 (SecLEACH 보다는 향상)	취약함	취약함

현재까지의 안전한 클러스터링에 대한 연구들은 모두 LEACH 프로토콜에 기반하고 있다. 하지만, LEACH 프로토콜은 여러 가지 제한점들이 있기 때문에, 실제 환경에서 적용하기에는 한계가 있다. LEACH 프로토콜은 모든 노드들이 베이스 스테이션에 한 홉만에 전송할 수 있다고 가정한다. 이는 실제 응용프로그램에서는 불가능한 일이다. 또한 LEACH 프로토콜은 균등한 클러스터링을 형성하기가 힘들다. 클러스터 헤드들이 서로의 위치에 대한 고려 없이 확률적으로 자가 선출하기 때문에, 어떤 클러스터 헤드들은 서로 너무 가까이 있을 수 있고, 어떤 클러스터 헤드들은 너무 떨어져 있어서 균등한 클러스터링들에 비해 에너지 소모가 크다.

이런 LEACH 프로토콜의 문제점들을 해결하기 위해서 HEED [7] , ACE [8] , FLOC [9] 같은 클러스터링 프로토콜들이 제안되었다. 이중에서 HEED 프로토콜은 LEACH 프로토콜과 달리 효율적 에너지 사용뿐만 아니라 균등한 클러스터링을 고려한다. HEED 프로토콜에서 클러스터 헤드는 자가 선출 알고리즘에 의해 한 번에 결정되는 것이 아니라, 여러 후보들 중에서 가장 적은 통신비용이 드는 클러스터 헤드가 선택된다. 그래서 각각의 클러스터 헤드들은 전체 네트워크에 균등하게 분포되고 LEACH 프로토콜보다 적은 양의 에너지를 사용한다. 또한 클러스터 헤드가 베이스 스테이션에 데이터를 전송할 때 LEACH 프로토콜처럼 한 번에 보낼 수 있다고 가정하지 않고, 클러스터 헤드간의 라우팅을 통해 전송한다. 두 프로토콜 의 형성시간은 상수로 일정하기 때문에 클러스터 형성을 위해 많은 에너지가 소모되지 않음을 알 수 있다. LEACH 프로토콜은 노드들의 이동성을 지원할 수 있는 반면에 HEED 프로토콜은 모든 노드들이 고정적일 때만 적용할 수 있다는 단점이 있다. 하지만 이것은 LEACH 프로토콜이 모든 노드들이 한 홉만에 베이스 스테이션에 데이터를 전송할 수 있다고 가정하기 때문에 가능한 일이다. 이런 HEED 프로토콜과 LEACH 프로토콜의 차이점들은 표 2에 정리되어 있다.

표 2. LEACH 프로토콜과 HEED 프로토콜의 성능 비교

클러스터링 프로토콜	형성 시간	노드 이동성	노드와 베이스 스테이션과의 연결	에너지 효율성 고려 여부	균형적인 클러스터 형성 여부	베이스 스테이션에 보고 방법
LEACH	O(1)	베이스 스테이션만 고정	한 홉으로 가정	안함	중간	직접적
HEED	O(1)	고정	없음	고려함	높음	라우팅

기존의 안전한 클러스터링 프로토콜은 무한한 노드들의 전송 거리등의 가정을 가진 LEACH 프로토콜을 기반

으로 하기 때문에 HEED 프로토콜에 바로 적용하는 것은 힘들다. 앞으로의 안전한 클러스터링 연구들은 LEACH 프로토콜보다는 HEED 프로토콜 같은 좀 더 발전된 클러스터링 프로토콜에 기반으로 해서 설계되어야 할 것이다.

또한 4장에서 언급했던 기존의 안전한 클러스터링 프로토콜들의 문제점을 해결하기 위해서 문제점이 많은 기존의 임의의 키 사전 분배방식을 버리고, 지역화된 키 관리 방법이 필요하다고 본다. 즉 자신의 주위의 통신 가능한 모든 노들과 연결성을 갖고 노드 탈취 공격으로 부터도 좀 더 안전할 수 있는 키 관리 방법이 필요하다. 이를 위해서 기존의 키 관리 방법을 클러스터링 프로토콜에 맞게 고치거나 새로운 키 관리 기법이 제안되어야 할 것이다. 위의 발전된 프로토콜들은 대부분 클러스터링을 형성할 때 여러 번 자신을 알리는 헬로우 메시지를 보내야 하기 때문에, LEACH 프로토콜보다 더욱 서비스 거부 공격에 취약하다. 클러스터 헤드의 브로드캐스트 인증이 지원된다면 대부분의 서비스 거부 공격들에 안전할 수 있을 것이다. 다만, 이런 브로드캐스트 인증은 센서 노드들의 제한된 배터리와 연산능력을 고려해서 대칭키 기반의 암호를 기반으로 해서 효율적으로 설계되어야 할 것이다.

V. 결론

배터리가 제한적인 무선 센서 네트워크에서 효율적인 에너지 사용은 중요한 문제들 중에 하나이다. 이런 문제를 해결하기 위해서 에너지를 효율적으로 사용함으로써 네트워크 수명을 늘릴 수 있는 클러스터링 프로토콜들이 연구되고 있다. 하지만, 다른 부분에 비해 안전한 클러스터링에 대한 연구는 아직 미미하다. 본 논문에서는 현재까지 연구된 안전한 클러스터링 프로토콜들을 대해서 살펴보고, 분석했다. 기존의 프로토콜들은 안전성에 문제가 있을 뿐만 아니라, 발전된 클러스터링 기법들에 적용하기가 무리가 있음을 말하고 이를 해결하기 위한 앞으로의 연구 방향에 대해서 제시하였다. 위의 연구 방향을 바탕으로 좀 더 안전하고, 발전된 클러스터링 프로토콜을 기반으로 하는 안전한 클러스터링 프로토콜을 설계하는 일은 향후 연구 과제로 남겨둔다.

참 고 문 헌

[1] Heinzelman. W.B., Chandrakasan. A.P., and Balakrishnan. H., "An Applicatoin-specific rotocol architecture for wireless microsensor networks," IEEE Transactions on Wireless Communications, volume: 1, Issue: 4, pp 660-670, Oct 2002.

[2] Leonardo B. Oliveira, Hao C. Wong, M. Bern, Ricardo Dahab, A. A. F. Loureiro, "SecLEACH - A Random Key Distribution Solution for Securing Clustered Sensor Networks," Fifth IEEE International Symposium on Network Computing and Applications (NCA'06), pp. 145-154, 2006.

[3] P. Banerjee, D. Jacobson, and S. N. Lahiri, "Security and Performance Analysis of a Secure Clustering Protocol for Sensor Networks", in Proceedings of the 6th IEEE International Symposium on Network Computing and Applications, July 2007.

[4] L. Eschenauer and V. D. Gligor, "A key management scheme for distributed sensor networks," In 9th

ACM conference on Computer and Communication Security (CCS), pages 41-47, 2002.

[5] T. Moore, "A Collusion Attack on Pairwise Key Predistribution Schemes for Distributed Sensor Networks", Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications Workshops(PERCOMW '06), 13-17 March 2006.

[6] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," In Proc. of 1st IEEE international Workshop on Sensor Network Protocols and Application, May 2003.

[7] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed clustering approach for Ad Hoc sensor networks," IEEE Transactions on Mobile Computing vol. 3, issue. 4, pp 366-379, 2004.

[8] H. Chan, A. Perrig, "ACE: an emergent algorithm for highly uniform cluster formation," Proc. of 1st European Workshop on Sensor Networks (EWSN), Berlin, Germany, January 2004.

[9] M. Demirbas, A. Arora and V. Mittal, "FLOC: a fast local clustering service for wireless sensor networks," Proc. of Workshop on Dependability Issues in Wireless Ad Hoc Networks and Sensor Networks (DIWANS'04), Palazzo dei Congressi, Florence, Italy, June 2004.