DOI: 10.3745/KIPSTC.2010.17C.6.441

센서 네트워크에서의 안전한 그룹통신을 위한 상호 인증 기법

고 혜 영[†] · 도 인 실[††] · 채 기 준[†††]

요 약

센서 네트워크는 유무선 네트워크 환경에 다양한 센서를 설치하고 이를 통해 데이터를 감지하며 감지된 데이터를 응용서비스 서버와 연동하는 기술로 최근 다양한 연구가 이루어지고 있다. 그러나 이러한 센서 네트워크는 센서 노드 자체의 제약점 때문에 메모리와 처리 능력, 에너지 수명에 제한을 가지며 그럼으로써 센서 네트워크 자체에 보안상의 취약점이 존재한다. 그러므로 센서 네트워크의 기술에 있어서 보안에 관한 연구가 매우 중요하며 센서의 계산 능력 또한 고려되어야 할 부분이다.

본 논문에서는 위와 같은 특징을 반영하여 일반 센서 노드들과 충분한 저장 공간과 계산 능력을 갖는 노드인 클러스터 헤더가 있는 이종의 센서로 네트워크를 구성하고 키 관리 기법 중 하나인 PCGR(Predistribution and local Collaboration-based Group Rekeying) 기법을 기반으로 하여 안전한 그룹 통신을 위한 그룹키 갱신 방법을 제안한다. 제안된 방법에서는 센서 노드 측에서 클러스터 헤더로부터 받은 새로운 키 정보를 인증하도록 하여 오염된 키 정보로 인해 네트워크의 안전성이 위협받는 상황을 최소화함으로써 보안성을 향상시킬 수 있도록 하였다. 즉, 그룹키를 갱신할 때, 클러스터 헤더는 노드가 보내주는 노드의 부분 정보를 검증함으로써 노드의 무결성을 확인하고 그룹키를 갱신하며 갱신된 그룹키는 다시 노드들에게 보내져서 노드들에 의해 재검증됨으로써 클러스터 헤더가 공격받았는지 아닌지에 대한 여부를 검사한 후 안전하게 그룹키를 갱신할 수 있도록 한다. QualNet 시뮬레이터를 이용한 실험을 통해 제안한 기법이 네트워크의 보안성을 높임은 물론 오버헤드 및 에너지 소모량이 기존의 그룹키 관리 기법보다 효율적임을 보인다.

키워드 : 센서 네트워크, 보안, 그룹키, 상호인증

Mutual Authentication Mechanism for Secure Group Communications in Sensor Network

Hyeyoung Ko[†] · Inshil Doh[††] · Kijoon Chae[†††]

ABSTRACT

Recently, a lot of interest is increased in sensor network which gathers various data through many sensor nodes deployed in wired and wireless network environment. However, because of the limitation in memory, computation, and energy of the sensor nodes, security problem is very important issue. In sensor network, not only the security problem, but also computing power should be seriously considered.

In this paper, considering these characteristics, we make the sensor network consist of normal sensor nodes and clusterheaders with enough space and computing power, and propose a group key rekeying scheme adopting PCGR(Predistribution and local Collaboration-based Group Rekeying) for secure group communication. In our proposal, we enhance the security by minimizing the risk to safety of the entire network through verifying the new key value from clusterheader by sensor nodes. That is, to update the group keys, clusterheaders confirm sensor nodes through verifying the information from sensor nodes and send the new group keys back to authentic member nodes. The group keys sent back by the clusterheaders are verified again by sensor nodes. Through this mutual authentication, we can check if clusterheaders are compromised or not. Qualnet simulation result shows that our scheme not only guarantees secure group key rekeying but also decreasesstorage and communication overhead.

Keywords : Sensor Network, Security, Group Key, Mutual Authentication

1. 서 론

오늘날 유비쿼터스 기술이 발전함에 따라 다양한 분야에서의 센서 네트워크의 활용이 증가하고 있으며 관련 분야의 IT 산업의 관심과 규모가 더욱 커지게 되었다. 따라서 유비쿼터스 컴퓨팅 환경의 기본이 되는 기반 기술인 센서 네트

※ 이 논문은 2009년도 정부(교육과학기술부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임.(NO.R01-2009-0083-985)
† 준 회 원 : 이화여자대학교 컴퓨터공학과 석사
†† 정 회 원 : 이화여자대학교 컴퓨터공학과 연구교수(교신저자)
††† 종신회원 : 이화여자대학교 컴퓨터공학과 교수
논문접수 : 2010년 1월 29일
수 정 일 : 1차 2010년 9월 13일, 2차 2010년 10월 19일
심사완료 : 2010년 10월 20일

워킹 기술의 중요성 또한 강조되고 있다. 센서 네트워크는 유무선 네트워크 인프라에 다양한 센서를 설치하고, 이를 통해 데이터를 감지하며 감지된 데이터를 응용서비스 서버와 연동하는 기술로 이러한 센서 네트워크는 작은 크기의 센서를 사용해야 하는 제약점 때문에 메모리와 처리 능력, 에너지에 제한을 갖는다. 그러므로 센서 네트워크상에서의 기술 연구에 있어서 센서가 갖는 무선 통신의 기본 취약점에 대한 보안이 매우 중요하며, 센서의 계산 능력 또한 고려되어야 할 부분이다[7, 8].

본 연구에서는 이러한 무선 센서 네트워크의 특징과 그룹키 관리 기법들을 분석하여 무선 센서 네트워크에 적합한 효율적인 그룹키 관리 기법을 제안하였다. 제안 기법에서는 클러스터헤더(CH)를 중심으로 한 클러스터링 구조를 기본으로 하였으며 그룹키를 이용하며 CH가 그룹키 다항식을 생성하고 이러한 정보를 암호화하여 저장함으로써 좀 더 안전한 그룹키 갱신을 기대할 수 있다. 또한 그룹키를 갱신할 때에 CH와 노드가 상호 인증을 함으로써 보안성을 더욱 향상시킬 수 있다.

본 논문의 구성은 다음과 같다. 1장의 서론에 이어 2장에서는 센서 네트워크에 적합한 그룹키 관리 기법과 관련된 연구에 대해 살펴본다. 이를 기반으로 3장에서는 기존의 연구들을 보완한 효율적인 그룹키 관리 기법을 제안할 것이다. 이어서 4장에서는 제안하는 기법의 에너지 효율성 및 안전성에 대한 성능 평가를 위한 시뮬레이션 환경과 시나리오를 설명하고 기존의 기법들과의 시뮬레이션을 통해 비교 분석한 후, 5장에서는 본 논문의 결론과 향후 연구에 대해서 기술한다.

2. 관련 연구

이 장에서는 센서 네트워크에서의 그룹키 관리 기법을 소개하고 본 연구의 기반이 되는 CPGR 기법에 대해 서술한다.

2.1 그룹키 관리 기법

그룹키 관리 기법을 관리의 주체에 따라 분류를 하면 중앙형 키 관리 기법, 중앙 분산형 키 관리 기법, 기여형 키 관리 기법으로 분류할 수 있다.

2.1.1 중앙형 키 관리 기법

중앙형 그룹 키 관리 기법은 하나의 그룹 관리자가 그룹키를 생성하여 생성된 키를 합법적인 전체 그룹 멤버에게만 나누어 주며 그룹키를 생성한 키 재료와 목록을 관리한다. 대표적인 연구를 살펴보면 먼저 Carlo Blundo, Alfredo De Santis가 제안하는 방식은 서버에서 만들어낸 다항식을 멤버들에게 나누어주어 같은 값을 넣어 같은 그룹키값을 도출하도록 하는 기법을 제안하였다[1]. 그 결과로 모든 멤버들은 같은 그룹키 값을 얻을 수 있게 되며 그것을 그룹키로 이용할 수 있다. 이 방법은 다항식을 사용하여 그룹키를 생성하는 방식의 기초가 되는 방법이지만 그룹키를 갱신하는 방법을 포함하고 있지 않을 뿐 아니라 각 노드가 그룹키를 생성하는 다항식을 가지고 있어 공격받을 시에 그룹키가 쉽게 드러날 수 있다는 단점이 있다. Yong Wang, Byrav Ramamurthy는 4가지의 안전한 그룹 통신 방법을 제안하였다[2]. 먼저 유니캐스팅 방법은 그룹키를 갱신할 때의 정보를 각 노드에게 일일이 보내주는 방법으로 그룹이 커질 경우 오버헤드가 발생한다. 이를 보완하여 브로드캐스팅을 제안하였는데 이는 처음 그룹을 생성할 때에 유니캐스팅보다는 좀 더 오버헤드가 발생하지만 키 갱신에서는 그보다 적은 오버헤드를 발생시키는 장점이 있다. 세 번째로 Flooding 공격을 예방하기 위해 오버랩핑을 제안하여 좀더 안전한 네트워크는 구축할 수 있도록 하였으며, 마지막으로 선적재방식을 통해 그룹이 생성되는 시간을 최소화하도록 하였다.

2.1.2 중앙 분산형 키 관리 기법

중앙 분산형 키 관리 기법은 여러 개의 그룹 관리자가 각각 그룹키를 생성하여 자신의 그룹 안에 있는 합법적인 멤버들에게 나누어주는 방식이다.

Wensheng Zhang, Sencun Zhu은 키 정보를 선분배하고 노드간의 협력을 통해 그룹키를 재생성하는 기법인 PCGR을 제안하였다[3]. 이 기법은 본 연구와 밀접한 관련이 있으므로 2.2절에서 좀 더 자세히 언급하도록 하겠다.

Jyh-How Huang, Jason Buckingham은 한 홉을 거칠 때마다 데이터를 재 암호화하도록 하여 데이터 기밀성을 제공하는 단계적 키 구조를 제안하였다[4]. 이 기법에서는 클러스터를 형성한 그룹들을 라우팅 트리로 나누어 관리하고 노드간의 통신은 같은 레벨에 있는 키로 암호화하고 키는 노드가 새로 들어오거나 떠날 때 갱신된다. 그룹 내의 노드가 제거되는 방법은 세 가지로 분류되는데 Active Leaving의 경우 노드가 CH에게 직접 알리는 것이고, Passive Leaving 일 경우 하드웨어 문제나 물리적인 손상을 입었을 경우 CH에게 알리지 못하고 제거되는 경우이다. 마지막 방법은 베이스 스테이션(BS)에서 이상행동을 하는 노드를 탐지하고 그것을 CH에게 알려서 그룹에서 제거하도록 하는 것이다. 이 방식은 오버헤드나 에너지 효율이 좋지만, 많은 노드가 바뀌게 될 경우 작업속도가 감소한다는 단점이 있다.

Sencun Zhu, Sanjeev Setia는 규모가 큰 센서 네트워크 환경 안에서의 지역적 암호화와 인증을 통한 키 관리 기법을 제안하였다[5]. 이 방법은 센서 노드와 BS 간의 공유키와 두 센서 간의 공유키, 같은 클러스터 안에 있는 센서들 사이에 공유된 클러스터 키, 마지막으로 네트워크 안에 있는 모든 센서들 사이에 공유된 그룹키를 이용하여 보안 서비스를 제공한다. 위의 키들을 이용하여 한 홉에서의 브로드캐스트 통신을 인증하고 위장 공격을 막으며 타임스탬프를 이용하여 노드 캡처나 시빌(Sybil) 공격을 막고 전체 네트워크 수명을 일정한 세션으로 나누어 각 세션마다 새로운 키를 갱신한다. 키의 종류가 많기 때문에 보안이 나누어져 좀더 안전하나 키를 갱신할 때에 많은 부하가 발생하게 된다.

2.1.3 기여형 키 관리 기법

기여형 키 관리 기법은 중앙형이나 중앙 분산형과 같은 특정한 키 관리자를 필요로 하지 않으며 노드들끼리 협력을 통해서 그룹키를 갱신하는 방법이다.

Zhen Yu, Yong Guan이 제안하는 그룹키 관리 기법[6]은 네트워크를 육각형 그리드로 분할한 후에 센서 노드에게 기본 행렬인 G외에 비밀 행렬 값인 A와 B를 사전에 분배받아 같은 그룹의 노드 간에는A행렬을 이용하여 키를 생성하고 다른 그룹에 속한 노드 간에는 B행렬을 이용하여 키를 생성한다. 이 방법은 노드 간의 키가 생성될 확률이 높아지지만 최적화된 그리드 사이즈가 결정되지 않았으며 그리드 사이즈가 클 경우에 그룹간의 키를 생성할 때에 많은 에너지가 소모되며 그리드 사이즈가 작을 경우에는 그룹간의 키가 생성되지 않을 수 있다.

2.2 PCGR기법

그룹키를 갱신할 때 그룹키를 실제로 만드는 노드가 공격을 받더라도 다음에 생성될 그룹키를 알 수 없도록 하는 방법으로, 노드 간의 협력을 통해서 안전하게 그룹키를 갱신할 수 있다. PCGR기법의 경우 그룹키를 갱신하는 노드가 별도로 정해져있지 않고 네트워크 필드에 있는 임의의 노드에서 키를 갱신할 수 있다. 즉, 자신이 가지고 있는 정보를 이용하여 주변 노드에게 정보를 받아 키를 갱신함과 동시에 주변 노드에게 자신이 갖고 있는 정보를 이용하여 부분정보를 계산하여 돌려줌으로써 주변 노드 역시 그룹키 갱신을 할 수 있도록 도와주는 역할을 해야하므로 키 갱신의 주체가 다수의 노드가 될 수 있다는 단점을 갖는다. 키 갱신은 다음의 세 단계에 의해 이루어진다.

• 그룹키 사전 분배

:키 서버가 그룹의 수를 정하고 각 그룹마다 t차 다항식 g(x)를 생성하여 그룹에 포함되는 노드에게 각각 사전에 분배한다.

• 다항식의 암호화 및 암호화 다항식 부분정보의 분배

:노드 N_u가 배치된 후 임의로 이변다항식 $e_u(x,y)$를 다음과 같이 생성한다. 이 때 μ는 시스템 변수이다.

$$e_u(x,y) = \sum_{0\le i\le t, 0\le j\le \mu} A_{i,j}x^i y^j$$

노드 N_u는 자신이 생성한 암호화 다항식의 y항에 자신의 아이디 u를 대입하여 g(x)와 동일한 차수의 다항식을 생성한 다음 이를 이용하여 그룹키 다항식g(x)를 다음과 같이 암호화함으로써 g'(x)를 만든다. g'(x)는 자신이 키 갱신의 주체가 되어 키를 계산할 때 사용할 정보이다.

$$g'(x) = g(x) + e_u(x,u)$$

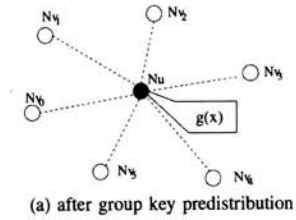
다음으로 노드 N_u는 자신의 암호화 다항식 $e_u(x,y)$의 y대신 주변 노드의 아이디 v_i를 대입하여 $e_u(x,v_i)(0\le i\le n-1)$ 형태로 만들어 주변 노드들(N_{vi})에게 보내고 난 후 자신이 가지고 있는 g'(x)를 제외한 g(x)와 e(x,y)를 삭제함으로써 그룹키를 생성하는 그룹키 다항식이 공격자에게 노출될 가능성을 제거한다. 추후에 주변 노드에게 나누어 주었던 부분 정보를 재취합함으로써 원래의 g(x)를 계산한다. 부분 정보 분배가 끝난 후 각 노드가 갖고 있는 정보는 (그림 1)-(d)와 같다.

• 키 갱신

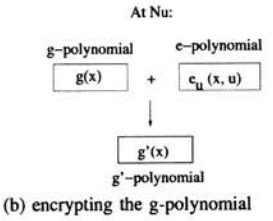
현재 그룹키 버전을 c라 할 때 각 노드 N_u는 c 값을 1 증가시키고 각 노드는 각각 갖고 있는 부분정보의 x 대신 증가된 c값을 입력하여 $e_u(c,v_i)$를 계산하여 이웃 노드가 g'(x)를 계산할 수 있도록 보내주고 자신은 이웃 $\mu+1$ 개의 노드로부터 부분정보를 받아 다음과 같은 방식으로 μ 차 암호화 다항식을 다시 복원한다.

$$\sum_{j=0}^{\mu}(v_i)^j B_j = e_u(c,v_i), (0\le i\le \mu).$$

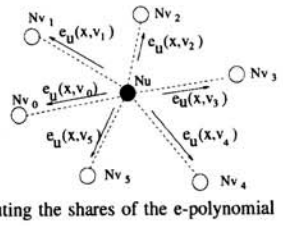
$$e_u(c,y) = \sum_{j=0}^{\mu} B_j y^j$$



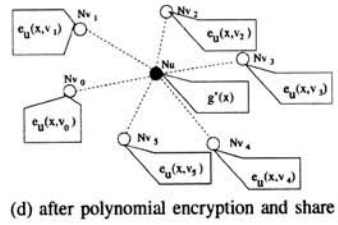
(a) after group key predistribution



(b) encrypting the g-polynomial

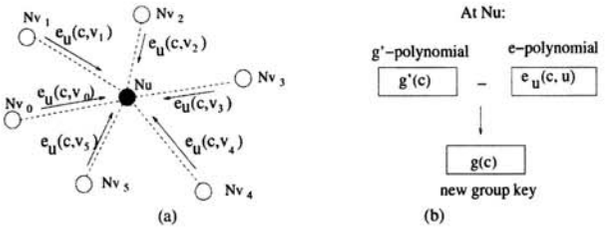


(c) distributing the shares of the e-polynomial



(d) after polynomial encryption and share distribution

(그림 1) PCGR기법의 그룹 초기화



(그림 2) PCGR기법의 키 갱신 방법

(그림 2)에서와 같이 주변 $\mu+1$ 개의 이웃 노드로부터 부분 정보를 받아 $e_u(c,y)$를 복원한 N_u는 y 항에 이번에는 자신의 아이디를 대입하여 다음과 같은 식으로 새로운 그룹키를 계산한다.

$$g(c) = g'(c) - e_u(c,u)$$

이와 같이 PCGR기법은 네트워크 필드의 중간 중간에 위치한 노드들에서 주기적으로 그룹키를 갱신하며 그룹키 갱신에 실제로 사용되는 다항식 자체를 보관하지 않고 필요할 때 주변 노드로부터 계산된 부분 정보를 받아 이를 취합함으로써 보안을 유지할 수 있다는 장점을 갖는다.

3. 그룹키 분배 기법 제안

본 장에서는 PCGR[3]을 기반으로 하여 CH와 노드 간의 상호 인증을 통해 안전한 그룹키 생성을 유도하며 그에 따라 안전한 네트워크 통신을 할 수 있도록 하는 그룹키 관리 기법을 제안한다. 본 연구에서는 PCGR기법에서 센서 노드가 그룹키를 만드는 노드로부터 받은 정보를 검증할 수 없다는 단점을 보완하여 상호 간에 인증을 수행함으로써 보안성을 더 향상시키고자 한다. PCGR 기법은 노드가 공격자에게 공격당하더라도 다음 그룹키가 드러나지 않는다는 장점이 있지만 일반 노드가 계산을 수행하고 계산 결과를 그룹키를 만드는 노드에게 전해주면 이 노드가 새로운 그룹키를 계산하여 일반 노드에게 전달하고 일반 노드는 이 정보를 무조건 신뢰한 상태에서 그룹키를 갱신하기 때문에 올바른 그룹키를 받았는지 알 수 없다는 단점이 있다. 특히 그룹키를 계산하는 노드가 오염되는 경우 네트워크 전체에 심각한 문제가 발생하기 때문에 제안하는 기법에서는 이러한 보안상의 문제점을 개선하기 위하여 그룹키를 계산하는 기능을 CH에서 하도록 하였고 CH에서 계산된 그룹키 값을 그룹의 멤버인 노드에서 다시 한번 인증함으로써 CH가 오염된 경우 네트워크 전체에 미치는 영향을 현저하게 줄이고자 한다.

3.1 제안 기법의 구조와 가정

본 논문에서는 이종의 노드로 센서 네트워크를 구성한다. 하나는 일반 노드보다 전력 면이나 계산 능력이 뛰어난 클러스터헤더(CH)이고 다른 하나는 제약적인 능력을 가진 일반 노드(SN)이다. 센서 네트워크는 BS와 CH, 그리고 각 클러스터 내에 존재하는 SN들로 구성되어 있다. BS에서는 모든 정보를 수집하고 그룹키를 생성할 때 신뢰할 수 있는 제3의 엔티티이다. CH는 일반 센서와의 원활한 통신을 위해 그룹의 중앙에 위치하고 있으며 그룹 당 하나씩 존재한다.

〈표 1〉 표기법

표기	설명
BS	베이스스테이션
CH	클러스터헤더
SN	센서 노드
CH_ID_u	CHu의 아이디
SN_ID_v	SNv의 아이디
Advertisement, Join_request, Accept, Request_share Confirm	이벤트 메시지
$g_u(x)$	그룹 u의 그룹키 다항식
$e_u(x,y)$	그룹 u의 그룹키 암호화 다항식
$g_u'(x)$	eu(x,y)로 암호화 된 그룹 u의 그룹키다항식
$a_u(x)$, $d_u(x)$, $q_u(x)$	gu(x)를 만들기 위한 다항식
$a_u(x,y)$, $d_u(x,y)$, $q_u(x,y)$	eu (x,y)를 만들기 위한 다항식
K_{vu}	두 노드(예:v와 u)사이의 공유키
$g_u(c)$	그룹 u의 현재 세션의 그룹키
h(c)	현재 세션의 CH 간 그룹키
$g_u(r)$	그룹 u의 새로운 그룹키
h(r)	새로운 CH간 그룹키
$i_u(r)$	오염된 노드를 제외한 그룹u의 새로운 그룹키
w(x)	오염된 노드나 CH의 아이디를 입력하면 0이 되도록 만들어진 다항식
H(r), I(r)	w(x)와 h(r),i(r)을 이용하여 만든 다항식
E_K{Message}	K로 암호화한 메시지
r	난수

CH의 전송 범위는 일반 SN보다 넓어 클러스터 내의 모든 센서 노드에 미칠 수 있고 센서 네트워크에 노드가 배치되고 나서 초기의 그룹키를 생성하는 시간 동안 노드에 대한 공격은 없다고 가정한다.

3.2 그룹키 관리 기법

제안하는 그룹키 관리 기법은 기존에 제안되었던 PCGR 기법을 기반으로 하였다. PCGR기법은 그룹키를 생성하는 다항식을 암호화한 상태로 저장하고 복호화할 때 필요한 정보를 주변 노드들에게 나누어줌으로써 CH의 역할을 하는 노드를 공격하더라도 실제 그룹키를 생성하는 다항식을 알 수 없으며 그룹키를 갱신할 때 노드가 보내주는 정보를 통해서 노드의 공격 여부를 알 수 있도록 하였다. 하지만 CH가 공격 당했을 경우, 노드는 그 여부를 알 수 없기 때문에 그룹키를 받지 못하거나 공격자에 의해 위조 혹은 변조된 그룹키를 받음으로써 그룹통신에 참여할 수 없는 상황이 생길 수 있다는 단점을 갖는다. 또한 PCGR 기법의 경우 일반 노드와 그룹키를 생성하는 노드를 구별하지 않음으로써 하나의 그룹 내에 그룹키를 갱신하는 노드가 다수가 존재할

수 있어 그룹키 생성 및 갱신 시 불필요한 오버헤드가 발생한다는 문제점이 있다.

제안하는 기법은 그룹키를 갱신하는 노드(CH)와 일반 노드(SN)를 사전에 구별하여 그룹키 갱신 시 키 갱신을 주도하는 노드만이 일반 노드로부터의 정보를 검증하는 것이 아니라 일반 노드들도 갱신된 그룹키 정보를 검증하는 상호인증을 수행함으로써 보다 안전한 그룹키 갱신을 수행할 수 있도록 하였다. 사전에 최초의 그룹키를 분배받은 각 노드는 초기화 과정을 통해 CH의 경우 그룹키를 생성하는 다항식을 암호화한 형태로 저장하고 SN의 경우는 암호화를 위한 다항식의 부분정보만을 저장한다. 실제 갱신이 필요한 상황에서 CH는 멤버 SN들에게 그룹키 갱신을 위한 부분정보를 요청하고 이 요청에 따라 멤버 노드들이 부분 정보를 보내면 CH는 이를 취합하여 새로운 그룹키를 계산한다. 이때 CH도 그룹키를 생성하는 다항식을 통해 새로운 그룹키를 계산하는 것이 아니라 멤버 노드가 보내온 정보를 취합함으로써 새로운 그룹키를 계산하도록 하여 CH가 공격을 당하는 경우에도 그룹키 생성 다항식이 노출될 가능성을 없앤다. 이와 같이 생성된 새로운 그룹키를 이전 그룹키를 이용하여 멤버 노드들에게 암호화하여 전송하고 멤버 노드들은 이를 받아 자신이 갖고 있던 정보를 이용하여 다시 인증함으로써 잘못된 정보로 인해 그룹키가 제대로 갱신되지 않을 가능성을 배제한다. 이와 같이 각 그룹의 키 갱신 주체인 CH와 해당 그룹의 멤버들 간에 상호 인증을 수행함으로써 안전한 그룹키 갱신이 가능하며 또한 정해진 노드(CH)만이 그룹키 생성 및 갱신을 수행하므로 그룹키 갱신 시의 오버헤드를 줄일 수 있다는 장점을 갖는다. 이 과정을 단계별로 좀 더 자세하게 기술한다.

3.2.1 그룹 초기화

CH와 SN은 네트워크상에 배치되기 전에 BS로부터 각각 BS와의 공유키를 받는다.

Step 1. 그룹 u의 CH인 CH_u는 $a_u(x) \times d_u(x) + q_u(x) = g_u(x)$ 형식으로 만들어진 그룹키 다항식을 생성한다. 또한, 그룹키 다항식을 암호화하기 위해 암호화 다항식인 $a_u(x,y) \times d_u(x,y) + q_u(x,y) = e_u(x,y)$를 생성한다. 이와 같은 형태로 다항식을 생성하는 이유는 주변 노드로부터 받은 정보를, 그리고 CH로부터 받은 정보를 효과적으로 검증하기 위함이다. 그리고 나서 그룹을 형성하기 위해서 주변 노드에게 광고 메시지를 보낸다. 최초의 그룹키 값인 $g_u(0)$는 노드를 배치하기 전에 각 노드에 저장하여 최초 그룹키 갱신 시 암호화 키로 사용한다.

$CH_u \Rightarrow SN_v$: $CH\_ID_u || Advertisement$
($1 \le v \le n$, n은 각 CH의 전송 반경 내 SN의 수)

Step 2. 메시지를 받은 노드 SN_v는 CH_u에게 그룹 가입 요청 메시지를 보낸다. 만약 여러 CH들로부터 메시지가 올 경우에는 CH의 신호 세기를 비교하여 큰 신호 세기를 가진 CH를 선택하여 그룹 가입 요청 메시지를 보낸다.

$SN_v \Rightarrow CH_u$: $SN\_ID_v \parallel CH\_ID_u \parallel Join\_request$
($1 \le v \le n$, n은 전송 반경 내 SN의 수)

일정 시간 동안 가입 요청 메시지를 받은 CH_u는 $e_u(x,y)$의 y에 자신의 아이디를 대입하여 $e_u(x,CH_ID_u)$를 만들어 다음과 같이 $g_u(x)$를 암호화하여 $g_u'(x)$를 만든다.

$$g_u'(x) = g_u(x) + e_u(x,CH\_ID_u)$$

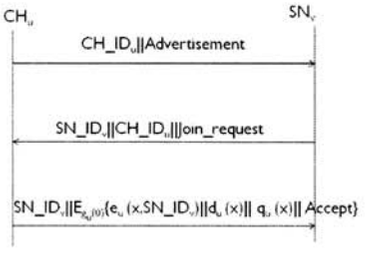
Step 3. 다음으로 CH_u는 가입 요청 메시지를 보낸 노드들의 아이디를 $e_u(x,y)$의 y항에 넣어 계산한 값과 $d_u(x)$ 및 $q_u(x)$를 담아서 $g_u(0)$로 암호화하여 전송한다. $g_u(0)$는 초기에 각 노드에 저장되어있으며 최초의 그룹키 설정 및 갱신에 필요한 정보를 전송할 때 일회성으로 사용하며 그 이후에는 각 그룹별로 생성한 그룹키를 이용하여 암호화한다.

$CH_u \Rightarrow SN_v$:
$SN\_ID_v || E_{gu(0)}\{e_u(x,SN\_ID_v) || d_u(x) || q_u(x) || Accept\}$
($1 \le v \le n$, n은 그룹 내 SN의 수)

전송이 끝나면 $g_u(x)$와 그것을 만들 때 사용된 $a_u(x)$, $d_u(x)$와 $e_u(x,y)$를 만들 때 사용된 $a_u(x,y)$를 삭제함으로써 CH가 오염되거나 정보가 노출됨으로써 $g_u(x)$가 공격자에 의해 생성되는 것을 방지한다. 그룹 초기화가 끝나면 CH_u와 SN_v가 가지는 정보는 아래와 같다.

CH_u: $g_u'(x)$, $d_u(x,y)$, $q_u(x,y)$
SN_v: $e_u(x,SN\_ID_v)$, $d_u(x)$, $q_u(x)$

(그림 3)은 그룹 초기화를 메시지 흐름에 따라 나타낸 그림이다.



(그림 3) 그룹 초기화 시의 클러스터헤더와 센서 노드 간의 메세지 흐름

3.2.2 그룹키 갱신

그룹키 갱신에는 두 종류가 있는데 CH와 SN간의 그룹키 갱신과 CH간의 그룹키 갱신이 있다.

• CH와 SN 사이의 그룹키 갱신

각 그룹별로 그룹키 갱신이 필요할 때 키를 갱신하는 과정이다.

Step 1. CH_u가 그룹에 속한 SN_v에게 $e_u(x,y)$의 부분 정보를 요청하는 메시지를 인증을 위한 난수 r을 포함하여 전송한다. 메시지를 받은 노드들은 CH_u에게 자신이 갖고 있는 암호화 다항식에 받은 난수를 대입하여 계산한 값을 현재의 그룹키 $g_u(c)$로 암호화하여 응답 메시지를 보낸다.

$SN_v \Rightarrow CH_u$: $E_{gu(c)}\{e_u(r,SN\_ID_v)\}$
($1 \le v \le n$, n은 그룹 내 SN의 수)

$e_u(r,SN_ID_v)$은 암호화를 위한 이변 다항식의 x에는 난수 값을, y에는 각 센서 노드의 아이디를 대입하여 계산하며 각 센서 노드가 갖는 그룹키 부분정보로서 상수 값을 갖는다.

Step 2. CH_u는 메시지를 받으면 먼저 부분 정보들을 검증한다. $a_u(x,y) \times d_u(x,y) + q_u(x,y) = e_u(x,y)$의 형태로 암호화 다항식을 생성했으므로 $e_u(r,SN_ID_v) \bmod d_u(r,SN_ID_v)$와 $q_u(r,SN_ID_v) \bmod d_u(r,SN_ID_v)$가 같은지 확인함으로써 보내온 정보를 검증한다. 그룹키 생성 다항식 자체가 사전에 이 식을 바탕으로 생성된 것이므로 값이 위조나 변조되지 않았다면 mod 연산에 의해 동일한 결과가 나오게 된다. 검증에 실패하면 SN_v로부터의 정보가 위조 혹은 변조된 것으로 판단한다. 검증을 통과한 t+1개의 부분 정보를 얻으면 $e_u(r,CH_ID_u)$를 재구성하여 아래와 같이 새로운 그룹키를 생성한다. 처음 생성한 그룹키 생성을 위한 다항식 $g_u(x)$가 t차인 경우 최소한 t+1개의 주변 노드로부터 부분정보를 받아야만 암호화 다항식의 값을 복원할 수 있고 이에 따라 새로운 그룹키 $g_u(r)$을 계산할 수 있다.

$$g_u'(r) - e_u(r,CH\_ID_u) = g_u(r)$$

Step 3. CH_u는 새로 만든 그룹키를 현재 세션의 그룹키 $g_u(c)$로 암호화하여 그룹 내 SN들에게 브로드캐스트 해준다.

$CH_u \Rightarrow SN_v$: $E_{gu\ (c)}\{g_u(r)\}$
($1 \le v \le n$, n은 그룹 내 SN의 수)

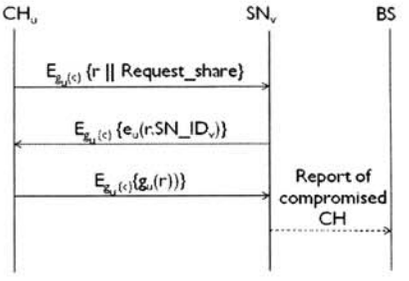
Step 4. 메시지를 받은 노드는 자신이 가지고 있는 $d_u(x)$와 $q_u(x)$의 x 대신 r값을 넣어 $q_u(r) \bmod d_u(r)$ 값과 $g_u(r) \bmod d_u(r)$ 값을 비교하여 결과 값이 다를 경우 검증이 실패한 것이므로 BS에게 자신이 속한 CH_u가 공격받았다는 것을 SN과 BS 사이의 대칭키로 암호화하여 전송한다.

(그림 4)는 CH와 센서 노드 간의 그룹키를 갱신하는 메시지의 흐름을 나타내는 그림이다.

• CH 사이의 그룹키 갱신

CH와 SN 간의 그룹키가 갱신된 후에 CH 간의 그룹키가 갱신된다. 초기의 그룹키는 BS와 CH간의 통신에 의해 각 CH에게 전달되며 이를 갱신하는 방법을 기술한다.

새로운 그룹키를 생성하는 다항식 h(x)는 BS에 의해 생성되며 SN 간에 공유하는 그룹키 생성 다항식과는 별도로



(그림 4) 클러스터헤더와 센서 노드 간의 그룹키 갱신 절차

존재한다. 또한 SN 간의 그룹키 생성을 위한 다항식과는 달리 차수의 제한이 없다. 현재 세션의 CH간 그룹키를 h(c)라 할 때 필요에 따라 그룹키를 갱신하게 되는데 특정CH가 공격당하여 오염되는 경우 해당 CH가 새로운 그룹키를 받을 수 없도록 하기 위해 아래와 같은 식에 의해 H(r)과 w(x)를 생성한다. w_i는 공격받은 CH의 아이디로서 BS에 의해 H(r)을 받더라도 공격받은 CH가 자신의 아이디를 대입하는 경우에 계산에 의해 H(r)이 0이 되기 때문에 그룹키를 얻어낼 수 없다. 그렇지 않은 CH는 받은 H(r)을 w(x)로 나누어 실제 새로운 CH간 그룹키 h(r)값을 얻어 다음 그룹 세션에 참가할 수 있게 된다.

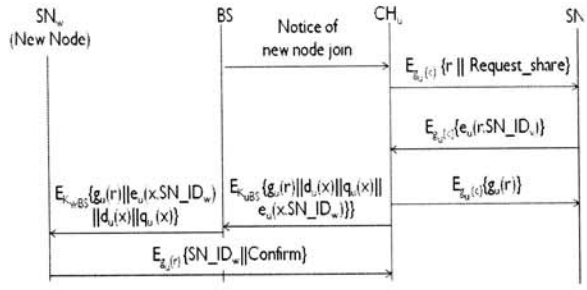
$H(r) = h(r) \times \{(x-w_1)(x-w_2)..(x-w_j)\}$
(단, j는 전체 CH의 개수)

이와 같이 계산된 정보를 현 단계의 CH간 그룹키 h(c)로 암호화해서 브로드캐스트하면 정상적인 CH_u는 새로운 그룹키를 계산할 수 있다.

$BS \Rightarrow CH_u$: $E_{h(c)}\{H(r)||w(x)\}$
($1 \le u \le j$, j는 전체 CH의 개수)
$h(r)=H(r)/w(CH\_ID_u)$

3.2.3 그룹 내에 새 노드 추가

네트워크 안에 새로운 노드가 추가될 경우에는 역방향 비밀성을 위해 그룹키를 갱신해야 할 필요가 있다. 따라서 CH는 새로운 노드가 추가된다는 알림을 받거나 새로이 멤버로 들어온 SN으로부터 가입요청 메세지를 받으면 그룹키 갱신 과정을 수행하게 되는데 이 때 새로 들어온 노드의 경우 현재의 그룹키를 알지 못하므로 앞에서 기술한 키 갱신 과정을 통해 새로운 그룹키를 받을 수 없다. 새로운 노드에게 그룹키를 전달하기 위해서는 BS의 개입이 필요하다. 이를 위해 CH는 기존 노드들에게 부분 정보를 요청하는 메시지를 보내어 암호화 다항식을 복구하며 CH와 센서 노드 간의 그룹키 갱신 과정을 거쳐 새 노드를 위한 그룹키와 부분 정보를 생성하고 해당 정보를 BS와 CH간의 공유키로 암호화하여 BS로 보낸다. BS에서는 새 노드에게 CH로부터 받은 정보를 BS와 새로 투입되는 SN 간의 공유키로 암호화하여 보내주어 그룹 안의 통신에 참여할 수 있도록 한다. 이 과정이 (그림 5)에 나타나 있다.



(그림 5) 그룹 내에 새 노드가 추가되는 경우 새로운 노드를 포함한 그룹 내의 그룹키 갱신 절차

3.2.4 그룹 내에 기존 노드 제거

센서 노드는 고장이나 이동성과 같은 여러 가지 이유로 그룹에서 제거되며 이 경우 순방향 비밀성을 위해서 그룹키를 갱신해야 한다. BS가 이러한 상황을 CH_u에게 공지하는 경우, 또는 일정 기간 동안 SN으로부터 메시지가 오지 않는 경우 노드에 문제가 생겼다고 판단하여 키 갱신을 수행한다. 단, 노드가 네트워크 필드에서 안전하게 제거되었다고 확신할 수 있는 경우에는 3.3.2에서와 같이 그룹키 갱신 과정을 수행한다. 이때 만일 t+1개의 부분 정보를 수집할 수 없는 경우 t차 다항식에 의해 계산되는 새로운 그룹키 계산이 불가능하므로 그룹키 다항식과 암호화 다항식을 새로이 생성하여 그룹키를 갱신한다. 그러나 만일 노드가 오염되었을 가능성이 있는 경우는 3.3.2에서와 같이 현 단계의 그룹키를 이용하여 브로드캐스트하면 다음 단계의 그룹키를 알게 되어 이를 악용할 소지가 있으므로 이 경우는 3.3.2(2)의 CH간 그룹키 갱신의 방법을 적용한다. 즉, 오염된 SN의 아이디를 입력하면 0이 되도록 다항식 $w_u(x)$를 생성하고, 이를 통해 다음 세션의 그룹키를 계산할 수 있도록 $I_u(r)$을 생성하여 $w_u(x)$와 $I_u(r)$을 현재 세션의 그룹키 $g_u(c)$를 이용하여 암호화하여 전송함으로써 오염된 SN이 새로운 세션의 그룹키를 알 수 없도록 한다.

$I_u(r) = i_u(r) \times \{(x-s_1)\ (x-s_2)..(x-s_n)\}$
(단, n은 그룹 내의 SN의 개수)

이와 같이 계산된 정보를 현 단계의 SN간 그룹키 $g_u(c)$로 암호화해서 브로드캐스트하여 정상적인 SN들만이 새로운 그룹키를 계산할 수 있다.

$CH_u \Rightarrow SN_v$: $E_{gu(c)}\{I_u(r)||w_u(x)\}$
$(1 \le v \le n,\ n$는 그룹 내 SN의 수)
$i_u(r)=I_u(r)/w_u(SN\_ID_u)$

4. 성능 평가

본 장에서는 계층적인 네트워크 안에서 안전하게 키를 분배하는 방법에 대해서 기존의 PCGR및 Blundo의 방법과 제안 기법을 비교하여 성능을 평가하기 위하여 시뮬레이션 환경 및 시나리오를 설명하고 기존의 기법들과 에너지 효율성과 안전성에 대해 비교 분석하였다. 특히, 본 연구는 다항식을 기반으로 하여 키를 갱신하는 방법을 제안한 것으로 대표적인 다항식 기반의 두 연구와 비교 분석함으로써 제안 기법의 효율성을 증명한다.

4.1 시뮬레이션 환경

제안한 기법에서는 그룹키를 갱신할 때에 상호 인증을 수행함으로써 CH와 센서 노드 모두 안전하게 그룹키를 갖게 된다. 이 때 미치는 에너지 소모나 통신 오버헤드를 분석함으로써 제안한 기법의 효율성을 평가할 수 있다. 본 논문에서는 효율적인 그룹키 관리 기법을 증명하기 위해서 C++언어와 ZigBee 프로토콜을 사용하는 QualNet 4.5 버전의 센서 네트워크 라이브러리를 추가하여 프로토콜 추가 및 수정을 통해 실험하였으며 에너지 효율성 및 키 갱신 시간, 보안성을 측정하였다.

분석 시에 사용하는 표기법은 <표 2>와 같다.

〈표 2〉 분석에서 사용하는 표기

표기	설명
N	전체 네트워크의 노드 개수
n	그룹 안의 노드 개수
m	이벤트 메시지 길이
L	부분정보 길이
l	키 길이
t	다항식 차수

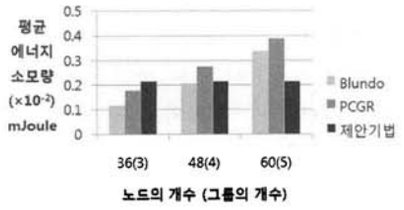
4.2 시뮬레이션 결과

4.2.1 에너지 효율성 분석

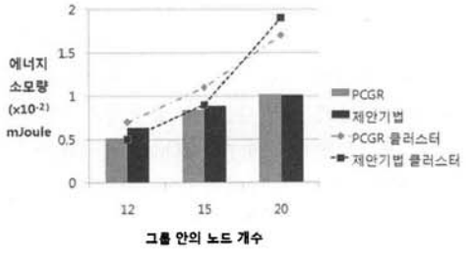
본 절에서는 제안된 기법과 기존의 기법들의 에너지 효율성에 대한 실험 결과를 보인다.

제안된 기법은 (그림 6)에서 보여지는 것처럼 그룹의 개수나 그룹키를 나눠주는 CH의 개수와 상관없이 일정한 에너지 소모량을 보였으나 그에 반해 Blundo는 노드수가 증가할수록 각 노드가 개별적으로 다항식을 갖고 그룹키를 계산하므로 에너지 소모량이 급격히 증가하였으며 PCGR은 그룹 내에 그룹키를 나눠주는 노드가 증가할 경우 에너지 소모량이 증가하는데 이는 PCGR 이 CH의 개념 자체를 가지고 있지 않아 임의의 노드들이 그룹키를 생성하는 과정을 수행해야하는데 그 수가 증가함으로인해 에너지 소모가 급증하는 것으로 분석된다. 제안하는 기법에서는 사전에 일반 멤버 노드와 CH 노드를 분류하고 CH 노드의 경우 보유 에너지나 저장능력 등에 있어서 일반 노드보다 뛰어나다고 가정하였기 때문에 평균적인 에너지 소모량에 있어서 변화가 없다는 장점을 갖는다.

(그림 7)에서는 각 그룹 내의 평균 노드 수를 증가시키면서 전체적인 에너지 소모량과 CH 역할을 하는 노드의 에너지 소모량을 비교 분석하였다. 그림에서 보는 것처럼 그룹 안의 노드 개수가 많아질 경우 PCGR과 제안 기법이 동일하게 각 노드와 CH의 평균적인 에너지 소모는 소폭으로 증



(그림 6) 제안 기법과 기존의 기법의 에너지 효율성

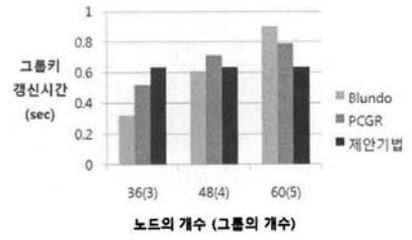


(그림 7) 그룹 내 노드에 따른 노드와 CH의 에너지 효율성

가하는 추세를 보인다. 그러나 PCGR에 비해 제안 기법의 노드들의 평균적인 에너지 소모량의 증가세가 좀 더 작은 반면 CH의 소모량은 멤버 노드의 수가 증가할수록 제안 기법이 좀 더 증가하는 성향을 보인다. 그러나 제안 기법에서는CH는 일반 센서 노드보다 에너지나 계산 능력이 높다고 가정하므로 시스템 성능에 저하를 가져오지 않는다고 본다. 오히려 PCGR의 경우 모든 노드가 동일한 사양을 가지므로 그림에서와 같이 일부 노드의 에너지 소비가 급증할 경우 해당 노드의 수명이 짧아져 결국 전체 네트워크의 수명에까지 영향을 미칠 수 있다. Blundo 기법의 경우 그룹키를 생성하는 노드가 따로 정해져있지 않기 때문에 분석에서 제외되어 있다. 위와 같은 실험 결과, 제안된 기법이 네트워크를 확장할 때에 기존의 기법보다 더 효율적임을 알 수 있다.

4.2.2 키 갱신 시간 분석

본 절에서는 기존의 방법들과 제안된 방법과의 그룹키 갱신 시간을 비교해 보았다. 실험은 그룹의 개수가 3개에서 4개, 5개까지 증가되며 각 그룹 내에 노드의 수는 12개로 고정시켰을 때, 즉, 네트워크의 규모가 커지는 상황에서 그룹키 갱신이 얼마나 빨리 이루어지는지 분석하기 위함이다. 실험 결과 (그림 8)에서처럼 Blundo 기법의 경우는 갱신시간이 큰 폭으로 상승하고 PCGR 기법의 경우에도 계속 증가하는 것을 볼 수 있다. 이는 노드의 개수가 증가할수록 그룹키를 갱신하는 노드들도 증가하며 모든 노드들이 이웃 노드에게 자신의 정보를 주고 또한 이웃 노드로부터 부분 정보를 받아 그룹키 계산을 해야하기 때문에 노드의 증가에 따라 키 갱신 시간도 증가할 수 밖에 없다. 제안 기법의 경우 CH에서 주도적으로 주변 노드에게 정보를 요청하고 주변 노드는 이에 대해 부분 정보를 보낸 후 CH로부터의 그룹키를 재검증 해야하기 때문에 노드 수가 적은 경우에는 다른 기법에 비해 시간이 조금 더 걸리지만 그룹 및 노드 수가 증가하는 경우에도 전체 그룹키 갱신 시간에 변화가



(그림 8) 기존의 기법과 제안된 기법의 그룹키 갱신 시간 비교

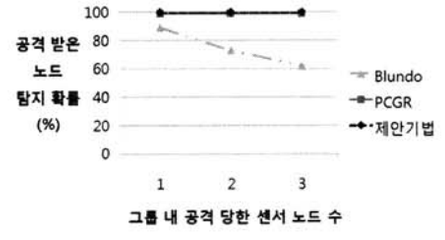
거의 없어 네트워크의 규모가 커지는 경우 더욱 효과적으로 적용될 수 있음을 알 수 있다.

또한, 센서 노드가 추가되거나 제거되는 경우에도 기존의 키 갱신 시의 오버헤드 외에 특별히 필요한 시간, 통신 상의 오버헤드는 필요치 않다. 즉, 노드 추가 시에는 기존의 키 갱신 시와 마찬가지로 CH의 요청에 의해 키 부분 정보를 전달하고 새로 받은 그룹키를 인증 방식에 의해 확인하고, 노드 제거 시에는 CH가 보내준 정보를 가지고 단순히 계산만 수행함으로써 새로운 그룹키를 가질 수 있다.

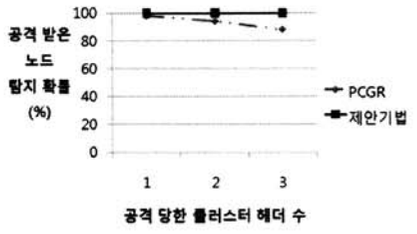
4.2.3 보안성 분석

본 절에서는 그룹 내에 있는 센서 노드가 공격당했을 경우를 감지하는 정도를 알아보았다. 실험은 그룹의 개수가 3개이고 그룹 내에 노드의 수는 12개로 고정시키고 각 클러스터 별로 센서 노드가 공격 당했을 때와 CH가 공격 당했을 때 각 기법이 이를 감지해낼 수 있는지 분석하였다. 먼저 일반 센서 노드가 공격 당한 경우 Blundo 기법은 중앙에서 모든 노드에게 그룹키를 생성하는 다항식을 분배하기 때문에 센서 노드가 오염되는 경우에 사실상 공격여부를 감지하는 것이 불가능하다. 더 큰 문제는 노드가 오염되는 경우 다항식 자체가 공격자에게 노출되어 모든 그룹키를 계산해낼 수 있기 때문에 네트워크에 미치는 영향은 더욱 심각하다. 반면, PCGR이나 제안 기법의 경우 중간에 부분 정보를 주고받는 과정에서 그룹키를 갱신하는 노드가 검증을 하면서 검증에 실패한 노드를 걸러낼 수 있다는 장점이 있다. (그림 9)는 센서 노드가 오염된 경우 이를 탐지해낼 수 있는지를 비교한 그림으로 Blundo방식이 그룹키 갱신을 한다고 해도 주고받는 값을 검증하는 과정이 없어 전혀 탐지가 불가능한 반면 PCGR과 제안 기법의 경우 이를 상당 수준 탐지할 수 있다.

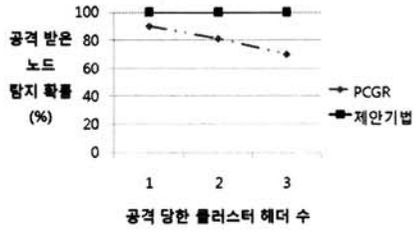
이 때, 기존의 PCGR 기법과 제안 기법의 차이를 볼 수 있는 부분은 PCGR의 경우 그룹키를 갱신하는 노드가 주변



(그림 9) 일반 센서 노드가 공격당했을 경우 이를 감지할 확률



(그림 10) CH가 키 정보를 줄 경우 공격을 감지할 확률



(그림 11) CH가 키 정보를 주지 않을 경우, 또는 감지를 피하기 위해 정확한 값을 주는 경우 공격을 감지할 확률

노드로부터 부분정보를 받아 검증을 수행하기 때문에 주변 노드의 오염여부를 알 수 있으나 그룹키 계산 후 이 정보를 주변 노드에게 보내면 주변 노드는 이 정보를 검증 없이 받아들인다. 그렇기 때문에 그룹키 갱신 노드가 오염된 경우 이를 감지할 수 없다는 단점을 갖는다. 물론 이 경우에도 PCGR의 경우 그룹키를 갱신하는 노드가 따로 정해져 있지 않고 모든 노드가 갱신의 주체가 될 수 있으므로 오염된 노드가 그룹키 계산을 할 뿐 아니라 이웃 노드의 부분정보 요청에 답을 할 수도 있어 이 때에 감지가 가능할 수 있다. 그러나 만일 감지를 피하기 위해 부분 정보를 보내지 않거나 또는 해당 정보만을 의도적으로 정확히 계산해서 보낸다고 하면 사실상 이를 감지하는 것은 불가능하다. 이를 보여주는 결과가 (그림 10, 11)이다. 즉, (그림 10)의 경우는 PCGR에서 오염된 키 갱신 노드가 주변의 또 다른 키 갱신 노드의 부분 정보 요청에 답을 하는 경우 검증과정을 통해 감지가 되는 경우이고, (그림 11)의 경우는 키 요청에 답을 하지 않거나 감지를 피하기 위해 일부러 정확한 정보를 보내는 경우 오염 여부를 감지할 확률이 현저히 떨어지는 것을 볼 수 있다. 반면 제안 기법의 경우는 CH, 즉, 키 갱신을 수행하는 노드가 키를 새로이 계산하여 주변 노드에게 보내는 경우 주변 노드에 의해 한번 더 검증이 되므로 오염된 CH를 100% 가까이 감지해 낼 수 있다는 장점을 갖는다.

앞에서 기술한 바와 같이 Blundo기법은 BS에서 전체 노드에게 다항식 자체를 배분하는 것이기 때문에 그룹키를 갱신하는 노드가 따로 없으므로 해당 노드가 감염되었을 경우에 대한 실험은 필요 없으므로 마지막 실험에서는 제외하였다.

4.2.4 저장 및 통신 오버헤드 분석

Blundo 기법의 경우 전체 네트워크 노드의 개수와 그룹의 개수에 따라 저장 오버헤드가 좌우된다. 그룹키를 갱신하기 위한 다항식이 t차라고 했을 때, 전체 노드의 개수가 N이라고 하면 BS의 경우 각 그룹별 다항식을 저장해야하므로 N/n개의 다항식을 저장하고 각 노드의 경우 자신에게 전송된 하나의 다항식을 저장한다. 즉, 다항식을 계수로 저장한다고 했을 때 다항식 각 계수의 길이를 L이라 하면 각 노드는 L*(t+1)의 저장공간을 필요로 하므로BS와 각 노드에서 필요한 저장 공간을 고려한 총 저장 오버헤드는 N/n*L*(t+1)+N*L*(t+1)= N*L*(t+1)*(1+1/n)이다.

PCGR 기법의 경우, 그룹 내 각 노드가 g'(x)를 보유하는데 필요한 공간이 L*(t+1), 주변의 키 갱신 노드들로부터 받은 부분 정보 저장에 n*L*(t+1), 전체 노드수가 N이므로 총 저장 오버헤드는 N*L*(t+1)(n+1)이다.

반면 제안된 기법의 경우, CH가 사전에 구분되어 있어 CH만이 g'(x)를 저장하므로 여기에 (t+1)*L이 필요하고 해당 그룹에 포함된 노드들이 CH로부터 전달받은 부분정보를 저장해야하므로 n*(t+1)*L, 전체 그룹 수가 N/(n+1)이므로 총 저장공간은 N*L*(t+1)이된다. 즉, PCGR에 비해 훨씬 정장 공간 오버헤드가 작음을 알 수 있다.

다음으로 통신 오버헤드를 생각해보면 먼저 Blundo 기법의 경우 다항식을 사전에 분배한다면 별도의 통신 오버헤드가 존재하지 않으나 그룹 형성 후 분배한다면BS에서 모든 모드에게 t차 다항식을 전달해야하므로 N*L*(t+1)만큼의 통신오버헤드가 요구된다.

PCGR의 경우 초기화를 위해 각 노드가 주변 노드에게 암호화 다항식을 전송하는 데에 n*L*(t+1)*N/(n+1), 키 갱신 시 각 노드가 각각의 timer에 의해 키 갱신이 필요한 시간이 되면 키 갱신 노드의 주변에 있는 n개의 노드에서 키 갱신 노드로 키의 부분정보값을 전송하는데n*l비트 만큼의 데이터를 전송하며 키 갱신 노드는 주변 n개의 노드에게 계산된 키값을 전송하는데 또 다시n*l 비트가 필요하며 이러한 그룹 수가 N/(n+1)이므로 전체적으로는 약 2*l*N만큼의 통신 오버헤드가 필요하다.

본 논문에서 제안한 기법의 경우 먼저 초기화 시CH가 주변 노드들에게 그룹형성을 위해 보내는 메시지가 n*m비트, 응답에 n*m비트, 암호화 다항식을 전송하는데 n*(t+1)*L가 필요하므로 전체적으로 n*(2m+(t+1)L) 비트가 필요하다. 또한 키 갱신 시 부분 정보를 요청하는데 n*m 만큼의 통신이 필요하고 이에 주변 n개의 노드가 계산한 키의 부분정보를 보내는데 n*l만큼이 필요하며 CH가 새로운 그룹키를 계산한 후 다시 주변 노드들에게 보낼 때 또 다시 n*l만큼의 통신 오버헤드가 필요하여 전체적으로 n*(2l+m)만큼의 통신 오버헤드가 필요하다. m값이 상대적으로 매우 작으므로 통신 상의 오버헤드도 PCGR에 비해 제안 기법이 훨씬 작음을 알 수 있다. <표 3>에 오버헤드에 관해 비교하고 정리하였다.

〈표 3〉 기존의 기법과 제안 기법의 오버헤드 비교

	저장 오버헤드	통신(초기화) 오버헤드	통신(키갱신) 오버헤드
Blundo	L*N*(t+1)(1+1/n) bit	N*L*(t+1)	
PCGR	L*N*(n+1)(t+1) bit	n*N*L*(t+1) /(n+1)	2*l*N
제안 기법	L*N*(t+1) bit	n*(2m+(t+1)*L)	n*(2l+m)

5. 결 론

본 논문에서는 이종의 센서로 구성된 네트워크 환경에서의 상호 인증을 통한 안전하고 효율적인 그룹키 관리 기법을 제안하였다.

센서 네트워크는 노드 자원의 제약과 제한된 에너지 및 메모리 공간, 낮은 연산 능력 등 여러 제약 사항을 고려하여야 한다. 이와 같은 특징을 고려하여 계산양이 많은 CH를 일반 노드보다 메모리나 계산 능력이 충분한 장비로 두고 일반 노드의 계산 량은 최소한으로 하도록 하여 에너지 효율적으로 사용함으로써 네트워크의 수명을 연장시킬 수 있도록 하였으며 그룹키를 갱신할 때 CH와 센서 노드 사이에 주고받는 정보를 이용하여 엔티티를 검증할 수 있도록 함으로써 이를 통해 네트워크상에서의 안전한 통신을 할 수 있는 그룹키 관리 기법을 제안하였다. 또한 기존의 PCGR기법을 기반으로 하여 그룹키를 갱신할 때에 CH는 노드가 주는 부분 정보를 다항식의 특성을 이용하여 검증하고 그룹키를 재생성하며 노드 또한 생성된 그룹키를 받아서 자신이 받은 그룹키가 올바른 그룹키인지 검증한다. 그룹키를 갱신할 때에 시간 동기화 방법이 아닌 메시지를 통해 요청하므로 동기화에 필요한 추가적인 비용이 들지 않는다. 뿐만 아니라 단방향 인증만을 시도했던 기존 PCGR에 반해 상호 인증을 수행함으로써 네트워크의 안전성을 극대화 시켰다. 특히 그룹키를 필요에 따라 갱신함으로써 네트워크 안에서의 통신을 안전하게 하는 것은 물론, 순방향 비밀성과 역방향 비밀성을 위해 노드가 추가되거나 제거될 때에도 그룹키를 갱신한다.

제안된 기법은 QualNet 시뮬레이터를 통해 Blundo 기법과 PCGR 기법과의 실험 및 분석을 하였다. 이를 통해 제안된 기법이 Blundo나 PCGR에 비해서 에너지 효율성이나 키 갱신 시간 면에서 오버헤드가 많이 증가되지 않으면서도 보안성을 높여 안전한 통신이 가능한 네트워크 환경을 제공할 수 있음을 보였다. 또한 저장이나 통신 상의 오버헤드도 더 낮음을 알 수 있다.

앞으로 향후 연구 일정에서는 노드가 이동할 때에 기존에 가지고 있는 정보를 이용하여 최소한의 통신으로 그룹에 추가되거나 제거될 수 있도록 하는 방안에 대해 연구를 진행하여 볼 것이다.

참 고 문 헌

[1] Carlo Blundo, Alfredo De Santis, Amir Herzbeerg, Shay Kutten, Ugo Vaccaro, Moti Yung, "Perfectly-Secure Key Distribution for Dynamic Conferences," Information and Computation, 1995.

[2] Yong Wang, Byrav Ramamurthy, Yuyan Xue, "Group Rekeying Schemes for Secure Group Communication in Wireless Sensor Networks," IEEE International Conference on Communications, 2007.

[3] Wensheng Zhang, Guohogn Cao, "Group Rekeying for Filtering False Data in Sensor Networks: A Predistribution and Local Collaboratio-Based Approach," IEEE INFOCOM, 2005.

[4] Jyh-How Huang, Jason Buckingham, Richard Han, "A Level Key Infrastructure for Secure and Efficient Group Communication in Wireless Sensor Networks," International Conference on Security and Privacy for Energing Areas in Communications Networks, 2005.

[5] Sencun Zhu, Sanjeev Setia, Sushil Jahodia, "LEAP+: Efficient Security Mechanisms for Large-Scale Disributed Sensor Networks," ACM Transactions on Sensor Networks, 2006.

[6] Zhen Yu, Yong Guan, "A Robust Group-based Key Management Scheme for Wireless Sensor Networks," IEEE Communications Society, 2005.

[7] 허재두, 최은창, 김동균, "센서 네트워크 응용 기술 동향", 정보통신연구진흥원 주간기술동향 1367호, 2008.

[8] 김지은, 김세한, 정운철, 김내수, "USN 센서노드 기술 동향", 전자통신동향분석, 제 22권 제 3호, 2007년 6월.

고 혜 영

e-mail : popo1211@neowiz.com

2007년 한국성서대학교 인터넷정보학과(학사)

2010년 이화여자대학교 컴퓨터공학과(석사)

2010년~현 재 NEOWIZ GAMES 재직

관심분야 : 정보보안, 네트워크 보안, USN

도 인 실

e-mail : isdoh@ewhain.net

1993년, 1995년 이화여자대학교 전자계산학과(학사, 석사)

1995년~1998년 삼성 SDS

2002년~2007년 이화여자대학교 컴퓨터공학과(박사)

2007년~2008년 서울대학교 박사후연구원

2008년~현 재 이화여자대학교 컴퓨터공학과 연구교수

관심분야 : 네트워크 보안, 무선통신망, 센서네트워크 보안, 홈네트워크 보안

채 기 준

e-mail : kjchae@ewha.ac.kr

1982년 연세대학교 수학과(학사)

1984년 미국 Syracuse University 컴퓨터학과(석사)

1990년 미국 NorthCarolina State University 컴퓨터공학과(박사)

1990년~1992년 미국 해군사관학교 컴퓨터학과 조교수

1992년~현 재 이화여자대학교 컴퓨터공학과 교수

관심분야 : 네트워크 보안, 센서 네트워크, 네트워크 프로토콜 설계 및 성능분석