Journal of Korea Multimedia Society Vol. 11, No. 5, May 2008(pp. 661-672)

이동 애드혹 네트워크에서 신뢰성 있는 클러스터 기반 동적 인증 기법

황윤철†, 김진일††

요 약

이동 애드혹 네트워크는 이동 호스트들만으로 이루어진 자율적인 네트워크로 그 활용 범위가 점차 다양한 분야로 확대되어 활발한 연구가 진행되고 있다. 그러나 이동 애드혹 네트워크는 무선 링크의 취약성과 중심이 되는 기반 구조의 부재로 인해 모든 네트워킹을 구성하고 있는 노드들의 협력에 의존하고 있어 일반 네트워크 보다 보안 측면에서 훨씬 심각한 위험에 노출되어 있다. 따라서 본 논문에서는 클러스터 기반으로 구성되는 계층적 이동 애드혹 네트워크 구조를 사용하여 중심이 되는 기반 구조의 부재를 해결하고 각 계층 노드 간 상호인증을 통하여 보안 취약점을 개선하여 신뢰할 수 있는 노드들만이 통신에 참여할 수 있도록 하는 클러스터 기반 동적 인증 기법을 제안한다. 제안된 기법이 이동 애드혹 네트워크의 보안 취약점을 보완하고 신뢰성이나 확장성 측면에 있어서 기존의 기법보다 우수함을 시뮬레이션을 통해 확인하였다.

A Reliable Cluster based Dynamic Authentication Mechanism in MANET

Yoon-Cheol Hwang†, Jin-Il Kim††

ABSTRACT

Mobile Ad-hoc NETwork is a kind of self-controlled network composed only of mobile hosts. Since its range of use is gradually expanding into various sections applicable to practical lives, active researches are being conducted on it. However, as it depends on cooperation of nodes composing the entire network, due to weakness of wireless link and lack of its central infrastructure, so it is exposed to more serious risk than general network in security. Therefore, this paper proposes Cluster-Based Dynamic Authentication that enables only reliable nodes to participate in communication, by solving lack of centralized infrastructure, using hierarchical Mobile Ad hoc NETwork structure based on cluster, and by complementing security weakness through mutual authentication between hierarchical nodes. Simulation shows that the proposed scheme can complement security weakness of Mobile Ad hoc NETwork and that it is more adequate in reliability and expandability than the existing schemes.

Key words: MANET(이동 애드혹 네트워크), Cluster-based(클러스터 기반), Secret Share(비밀공유), Reliability(신뢰성), Dynamic Authentication(동적 인증)

1. 서 론

이동 애드혹 네트워크(Mobile Ad-hoc NETwork, MANET)는 기존에 존재하는 유선 네트워크나 기간망을 사용하지 않고 이동 호스트만으로 이루어진 자율적인 네트워크이다[1]. 이동 애드혹 네트워크는 기

※ 교신저자(Corresponding Author) : 김진일, 주소 : 대전광역시 서구 연자 1길 14(도마동)(302-735), 전화 : 042)520-5935, FAX : 042)520-5533, E-mail : comclass@pcu.ac.kr

접수일 : 2007년 11월 1일, 완료일 : 2008년 3월 25일

† 정회원, 배재대학교 교양교육지원센터 강사
(E-mail : dolpin98@paran.com)
†† 정회원, 배재대학교 교양교육지원센터 IT분야 교수

존 기간망을 사용하는 네트워크와는 달리 임시적이고 즉흥적으로 네트워크를 구성하는 데 매우 용이한 특성 때문에 유선 네트워크로 구성하기 어려운 환경에 적합한 것으로 인식되어, 주로 재해/재난 지역과 전쟁 지역과 같은 특수한 목적으로 활용되었다[2]. 그러나 최근에 인터넷의 급속한 성장에 따라서 IP 이동성에 관한 연구가 활발히 진행되면서 이를 지원하기 위한 근거리 망으로 이동 애드혹 네트워크가 주목을 받게 되었다. 특히 유비쿼터스 네트워크로 기대되는 차세대 네트워크로서 무선 사설 망, 블루투스, 홈 네트워크, 센서 네트워크 등 실생활에 적용할 수 있는 이동 네트워크에서 가장 각광받고 있다. 그러나 이동 애드혹 네트워크은 동적 토폴로지의 변화, 중앙의 감시와 관리의 결여, 자원의 제약성, 무선 매체의 사용 등의 고유한 특성 때문에 네트워크를 구성하는 노드들 간의 안전한 통신이 보장되지 않다는 문제점을 가지고 있다. 따라서 이동 애드혹 네트워크가 가지는 고유의 서비스 영역 뿐만 아니라 많은 분야에서 기존의 서비스를 이동 에드혹 네트워크와 부분적으로 접목하려면 통신에 참여하는 노드의 신뢰성을 보장하고 다양한 공격을 방어할 수 있는 보안 알고리즘이 반드시 필요하다[3].

기존의 유·무선 환경에서는 신뢰성을 보장하기 위해서 인증기관(Certificate Authority, CA)을 통한 인증 기법을 사용했지만 이동 애드혹 네트워크에서는 고정된 인프라가 존재하지 않는 환경이기 때문에 이동 노드 간 협력을 통한 분산 인증 기법을 사용한다[4-6]. 그러나 이 기법들은 인증기관의 기능을 수행하는 노드의 수는 제한되어 있고 분산되어 있어 각 클라이언트들은 인증서를 얻기 위해 많은 통신 오버를 유발하고 노드의 신뢰성과 다른 노드로부터 수신한 조각키의 검증도 고려하지 않았다는 단점이 있다.

따라서 본 논문에서는 이동 애드혹 네트워크를 보다 효과적이고 안정적으로 관리할 수 있도록 클러스터를 기반으로하여 계층적으로 구성한다. 그리고 이러한 구조에서 신뢰할 수 있는 노드들만이 통신에 참여할 수 있는 클러스터 기반 동적 인증 기법(Cluster-Based Dynamic Authentication, CBDA)을 제안한다. 제안된 기법은 키 관리자와 클러스터 헤드, 멤버 노드로 구성되는 계층적 구조를 사용한다. 클러스터 헤드는 분산 조합 가중치 알고리즘에 의해 선출되며 선출된 클러스터 헤드 중에서 다시 클러스터 헤드 선출 알고리즘에 의해 키 관리자를 선출한다. 선출된 키 관리자에 의해 클러스터 헤드와 키 관리자간 인증 및 클러스터 헤드 간 인증이 이루어지며 그 다음에 클러스터 헤드와 멤버 노드간의 인증이 실행되는 계층적 인증 절차를 통해 신뢰 관계를 구축한다. 키 관리를 위해서는 비밀키 공유 기법 중에 부분 분산 기법을 적용하고 공개키 및 ID 교환을 위해서는 공개키 암호 기법을 사용하며 인증과 부인봉쇄를 위해서는 전자 서명 기법을 사용한다.

2. 관련 연구

2.1 이동 애드혹 네트워크 보안

초기의 이동 애드혹 네트워크에 대한 연구들은 주로 라우팅 분야에 한정되어 이루어졌지만 최근에는 네트워크 전체에 대한 보안의 중요성이 크게 부각되고 있다. 이동 애드혹 네트워크에 대한 보안은 크게 라우팅, 키 관리, 이기적 노드 그리고 침입 탐지에 대한 연구로 구분할 수 있다. 라우팅에 대한 연구는 최단 경로 보다는 안전한 경로를 제공하는 것을 우선으로 하며 네트워크에 참여하는 일정 수준 이상의 신뢰성을 가지는 노드만을 경로로 삼는 방법과 소스 노드와 목적지 노드 사이에 있는 중간노드에 의한 악의적인 공격을 감지할 수 있는 방법이 있다[1,2]. 키 관리에 대한 연구는 이동 애드혹 네트워크의 특성에 맞는 새로운 키 관리 메커니즘을 개발하는 데 중점을 두고 있으며 몇몇의 특정 노드가 서버 역할을 맡아 인증기관의 능력을 나누어 갖거나 네트워크의 모든 노드가 인증기관의 공개키 정보를 공유하는 방법으로 연구가 진행되고 있다[5]. 이기적인 노드에 대한 연구는 탐지하기가 어렵고 네트워크에 심각한 피해를 주기 때문에 빠르고 정확한 탐지와 관리가 최우선적으로 고려되어야 하며 대부분의 연구가 각각의 노드들이 자기 자신의 이웃 노드를 감시하게 하거나 네트워크에 일정량을 데이터를 전달해 주어야만 자신이 생성한 데이터를 전송할 수 있게 하는 방법에 대한 연구가 이루어지고 있다. 마지막으로 침입 탐지에 대한 연구는 이동 애드혹 네트워크에서는 노드들을 관리하는 기반시설이 없기 때문에 네트워크에 침입이 발생할 경우, 기존의 네트워크에서의 침입 탐지와는 달리 노드들 간의 협업을 통한 침입 탐

지 기법이 연구되고 있다[6].

이동 애드혹 네트워크의 특성상 유선 네트워크에 비해 보안이 많이 취약하며 그 사용 용도상 공격이나 침입이 발생하면 가능한 빨리 검출하고 이에 상응하는 적절한 조치가 이루어져 이동 애드혹 네트워크가 보호되어야 한다. 이를 위한 방법으로 사전에 예방하는 방법과 사후에 조치를 취하는 방법이 있는 데 본 논문에서는 제안하는 기법은 침입을 사전에 예방하여 이동 애드혹 네트워크의 보안을 강화시킨다.

2.2 비밀키 공유 기법

비밀키 공유 기법은 Shamir가 제안한 기법으로 임의의 비밀키를 여러 사용자들에게 나누어 갖게 하여 단일 사용자만으로는 원래의 비밀키를 복원할 수 없도록 하는 기법으로 (k, n) 임계치 암호화 기법이 대표적이다[4,7,8]. 이 기법을 처음으로 이동 애드혹 네트워크에 적용한 방식이 분산 인증기관 기법으로 부분 분산 인증기관 기법과 완전 분산 인증기관 기법으로 분류한다. 부분 분산 인증 기법은 인증기관의 기능을 특정 노드 집합에게 분배하는 방법으로 인증기관의 기능을 수행하는 노드의 수는 제한되어 있고 분산되어 있어서 각 클라이언트들은 인증서를 얻기 위해 많은 통신 오버를 유발한다. 또한, 노드의 신뢰성과 다른 노드로부터 수신한 조각키의 검증도 고려하지 않았다는 단점이 있다. 완전 분산 인증기관 기법은 네트워크내의 모든 노드들이 인증기관의 기능을 수행하도록 하는 기법으로 이동 애드혹 네트워크가 구성된 후 자가 초기화 알고리즘을 사용하여 중앙 관리자 없이 부분 비밀키를 관리할 수 있다. 그러나 이 방법은 모든 노드가 부분 비밀키를 소유하고 있기 때문에 부분 분산 기법에 비해 많은 위험에 노출되어 있고 네트워크 이전에 부분 비밀키를 미리 분배한다는 점에서 네트워크 확장성에 부정적인 영향을 미친다.

3. 제안된 클러스터 기반 동적 인증 기법

본 장에서는 비밀 공유 및 공개키 기법을 이용하여 신뢰관계 구축을 위한 효율적인 클러스터 기반 동적 인증 기법을 제안한다. 이 기법은 신뢰성 있는 통신을 위해 그림 1과 같은 여러 독립적인 클러스터로 구성된 네트워크 모델을 사용한다. 클러스터 기반

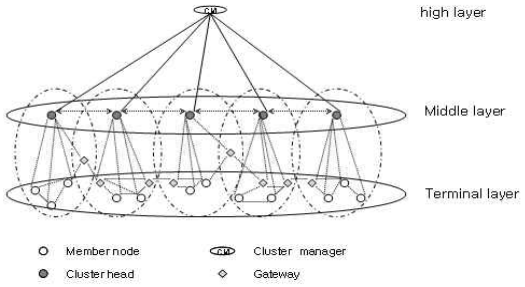

그림 1. 클러스터 기반 MANET의 동적 인증 모델

동적 인증 기법의 안전성은 [9]의 Diffie-Hellman 키 분배 방식을 이용한 이산대수에 근거하고 있으며 노드들의 에너지 소모와 네트워크 오버헤드가 적게 발생되도록 설계한다.

제안된 기법은 클러스터 형성(Cluster Information), 키 생성 및 분배(Key Generation and Deployment), 클러스터 구성 요소 간 키 설립(Establishment Key) 그리고 동적 인증(Dynamic Authentication) 등 4단계로 구성되며, 각 단계의 세부적인 동작 과정은 다음과 같다. 1단계에서는 이동 애드혹 네트워크의 효율성과 안정성을 향상시키기 위해 전체 네트워크를 클러스터로 분할하는 클러스터링을 수행한다. 2 단계는 비밀 분산 기법을 이용하여 부분 비밀키를 생성하고 분배하는 과정이다. 3단계에서는 클러스터를 구성하는 노드들 사이의 키 설립이 이루어지고 4단계에서 생성된 키들을 사용하여 클러스터 구성 요소들 사이에 동적 인증을 수행한다.

3.1 가정 및 기호 정의

제안된 기법은 단일 도메인으로 구성 되며, 이동 애드혹 네트워크 노드 간에 전송되는 메시지는 데이터의 무결성과 기밀성이 보장된다. 그리고 악의적인 내부 공격의 가능성을 배제하는 것으로 가정한다. 본 논문에서 계층적 신뢰 관계 구축을 위한 인증 기법에 사용되는 주요 파라미터에 대한 표기법은 표 1과 같다.

3.2 클러스터 형성

본 논문에서는 이동 애드혹 네트워크에 참여하는 노드들을 클러스터로 구성할 때 [10,11]의 클러스터링 알고리즘을 개선한 조합 가중치 분산클러스

표 1. 주요 파라미터

표기법	의 미
CM	클러스터 관리자
CH	클러스터 헤드의 집합
CH_x	x번째 클러스터의 헤드
M	멤버 노드의 집합
M_x	멤버노드 x
K_{xy}	x와 y 사이의 공유키
sm_x	x의 부분 비밀키
id_x	x의 ID
GK_x	x번째 클러스터의 그룹키
P_x	x의 공개키
S_x	x의 비밀키
→	유니캐스트
↔	양방향 통신
||	연결
$\{M\}_k$	키 k로 M을 암호화

터링 알고리즘을 사용한다. 그리고 초기 클러스터 형성 과정에서 발생하는 초기 설정 오버헤드를 줄이기 위해 클러스터 관리자를 선출할 때는 네트워크의 전체 노드들에 대한 최소 가중치를 계산하는 전역 최소치(global minima)방식을 사용하고, 클러스터 헤드를 선출할 때는 지역 최소치(local minima) 방식을 사용한다. 클러스터 헤드는 각 클러스터를 대표하여 독립적으로 관리하고 네트워크내의 모든 클러스터 헤드는 주소 자원 및 경로정보를 공유한다. 각 노드는 네트워크에 참여하기 전에 각각 고유의 IP 주소를 할당 받았다고 가정하며, 본 논문에서는 이를 노드의 ID로 사용한다.

3.3 키 생성 및 분배

키 생성 및 분배 과정은 클러스터 형성 후에 다음과 같이 7단계로 구성된다.

1단계 : 선출된 클러스터 관리자는 클러스터 관리자와 클러스터 헤드 사이의 공유키 K_{CM-CH}를 생성하기 위해 $(T,\ t)$ 임계치 암호화에 의해 s_i개의 공유정보를 클러스터 헤드에게 나누어주고, $r_i \in Z_p$개의 랜덤 값을 선택한 다음 공유키 K_{CM-CH}을 다음과 같이 생성한다. 단 P_{CH_i}는 CH_i의 공개키이다.

$$K_{CM-CH} = (P_{ch_i})^{r_i},\ \text{i= 1,...,}s_i \qquad \text{(식 1)}$$

2단계 : 클러스터 헤드의 비밀키는 각 n개의 멤버 노드들에게 분산시키기 위해 $(S,\ n)$ 임계치 암호화 기법[4]을 사용한다.

3단계 : 비밀키 분산을 위해 클러스터 헤드는 식(2)과 같이 차수가 $k-1$인 다항식을 선택하며, n개의 비밀 부분키(S_i)를 식(1)에 의해 생성한다.

$$f(x) = a_{k-1}x^{k-1} + a_{k-2}x^{k-2} + ... + a_1x + a_0 (\mathrm{mod}\, p) \qquad \text{(식 2)}$$

여기서 a_0=S이고 a_i, i=1,2....,k-1는 Z_p로부터 임의로 선택한 값이다.

4단계 : 부분 비밀키를 분배하기 전에 공유하는 다항식의 공계수에 대한 증거인 $g^{a_0}g^{a_1}...,g^{a_{k-1}}$와 클러스터 헤드가 생성한 그룹 키 $GK_{ch_i}(1<i<\#CH)$을 멤버 노드들에게 알린다.

$$S_{m_i} = f(id_i) \bmod p \quad (S_{m_i} i = 1,2,3,.....,n) \qquad \text{(식 3)}$$

5단계 : 생성한 부분 비밀키를 각각의 비밀 부분키 소유자인 멤버 노드(id_1,id_2,.....)들에게 분배한다. 비밀 부분키와 다항식의 계수는 클러스터 헤드만 알고 있다.

6단계 : 부분 비밀키를 수신한 멤버 노드들은 이 부분 비밀키 값의 유효성을 (식 4)와 같이 계산하여 검증한다.

$$(g^{a_{k-1}})^{id_i^{k-1}} \cdot (g^{a_{k-1}})^{id_i^{k-2}} .....,(g^{a_1})^{id_i} \cdot g^{a_0} = g^{S_i} \qquad \text{(식 4)}$$

7단계 : 검증이 끝난 멤버 노드들은 클러스터 헤드를 통해 다른 클러스터 멤버들에게 해당 노드의 부분 비밀키를 요청하여 k개 이상의 부분키를 얻으면 (식 5)를 통해 원래의 다항식 $f(x)$를 복원하여 같은 클러스터 내의 노드들과 통신하는데 있어 세션키로 사용한다.

$$f(x) = \sum_{i=1}^{k} S_i \times l_{id_i}(x) (\mathrm{mod}\, p),\ \ l_{id_i}(x) = \prod_{i=1 j \neq i}^{k} \frac{x - id_j}{id_i - id_j} \qquad \text{(식 5)}$$

3.4 클러스터 구성 요소 간 키 설립

클러스터 구성 요소 간의 키 설정은 클러스터 관리자와 클러스터 헤드 사이의 키 설정, 클러스터 헤드 사이의 키 설정 그리고 클러스터 헤드와 멤버 노드 간 키 설정으로 구분하여 동작 과정과 특징을 살펴본다.

3.4.1 클러스터 관리자와 클러스터 헤드 간 키 설립

클러스터 관리자와 클러스터 헤드는 공개키와 비밀키를 생성하여 공개키는 네트워크에 공개한다. 그런 다음 클러스터 헤드의 등록 요청이 들어오면 다음과 같은 절차를 거쳐 클러스터 관리자와 클러스터 헤드간 키 설정을 이루어진다.

1단계 CH → CM : $\{id_{ch}||id_{cm}||GK_{ch}||P_{ch}||RV_1||\{id_{ch}\}_{s_{ch}}\}_{p_{cm}}$

2단계 CM → CH : $\{id_{cm}||id_{ch}||RV_1||(k_{cm-ch_j}||id_{cm}\}_{s_{cm}}\}_{p_{ch}}$

3.4.2 클러스터 헤드 간 키 설립

각 클러스터 헤드는 다른 클러스터 헤드와의 통신을 위해서는 다음과 같이 키 설정 과정을 수행한다. RV_2는 클러스터 CH_i가 생성한 랜덤수이며 클러스터 간 공유키 $K_{CH_i-CH_j}$를 생성하기 위해 CH_j의 공개키로 RV_2와 id_{ch_i}를 암호화한다. step 5에서는 CH_j로부터 전달받은 $RV_2^{'}$를 RV_2와 비교하여 일치할 경우 클러스터 간 공유키 $K_{CH_i-CH_j}$를 사용하여 $data$를 전달한다.

1단계 CH_i→CM: $\{id_{ch_i}||id_{ch_j}||RV_2||\{id_{ch_i}\}_{s_{ch_i}}||\{RV_2||\{id_{ch_i}\}_{s_{ch_i}}||id_{ch_i}\}_{p_{ch_j}}\}_{K_{CM-CH_i}}$

2단계 CM→CH_j: $\{id_{cm}||id_{ch_j}||RV_3||\{id_{cm}\}_{s_{cm}}||p_{ch_i}||\{RV_2||\{id_{ch_i}\}_{s_{ch_i}}||id_{ch_i}\}_{p_{ch_j}}\}_{K_{CM-CH_j}}$

3단계 CH_j→CM : $\{id_{ch_j}||id_{cm}||RV_3||\{id_{ch_j}\}_{s_{ch_j}}||\{RV_4||id_{ch_i}||\{id_{ch_j}\}_{s_{ch_j}}\}_{p_{ch_j}}\}_{K_{CM-CH_j}}$

4단계 CM→CH_i: $\{id_{cm}||id_{ch_i}||RV_2||\{id_{cm}\}_{s_{cm}}||p_{ch_j}||\{RV_4||id_{ch_i}||\{id_{ch_j}\}_{s_{ch_j}}\}_{p_{ch_j}}\}_{K_{CM-CH_i}}$

5단계 CH_i↔CH_j: $\{id_{ch_i}||id_{ch_j}||\{id_{ch_i}\}_{s_{ch_i}}||\{id_{ch_j}\}_{s_{ch_j}}||RV_4||data\}_{K_{CH_i-CH_j}}$

3.4.3 클러스터 헤드와 멤버 노드 간 키 설립

각 클러스터에 소속된 멤버노드들은 해당 클러스터 헤드에게 멤버로서 등록하기 위해 아래와 같은 키 설정 과정을 수행하여 그룹키를 획득한다.

1단계 CH_i → M : request public key $\{id_{ch_i}||\{id_{ch_i}\}_{s_{ch_i}}||p_{ch_i}\}$

2단계 M → CH_i : $\{id_m||id_{ch_j}||p_m||RV_5||\{id_m\}_{s_m}\}_{p_{ch_i}}$

3단계 CH_i → M: $\{id_{ch_i}||id_m||RV_5||\{id_{ch_i}\}_{s_{ch_i}}||GK_{ch_i}||sm_m\}_{p_{m_i}}$

3.4.4 클러스터 내 멤버 노드 간 키 설립

클러스터내 멤버 노드들간 통신을 하기 위해서는 아래와 같은 과정을 수행하여 통신에 필요한 세션키를 설정한다. 이 세션키는 같은 클러스터내의 노드간 통신에만 사용된다.

1단계 M_i→CH_i : $\{id_{m_i}||id_{ch_i}||\{id_{m_i}\}_{s_{m_i}}||RV_6||\{id_{m_i}||sm_{m_i}||RV_6\}_{p_{m_j}}\}_{GK_{ch_i}}$

2단계 CH_i→M_j : $\{id_{ch_i}||id_{m_j}||p_{m_i}||\{id_{ch_i}\}_{s_{ch_i}}||RV_7||\{id_{m_i}||sm_{m_i}||RV_6\}_{p_{m_j}}\}_{GK_{ch_i}}$

3단계 M_j→CH_i : $\{id_{m_j}||id_{ch_i}||\{id_{m_j}\}_{s_{m_j}}||RV_7||\{id_{m_j}||sm_{m_j}||RV_8\}_{p_{m_i}}\}_{GK_{ch_i}}$

4단계 CH_i→M_i : $\{id_{ch_i}||id_{m_i}||p_{m_j}||RV_6||\{id_{ch_i}\}_{s_{ch_i}}||\{id_{m_j}||sm_{m_j}||RV_8\}_{s_{m_j}}\}_{GK_{ch_i}}$

5단계 M_i↔M_j : $\{id_{M_i}||id_{M_j}||\{id_{m_i}\}_{s_{m_i}}||\{id_{m_j}\}_{s_{m_j}}||RV_8||data\}_{s_{m_i-m_j}}$

3.5 클러스터 기반 동적 인증

네트워크 내에서 인증 과정이 시작되면 클러스터 관리자가 선출되고, 클러스터 관리자는 클러스터 헤드에게 키 분배를 한다. 이 과정에서 클러스터 관리자와 클러스터 헤드 간 그리고 클러스터 헤드 사이의 신뢰성 여부를 확인한다. 이들 간의 관계가 성공적으로 증명되면 클러스터헤드와 멤버 노드사이에 등록절차가 시작된다. 멤버 노드의 공개키 등록절차가 끝나면 통신하고자 하는 다른 노드와의 인증 절차를 갖고 통신이 시작된다. 각 클러스터 구성 요소 간의 키 공개 관계는 그림 2와 같다.

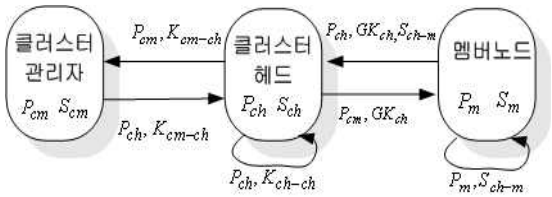

그림 2. 각 클러스터 구성 요소간 키 공개 관계

3.5.1 클러스터 관리자 노드와 클러스터 헤드 노드 간 인증

클러스터 헤드는 메시지 교환을 위해 공개키를 키 관리자와 다른 클러스터 헤드에게 알린다. 또 자신이 관리하는 클러스터내의 노드들에게도 클러스터 헤드 및 노드 간 신뢰성을 검사하기 위해 공개키를 알린다. 클러스터 관리자에 의해 각 클러스터 헤드에 대한 인증 과정이 수행된다. 클러스터 내에서 선출된 클러스터헤드에 대한 공개키 등록과정과 인증 과정을 통해 클러스터 관리자는 각 클러스터헤드에 대한 정보를 획득한다. 클러스터 관리자와 클러스터 헤드 간의 인증과정은 그림 3과 같다.

1단계 : 클러스터 헤드(CH_i)는 클러스터 관리자에게 자신을 인증해 달라고 임의의 값(RV_1)과 자신의 ID 및 공개키, 자신의 클러스터 그룹키를 클러스터 관리자의 공개키로 암호화하여 보낸다.

2단계 : 클러스터 관리자는 단계 1의 메시지를 받아서 해독한 후 클러스터 헤드의 그룹키와 공개키를 저장하고, 클러스터 헤드와의 공유키를 클러스터 헤드가 보내온 임의의 값(RV_1)과 자신임을 확인하는 정보와 함께 클러스터 헤드의 공개키로 암호화하여 전송한다.

3단계 : 클러스터 헤드(CH_i)는 클러스터 관리자가 보내온 메시지를 받은 후 자신의 비밀키로 해독하여 클러스터 관리자에게서 온 RV_1값과 자신이 보낸 임의의 값(RV_1)을 비교하여 일치하면 클러스터 관리자 인증 과정을 마친다.

4단계 : 클러스터 헤드 간 인증은 클러스터 헤드 사이에 생성된 공유키를 이용한다. 두 클러스터 헤드들은 자신임을 확인할 수 있는 임의의 값을 생성해 상호 교환하는 방식으로 인증 과정을 그림 4와 같이 수행한다.

3.5.2 클러스터 헤드 노드와 멤버 노드 간 인증

클러스터 헤드와 클러스터 관리자간 키 분배 및 인증 과정이 완료 되면 클러스터 헤드의 멤버 노드들은 신뢰성 있는 종단간 통신을 위해 공개키를 클러스터 헤드에게 알려야 한다. 그 과정은 그림 5와 같다.

1단계 : 클러스터 헤드(CH_i)는 클러스터 내에 자신의 공개키와 그룹 키를 공개하고 동일 클러스터 내의 모든 노드에게 공개키를 요구한다.

2단계 : 멤버 노드(M_i)는 임의의 값(N_i)을 설정하여 그 값과 자신의 ID를 자신의 비밀키로 암호화한 정보를 그룹 키로 암호화하여 클러스터 헤드(CH_i)에게 보낸다.

3단계 : 클러스터 헤드(CH_i)는 이 메시지를 받고 그룹 키로 해독한 후 M_i의 공개키를 자신의 공개키 관리 테이블에 멤버노드의 ID와 공개키를 저장한다. 그리고 M_i 보내온 임의의 값(N_i)과 자신의 ID를 자신의 비밀 키로 암호화한 정보를 다시 그룹 키로 암호화하여 M_i에게 전송한다.

4단계 : 메시지를 받은 멤버 노드(M_i)는 단계 3에서 받은 메시지를 해독하여 자신이 보낸 임의의 값(N_i)과 받은(N_i)가 같은지 확인하고, 같으면 멤버 노드(M_i)는 클러스터 헤드(CH_i)에게 인증되었다는 것을 확인하고, 다르면 새로운 임의의 수(N_b)를 선택하

Cluster manager, cluster-head, member node generate public/private key
SEND CM Broadcast all node in cluster
START Authentication between cluster manager and cluster-head
SEND **CHi→CM** $\{RV_1 \| id_{ch_i} \| GK_{ch_i} \| P_{ch_i} \| \{id_{ch_i}\} S_{ch_i}\} p_{cm}$
RECEIVE **CM** Decryption
STORE id_{ch_i}, P_{ch_i}, GK_{ch_i} of CHi
SEND **CM→CHi** $\{RV_1 \| \{k_{cm-ch_j} \| id_{cm}\}_{s_{cm}}\} p_{ch_i}$
RECEIVE **CHi** Decryption
if (send RV_1==receive RV_1)
{ success authentication
STORE $k_{cm-ch_j} id_{ch_i}$ }
else fail authentication
END Authentication between cluster manager and cluster-head

그림 3. 클러스터 관리자와 클러스터 헤드간 인증 알고리즘

START Authentication between cluster-head and cluster-head
SEND CHi→CM $\{id_{ch_i}||id_{ch_j}||RV_2||\{id_{ch_i}\}_{s_{ch_i}}||\{RV_2||\{id_{ch_i}\}_{s_{ch_i}}||id_{ch_i}\}_{p_{ch_j}}\}_{K_{CM-CH_i}}$
SEND CM→CHj $\{id_{cm}||id_{ch_j}||RV_3||\{id_{cm}\}s_{cm}||p_{ch_i}||\{RV_2||\{id_{ch_i}\}_{s_{ch_i}}||id_{ch_j}\}_{p_{ch_j}}\}_{K_{CM-CH_j}}$
RECEIVE CHj STORE id_{ch_i}, P_{ch_i} of CHi
SEND CHj→CM $\{id_{ch_j}||id_{cm}||RV_3||\{id_{ch_j}\}_{s_{ch_j}}||\{RV_4||id_{ch_i}||\{id_{ch_j}\}_{s_{ch_j}}\}_{p_{ch_j}}\}_{K_{CM-CH_j}}$
RECEIVE CM Decryption
if (send RV_3==receive RV_3)
SEND CHi $\{id_{cm}||id_{ch_i}||RV_2||\{id_{cm}\}s_{cm}||p_{ch_j}||\{RV_4||id_{ch_i}||\{id_{ch_j}\}_{s_{ch_j}}\}_{p_{ch_j}}\}_{K_{CM-CH_i}}$
else deny
RECEIVE CHi if (send RV_2==receive RV_2)
{ STORE id_{ch_j}, P_{ch_j} of CHj
SEND CHj $\{id_{ch_i}||id_{ch_j}||\{id_{ch_i}\}_{s_{ch_i}}||\{id_{ch_j}||K_{ch_i-ch_j}\}||RV_4||RV_5\}p_{ch_j}$ }
else deny
RECEIVE CHj if (send RV_4==receive RV_4)
SEND CHi $\{id_{ch_i}||id_{ch_j}||\{id_{ch_i}\}_{s_{ch_j}}||RV_5\}_{K_{CH_i-CH_j}}$
else deny
RECEIVE CHj if (send RV_5==receive RV_5)
success authention
else fail authentication
END Authentication between cluster-head and cluster-head

그림 4. 클러스터 헤드간 인증 알고리즘

START Authentication between cluster head and member node
SEND CHi→M request member node public key $\{id_{ch_i}||\{id_{ch_i}\}s_{ch_i}||p_{ch_i}||RV_6\}$
RECEIVE M STORE id_{ch_i}, P_{ch_i} of CHi
SEND M→CHi $\{id_m||id_{ch_j}||p_m||RV_6||RV_7||\{id_m\}s_m\}_{p_{ch_i}}$
RECEIVE CHi if (send RV_6==receive RV_6)
{ STORE id_m, P_m of M
SEND M $\{id_{ch_j}||\{id_{ch_i}\}_{s_{ch_i}}||GK_{ch_i}||sm_M||RV_7||RV_8\}p_M$ }
else deny
RECEIVE M if (send RV_7==receive RV_7)
{ success authentication
validation sm_m
STORE GK_{ch_i} of CH_i }
else fail authentication
END Authentication between cluster head and member node

그림 5. 클러스터 헤드와 멤버간 인증 알고리즘

여 재시도하고 또다시 실패를 하면 네트워크 내에서 새로운 클러스터 헤드를 찾는다.

3.6 클러스터 관리 과정에서의 키 관리

이동 애드혹 네트워크의 특성상 토폴로지의 변화는 빈번히 발생 할 수 있다. 따라서 이 장에서는 형성된 클러스터를 유지하는 과정에서 발생되는 클러스터 관리자의 부재, 클러스터 헤드의 양도 및 부재, 일반 노드의 진출입 등으로 인한 토폴로지 변화에 적절히 대처하기 위한 키 관리 방법은 다음과 같다.

클러스터 관리자가 자신의 역할을 수행하는데 있어 예상치 못한 상황이 발생하거나 자원 고갈로 인해 네트워크에서 사라질 경우, 재클러스터링을 통해 클러스터 관리자를 다시 선출하고 선출된 클러스터 관리자는 키 쌍을 생성하여 이를 네트워크에 알린 후 3.3절에서부터 3.5절까지의 과정을 수행하여 새로운 클러스터 기반 이동 애드혹 네트워크를 구성한다. 클러스터 헤드가 자신의 역할을 포기하거나 자원의 한계치에 도달한 경우, 그리고 자원 고갈 문제로 인해 예기치 못한 사항에서 네트워크에서 갑자기 사라질 수 있다. 이런 상황이 전개되면 부 클러스터 헤드가 클러스터 헤드의 역할을 양도 받는다. 그런 다음 해당 클러스터에 한해 재클러스터링 수행하여 새로운 부 클러스터 헤드를 임명 한다. 새로이 선출된 클러스터 헤드는 앞서 설명한 키 분배 절차를 반드시 거쳐 그룹키를 클러스터 관리자에게 등록한다. 그리고 클러스터 내 노드들의 공개키를 획득하기 위해 클러스터 헤드 및 인증 절차를 갖는다. 이동 애드혹 네트워크를 구성하는 노드는 이동 노드이므로 각 노드는 해당 클러스터를 벗어나거나 이웃 클러스터로 진입할 수 있다.

일반적으로 클러스터 내의 노드들은 주기적으로 클러스터 헤드가 보내는 inform 메시지를 통해서 자신이 클러스터 내에 있다는 것을 인식한다. 만약에 이 메시지를 받지 못하는 경우에는 자신이 클러스터 범위를 벗어났거나 혹은 클러스터 헤드가 그 기능을 하지 못한다고 인식하여 Search Cluster Request 메시지를 이용하여 자신이 포함된 클러스터 범위를 확인한다. 이 때 자신이 클러스터 안에 있음에도 불구하고 해당 클러스터의 클러스터 헤드로부터 inform 메시지를 받지 못했다면 이는 클러스터 헤드가 기능을 하지 못한다고 판단하여 새로운 클러스터 헤드를 선출한다. 이후 키 배분 과정을 수행하고 클러스터 내 노드들의 공개키를 획득하기 위해 클러스터 헤드 및 인증 절차를 갖는다. 반대로 inform 메시지를 받았는데 이 inform 메시지안의 클러스터 헤드의 ID가 다른 클러스터의 클러스터 헤드의 ID였다면, 새로운 클러스터 헤드와 인증 절차를 갖은 후 자신의 공개키를 등록하고 새로운 그룹키를 부여 받아 통신을 개시한다. 노드가 진입된 클러스터의 클러스터 헤드는 새로운 그룹키를 생성하여 해당 클러스터 멤버들에게 전송하고, 새로운 노드의 등록시 전에 해당 노드가 속해 있는 클러스터에게 이동 사실을 알린다. 이동 사실을 인지한 이전의 클러스터 헤드는 해당 노드의 공개키를 삭제하고 자신의 그룹키를 갱신하여 자신의 클러스터 멤버 노드들에게 전송한다. 이는 노드의 진출입으로 인해 발생할 수 있는그룹키의 무결성과 기밀성을 보장하기 위함이다.

4. 실험 및 성능 평가

제안된 클러스터 기반 동적 인증 기법은 오버헤드와 보안 요구사항 충족여부, 소요되는 인증시간을 통하여 평가한다.

4.1 실험환경 및 파라미터

Windows XP 환경에서 matrix-based 시스템의 프로그래밍 언어로써 각종 공학에 필요한 계산뿐만 아니라 여러 공학 분야에서 다양하게 시스템을 해석할 수 있는 최적의 도구인 MATLAB(MATrix LABoratory)을 이용하여 시뮬레이션을 수행했다.

클러스터 기반 동적 인증 기법의 성능을 평가하기 위해 사용되는 파라미터는 표 2와 같다.

표 2. 수식에 사용되는 파라미터

표기	설 명
TC_{CH}	클러스터 헤드가 부분 서명키를 생성하는데 걸리는 시간
TD_{CH}	클러스터 헤드가 자신이 받은 부분 비밀키를 클러스터 멤버들에게 분배하는데 걸리는 시간
TC_{Pk}	각 노드가 부분 비밀키를 생성하는데 걸리는 시간
TR_{CH}	멤버 노드가 헤드가 인증된 노드임을 확인하는데 걸리는 시간
P_1	클러스터 헤드 간, 클러스터 헤드와 일반 노드 간에 비밀키 전송시 2/3*n개 이상의 비밀키를 받는데 성공할 확률
P_2	네트워크 내의 모든 노드 중 1개 이상의 비밀키를 받는데 성공할 확률
P_3	클러스터 헤드간 비밀키 전송시 K개 이상의 비밀키를 받을 확률
l	2/3*m*n
r	m-n-1
T_A	전송 시간
T_D	암호 해독 시간

4.2 실험 결과

4.3.1 오버헤드

제안된 기법의 오버헤드는 크게 저장 공간 오버헤드, 계산 오버헤드 그리고 통신 오버헤드로 분류하고 제안된 기법이 포함된 전체 시스템 모델에 대한 오버헤드는 다음과 같다.

(1) 저장 공간 오버헤드

클러스터 기반 동적 인증 기법에서 하나의 클러스터에 속한 인증된 노드는 자신이 속한 클러스터의 그룹키, 클러스터 헤드의 공개키와 부분 비밀키, 자신의 개인키와 공개키를 소유한다. 그리고 멤버 노드는 추가적으로 다른 멤버 노드와의 통신을 위하여 세션키를 소유하게 되므로 각각의 노드 역할에 따라 소유하게 되는 키에 대한 저장 공간이 필요하다.

각 키의 길이는 8byte, 테이블의 크기는 30byte로 가정하고 클러스터 헤드의 수는 N_{ch}, 멤버 노드의 수는 N_m, 이웃 노드의 수는 N_n, 이웃 클러스터 헤드는 Nn_{ch}라고 정의하고 제안된 클러스터 기반 동적 인증 기법에서의 저장 공간 오버헤드는 기본 데이터 구조 이외에 발생되는 추가적인 저장 공간으로 한정한다. 임의의 노드가 클러스터 관리자이면 $((8\times N_{ch})+90)$ 바이트의 저장 공간 오버헤드가 발생하고, 클러스터 헤드이면 $((8\times N_m \times Nn_{ch})+98)$ 바이트 그리고 멤버 노드이면 46 바이트의 저장 공간 오버헤드가 발생한다.

(2) 계산 오버헤드

클러스터 기반 동적 인증 기법은 비밀분산 기법과 공개키 방식을 사용하므로 키 교환 및 암호화하는데 소비되는 시간이 적어서 이동 노드에게 매우 적은 계산 오버헤드가 발생한다. [12]의 RC5를 클러스터 기반 동적 인증 기법의 블록 암호화에 사용하면 8byte의 데이터를 가진 메시지를 암호하고 하나의 키를 가진 MAC 코드를 생성하는데 소요되는 시간이 2.38~3.32㎳이고 이 연산에 소비되는 에너지는 30μJ이다. 클러스터 기반 동적 인증 기법에서 하나의 이동 노드가 n개의 이웃노드로부터 전송된 패킷을 처리하는데 소요되는 계산 오버헤드는 $O(n)$이다.

(3) 통신 오버헤드

클러스터 기반 동적 인증 기법에서 노드사이에 데이터를 송수신하는데 소요되는 통신비용을 [13]에 제시된 값을 준용하여 산출한다. [13]에 따르면 36바이트의 패킷을 수신하는데 12.25μJ/byte가 소요되고, 송신하는 데는 16.25μ J /byte가 소요된다. 제안하는 기법에서는 클러스터 형성과정과 동적 인증과정에서 통신 오버헤드가 발생한다. 클러스터 기반의 안전한 통신에서 사용되는 모든 종류의 패킷은 8 바이트로 가정한다.

멤버 노드의 수는 q, 한 클러스터의 적정 노드수는 δ라고 정의하면, 클러스터 형성과정에서 클러스터 헤드의 통신 오버헤드는 초기 브로드캐스트 패킷, Join 패킷 그리고 Accept 패킷을 송수신하는데 (98+130δ+130q)μJ이 발생한다. 또한 멤버노드가 클러스터 헤드로부터 브로드캐스트 패킷을 p개 받았다면 클러스터 형성과정에서 멤버노드의 통신 오버헤드는 (16.25+12.25p)μJ이 발생한다.

동적 인증과정에서 임의의 노드가 클러스터 관리자이면 $((456\times N_{ch})+ (912\times Nn_{ch}))\mu J$의 통신 오버헤드가 발생하고 클러스터 헤드이면 $(912+(456\times Nn_{ch})+ (1368\times N_m))\mu J$ 통신 오버헤드가 발생한다. 그리고 멤버노드이면 $(652+(912\times N_m))\mu J$ 통신 오버헤드가 발생한다.

4.3.2 보안성 평가

제안된 기법은 이동 애드혹 네트워크 환경에 적합하도록 계층적 구조로 만들어졌기 때문에 네트워크 확장성이 용이하다. 또한 계층간 상호 인증을 통해 신뢰관계를 구축하여 기존 기법들이 고려하지 않았던 신뢰성 문제를 해결하였으며 두 노드간 통신에 있어서 항상 자신임을 입증하는 정보를 추가하여 전송하기 때문에 부인봉쇄가 가능하다. 또한 제안된 기법을 기존의 기법들과 안전성과 효율성 그리고 신뢰성과 같은 보안 평가 항목을 통해 비교해 보면 표 3과 같고, 기존의 기법들보다 효율성과 신뢰성, 부인봉쇄, 그리고 확장성이 우수함을 알 수 있다.

표 3. 완전 분산, 부분 분산 및 제안기법 비교

항 목	완전 분산	부분 분산	제안기법
키 분 배	오프라인	오프라인	온라인
고 정 C A	필 요	필 요	불필요
CH통신참가	참 가	참 가	참 가
신뢰성 고려	안 함	안 함	고려함
부 인 봉 쇄	지원안함	지원안함	지원함
효 율 성	보 통	보 통	좋 음
안 전 성	보 통	보 통	보 통
확 장 성	없 음	보 통	좋 음

4.3.3 효율성 평가

클러스터 기반 동적 인증 기법과 부분 분산 인증, 완전 분산 인증 비밀키 관리 기법의 노드 간 인증에 걸리는 시간 비교를 통해 효율성을 평가해 본다. 우선 (k,n) 임계치 기법을 사용하는데 있어 k는 전체 분산된 비밀키의 개수 n개 중 ((2/3)*n) 값으로 가정하고, 네트워크내의 클러스터 수는 n, 한 클러스터내의 멤버 노드 수는 m(0<m<10)라고 정의 한다.

부분 분산 기법에서 소요되는 전체 노드의 인증 시간(T_{PA})은 비밀키를 클러스터 헤드에게 분배하는 시간(T_{DK})과 n개의 클러스터 헤드 인증 시간(T_{CHA}) 및 m개의 일반 노드의 인증 시간(T_{MA})의 합으로 나타낼 수 있으면 식으로 표현하면 (식 6)~(식 8)과 같다.

$$T_{PA} = T_{Dk} + \sum_{i=1}^{n} T_{CHA} + \sum_{i=1}^{m} T_{MA} \quad \text{(식 6)}$$

$$T_{CHA} = TC_{CH} + T_A \times (nk)\binom{n}{nk} P_1^{nk}(1-P_1)^{n-nk} \quad \text{(식 7)}$$

$$T_{MA} = TD_{CH} + T_A \times (mk)\binom{m}{mk} P_1^{mk}(1-P_1)^{m-mk} \quad \text{(식 8)}$$

완전 분산 기법에서의 전체 노드의 인증 시간(T_{PA})은 오프라인 상에서 키를 분배하는데 걸리는 시간(T_{DK})과 각 노드가 부분키를 생성하는데 걸리는 시간 및 2/3*m*n(=1)개의 부분키를 모아 완전한 비밀키로 만드는 데 드는 시간의 합(T_i)으로 나타내고 (식 9)~(식 10)과 같이 표현 한다.

$$T_{AP} = T_{DK} + \sum_{i=1}^{l} T_i \quad \text{(식 9)}$$

$$T_i = TC_{Pk} + T_A \times (lk)\binom{l}{lk} P_2^{lk}(1-P_2)^{l-lk} \quad \text{(식 10)}$$

제안 기법에서 전체 노드의 인증 시간(T_{FA})은 클러스터 관리자와 클러스터 헤드 간 인증시간(T_{init}) 및 클러스터 멤버가 2/3*n개의 부분키를 모아 완전한 비밀키로 만드는 걸리는 시간과 멤버 노드가 헤드가 인증된 노드임을 확인하는데 걸리는 시간에 멤버 노드에게 온 메시지를 복호화하는데 드는 시간의 합(T_{cp})으로 나타낼 수 있으며 (식 11)~(식 12)과 같이 표현 한다.

$$T_C = T_{init} + \sum_{i=1}^{n} T_{cp} \quad \text{(식 11)}$$

$$T_{cp} = TC_{ch} + TD_{ch} + T_A \times (m\ k)\binom{m}{k} p_2^{mk}(1-P_2)^{m-mk} \quad \text{(식 12)}$$

전체 네트워크 구성 노드수를 150개로 설정하고 각각의 클러스터에 들어갈 노드의 크기는 0에서 10까지로 가정하고 클러스터의 증가에 따른 인증 시간을 분석 하였다. 클러스터 헤드의 전파 시간은 1㎳, 멤버 노드의 전파 시간은 2㎳로 설정하고 링크 오류 및 메시지 충돌로 인한 오류는 고려하지 않고 각 기법에서 비밀키를 받을 수 있는 확률을 0.9, 0.8, 0.7, 0.5로 고정값을 사용하였다. 그림 6에서 보는 것과 같이 클러스터 수가 증가함에 따라 기존의 기법보다 제안된 기법이 인증에 걸리는 시간이 짧게 나타냄을 볼 수 있다.

5. 결 론

이동 애드혹 네트워크는 기반 구조가 없는 구조에서도 사용가능하다는 장점을 가지고 있지만 동적 토폴로지의 변화와 중앙의 감시와 관리 결여, 무선 매체의 사용과 같은 고유의 특성으로 인해 기존에 사용되고 있는 유·무선망에 비해 보안에 많은 취약점을 갖는다. 현재 이동 애드혹 네트워크의 보안을 위해서 많은 인증 기법들이 설계 되었지만 비밀키 분배에 사용되는 이동 노드들의 신뢰성에 문제가 있다. 본 논문에서는 클러스터링 기법을 이용해 계층적 구조의 이동 애드혹 네트워크를 형성하고 이를 기반으로 신뢰성을 갖는 노드들만이 통신에 참여할 수 있도록 하는 계층적 인증 기법을 제안했다.

제안된 신뢰 인증 모델은 키 관리자와 클러스터 헤드, 멤버 노드로 구성되며, 클러스터 헤드는 분산 조합 가중치 알고리즘에 의해 선출하며, 선출된 클러스터 헤드 중에서 다시 클러스터 헤드 선출 알고리즘에 의해 키 관리자를 선출한다. 선출된 키 관리자에 의해 클러스터 헤드와 키 관리자간 인증 및 클러스터 헤드 간 인증이 이루어지며 그 다음에 클러스터 헤드와 멤버 노드간의 인증이 실행되는 계층적 인증 절차를 통해 신뢰 관계를 구축한다. 키 관리를 위해서는 비밀키 공유 기법 중의 하나인 부분 분산 기법을 적용했다. 공개키 및 ID를 교환하기 위해서는 공개키 암호 기법을 사용했으며 인증과 부인봉쇄를 위해 전

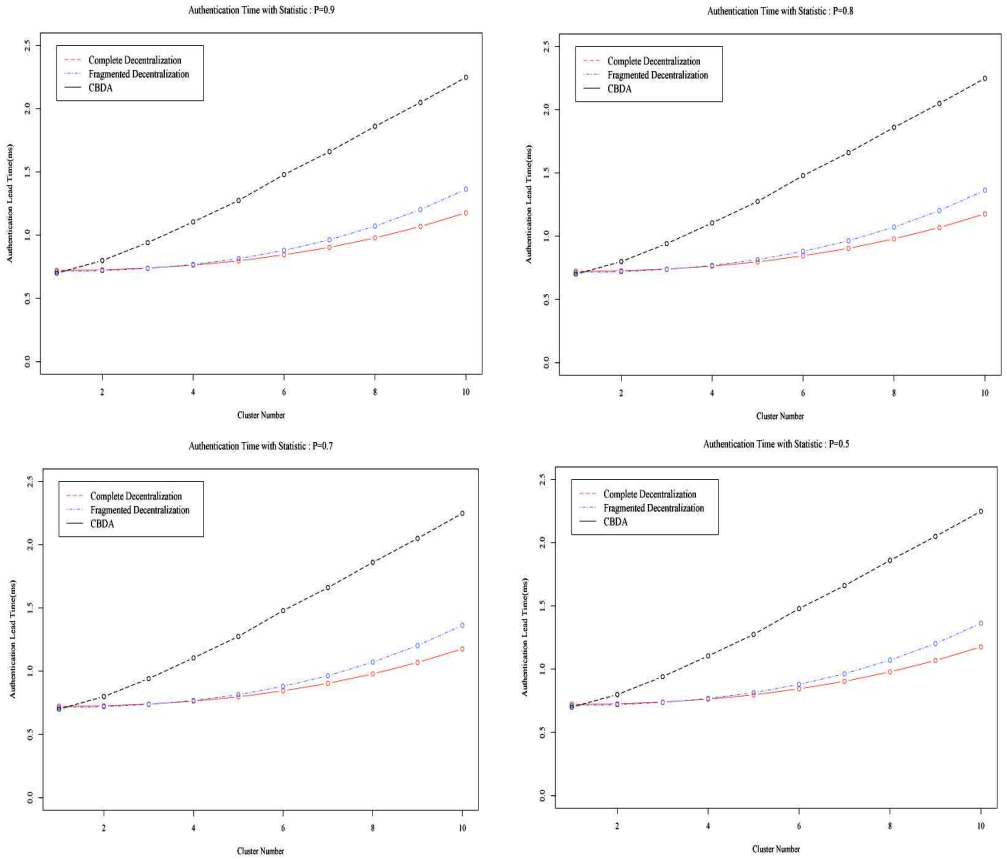

그림 6. 클러스터 수에 따른 인증 시간

자 서명 기법을 사용해 보안 요구 사항을 만족 시켰다. 또한 제안된 클러스터 기반 동적 인증 기법은 기존의 비밀 공유 기법에 비해 신뢰성과 확장성 그리고 부인봉쇄 기능이 있음을 보였다.

향후 연구과제로는 본 논문에서 제안된 기법을 실생활에 적용하여 사용할 수 있는 보다 안전적이고 효율적인 인증 기법의 연구가 지속적으로 필요하다.

참 고 문 헌

[1] M. S. Corson and J. Macker, "Mobile ad hoc networking(MANET): Routing protocol performance issues and evaluation considerations," *RFC 250*, Internet Engineering Task Force, Jan. 1999.

[2] A. Nasipuri and S. R. Das, "On-Demand Multipath Routing for Mobile Ad Hoc Networks," *Proc. of IEEE ICCCN'99*, Boston, Ma, pp. 64-70, 1999.

[4] A. Shamir, " How to Share a secret," *communication of the ACM*, Vol.22, No.11, pp. 612-613, 1979.

[3] A. Ephremides, J. Wieselthier and D. baker, "A design Concept for reliable Mobile Radio Networks with Frequency Hopping Signaling," *Proc. IEEE*, Vol.75, No.1, pp. 56-73, 1987.

[5] L. Zhou and Z. J. Hass, "Securing Ad -Hoc Networks," *IEEE Networks*, Vol.13, No.6, pp. 24-30, 1999.

[6] J.Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing Robust and Ubiquitous Secrity support for Mobile Ad-Hoc Network,"

IEEE ICNP 2001, 2001.

[7] Y. Desmedt, "Threshold Cryptography," *European Transaction on Telecommunication and Related Technologies*, Vol.5, No.4, pp. 35-43, 1994.

[8] C. Li, t. Hwang and M. Lee, "(t,n)-threshold signature schemes based on discrete Logarithm," *Advances in Cryptology-EUROCRYPT '94*, Springer-Verlag, LNCS 950, pp. 191-200, 1995.

[9] W. Diffie and M. E. Hellman, "New Directions in cryptography," *IEEE Transaction on Information Theory*, Vol.22, No.6, pp. 644-654, 1976.

[12] R. L. Rivest, "The RC5 encryption algorithm," *in: Bart Preneel (ED), Fast Software Encryption*, Springer, pp. 86-96, Mar. 1995.

[13] L. B. Oliveira, Hao Cho Wang, A. A Loureiro, "LHA-SP:secure pr otocols for hierarchical wireless sensor networks," *in:9th IFIP/IEEE International Symposium on Integrate Network Management*, pp. 31-44, May. 2005.

황 윤 철

1994년 한남대학교 전자계산공학과 졸업(공학사)
1996년 한남대학교 대학원 전자계산 공학과졸업(MS)
1999년~2008년 충북대학교 대학원 전자계산학과 박사 졸업
현재~배재대 교양교육지원센터 강사
관심분야 : 네트워크, IDS, ITS, Ad Hoc 및 센서네트워크 보안

김 진 일

2000년 한남대학교 대학원 컴퓨터공학과 박사 졸업
2003년 고려대학교 대학원 교육학 석사
2006년~현재 고려대학 대학원 교육학과 박사 과정
2006년~현재, 배재대 교양교육지원센터 IT분야 교수
관심분야 : 교육공학, 분산병렬, 네트워크, 보안, 유비쿼터스