CGSR 기반의 이동 애드 혹 네트워크에서 신뢰성 있는 통신을 위한 노드간 인증 기법

(A Multistage Authentication Strategy for Reliable N-to-N Communication in CGSR based Mobile Ad Hoc Networks)

이 혜 원 † 문 영 성 ††
(Hyewon K. Lee) (Youngsong Mun)

요 약 이동 애드 혹 네트워크(Mobile Ad Hoc Network, MANET)는 유선 기반 망에 의존하지 않으면서 이동 단말기들로 구성된 망으로 다중 홉 기반의 무선 통신을 제공한다. 그러나 동적인 토폴로지 변화, 중앙의 감시와 관리의 결여, 자원의 제약성, 무선 매체의 사용 등의 문제점 때문에 수동공격인 도청에서 능동공격인 DoS까지 다양한 공격에 노출되기 쉽다. 이를 위해 메시지 인증이나 사용자 인증, 안전한 패킷 전송 기법 등 다양한 보안 기법을 적용할 수 있으나, 인증이 이루어지지 않은 네트워크는 다른 보안성이 만족된다 하더라도 공격자에게 쉽게 노출된다. 본 논문에서는 CGSR[1]에서 제안하고 있는 클러스터링 기법을 기반으로 하여 인증된 노드들만이 통신에 참여할 수 있도록 하는 일반 노드와 클러스터 헤드 키 관리자로 구성된 계층적 노드 인증기법을 제안한다. 키 관리를 위해서는 부분 분산 기법[2]을 적용하며, 키 관리자와 클러스터 헤드 간 인증 및 클러스터 헤드간 인증, 일반 노드와 클러스터 헤드간의 인증 등의 다단계 인증절차 갖는다. 더 나아가 노드간 통신시 자신의 ID를 교환함으로써 부인봉쇄를 제공한다. 본 논문에서는 제안하는 메커니즘이 보안 요구사항을 어떻게 만족시키는지 분석하고 각 공격유형에 대한 방어 기법을 보인다. 성능평가를 위해서 제안하는 메커니즘의 인증 시간을 분석함으로써 노드증가 시에도 제안하는 모델이 잘 동작할 수 있음을 보인다.

키워드 : 이동 애드 혹 네트워크(MANET), CSGR, 클러스터링, 인증, 키 관리자, 클러스터 헤드, 일반 노드

Abstract A Mobile Ad Hoc Network(MANET) is a multi hop wireless network with no prepared base stations or centralized administrations, where flocks of peer systems gather and compose a network. Each node operates as a normal end system in public networks. In addition to it, a MANET node is required to work as a router to forward traffic from a source or intermediate node to others. Each node operates as a normal end system in public networks, and further a MANET node work as a router to forward traffic from a source or intermediate node to the next node via routing path. Applications of MANET are extensively wide, such as battle field or any unwired place; however, these are exposed to critical problems related to network management, node's capability, and security because of frequent and dynamic changes in network topology, absence of centralized controls, restricted usage on network resources, and vulnerability of mobile nodes which results from the special MANET's character, shared wireless media. These problems induce MANET to be weak from security attacks from eavesdropping to DoS. To guarantee secure authentication is the main part of security service in MANET because networks without secure authentication are exposed to exterior attacks. In this paper, a multistage authentication strategy based on CGSR is proposed to guarantee that only genuine and veritable nodes participate in communications. The proposed authentication model is composed of key manager, cluster head and common nodes. The cluster head is elected from secure nodes, and key manager is elected from cluster heads. The cluster

· 본 연구는 숭실대학교 교내연구비 지원으로 이루어졌음
† 학생회원 : 숭실대학교 컴퓨터학과
kerenlee@cherry.ssu.ac.kr
†† 종신회원 : 숭실대학교 컴퓨터학과 교수
mun@computing.ssu.ac.kr
논문접수 : 2005년 4월 7일
심사완료 : 2005년 9월 22일

head will verify other common nodes within its cluster range in MANET. Especially, ID of each node is used on communication, which allows digital signature and blocks non repudiation. For performance evaluation, attacks against node authentication are analyzed. Based on security parameters, strategies to resolve these attacks are drawn up.

Key words : mobile ad hoc network(MANET), CSGR, clustering, authentication, key manager, cluster head, common node

1. 소 개

이동 애드 혹 네트워크는 유선 기반 망에 의존하지 않으면서 이동 단말기들로 구성된 망으로 다중 홉 무선 통신을 제공한다. 초기에는 군사적인 응용 목적으로 연구되었으나 최근에는 무선 사설 망(WPAN), Bluetooth, 홈 네트워크, 센서 네트워크 등 실생활에 적용할 수 있는 다양한 응용 분야로 그 범위가 확대되고 있다[3]. 그러나 동적인 토폴로지 변화, 중앙의 감시와 관리의 결여, 자원의 제약성, 무선 매체의 사용 등의 문제점 때문에 수동공격인 도청에서 능동공격인 DoS까지 다양한 공격에 노출되기 쉽다. 노드의 신분이 서로에게 불확실한 경우가 많으며 멀티 홉 방식에 의해 라우팅을 할 경우 중간 노드에 의해 발생될 수 있는 데이타 보안 문제도 존재한다[4-6]. 이를 위해 메시지 인증이나 사용자 인증, 안전한 패킷 전송 기법 등 다양한 보안 기법을 적용할 수 있으나, 인증이 이루어지지 않은 네트워크는 다른 보안성이 만족된다 하더라도 공격자에게 쉽게 노출된다. 이에 따라, 현재까지 안전한 인증 서비스를 제공하기 위한 연구로 [2,7]과 같은 인증서 서비스 방안이 제시되었다. 그러나 이러한 서비스는 이동 노드 간 협력을 통한 분산 인증기법을 사용하면서도 인증에 참여하는 노드의 신뢰성을 언급하지 않는다. [7]은 네트워크 구성 초에 마스터 비밀키의 부분키를 미리 CA 노드들에게 분배하는 것을 가정하므로 애드 혹 네트워크의 확장성을 반영하지 못한다. 특히 [2]의 경우, 모든 통신이 서버(혹은 클러스터 헤드)를 거쳐 이루어지므로 서버의 과중한 오버헤드가 발생한다. 따라서 통신에 참여하는 노드의 신뢰성을 보장하며 다양한 공격을 방어할 수 있고 클러스터 헤드에 걸리는 부하를 최적화 시킬 수 있는 메커니즘이 필요하다.

본 논문에서는 CGSR을 기반으로 한 MANET에서 인증된 노드들만이 통신에 참여할 수 있도록 하는 계층적 노드 인증 기법을 제안한다. 제안된 인증모델은 일반 노드와 클러스터 헤드 및 키 관리자로 구성된다. 클러스터 헤드 및 키 관리자는 CGSR을 기반으로 선출되며 선출된 클러스터 헤드 중에서 키 관리자를 선출 후 키 관리자와 클러스터 헤드 간 인증 및 클러스터 헤드간 인증, 일반 노드와 클러스터 헤드간의 인증 등의 다단계 인증절차 갖는다. 키 관리를 위해서는 부분 분산 기법[2]을 적용하고 공개키 및 *ID* 교환을 위해 공개키 암호 메커니즘을 사용하며 인증과 부인봉쇄를 위해 전자서명 기법을 사용한다. 본 논문에서는 제안하는 메커니즘이 보안 요구사항을 어떻게 만족시키는지 분석하고 각 공격유형에 대한 방어 기법을 보인다. 성능평가를 위해서 제안하는 메커니즘의 인증 시간을 분석함으로써 노드증가 시에도 제안하는 모델이 잘 동작할 수 있음을 보인다.

2. 관련연구

2.1 MANET 환경에서의 공격 유형

네트워크에서의 공격 유형은 크게 외부공격과 내부공격으로 나눌 수 있고, 외부공격은 다시 수동공격과 능동공격으로 나눌 수 있다. 수동공격이란 전송중인 인증 받지 못한 노드들의 도청, 채널의 변경, 트래픽 분석 등을 말한다. 능동공격은 인가 받지 못한 노드들이 합법적인 노드들 간의 정상적인 통신 흐름을 방해하는 것으로 네트워크 정체를 유발하기 위한 무의미한 데이타 패킷 전송이나 전송중인 패킷의 변조 후 전송 등을 예로 들 수 있다. 이들은 네트워크의 혼잡 및 노드들 간의 라우팅 충돌을 유발시키기 위한 것으로 통신 노드들 간의 상호 인증과 메시지 인증 코드를 이용하여 해결할 수 있다. 내부공격은 타협된 이동 노드들 간에 발생되는 공격을 말하는 것으로 외부공격보다 발견 및 방어가 어렵다. 타협된 노드에 의한 공격을 방지하기 위해 침입 탐지 시스템을 이용해서 악의적인 노드를 배제시키거나 공격행위를 회피하는 등의 기법이 제시되고 있다. 본 논문에서는 외부공격 중에서도 능동공격만을 고려한다.

2.2 인증기법

2.2.1 비밀키 공유(secret key sharing) 기법

비밀키 공유 기법[8]이란 임의의 비밀키를 여러 사용자들이 나누어 갖게 하여 단일 사용자만으로는 원래의 비밀키를 복원할 수 없도록 하는 기법으로 (k, n) 임계치 암호화 기법[9]이 가장 대표적이다. (k, n) 임계치 암호화 기법은 하나의 비밀키를 n 명의 사용자에게 분산시켜 k 개의 분산된 비밀키 조각을 조합하여야만 비밀키를 복원할 수 있는데 어떤 악의적인 노드가 비록 k-1 개의 서버들에 대한 침입이 성공하더라도 비밀키 추출이 불가능하다. 하지만 k 개 또는 그 이상의 서버에 대한 침입이 성공하면 비밀키는 유출된다.

2.2.2 부분 분산 기법

[2]는 비밀키 공유기법을 애드 혹 네트워크에 적용시킨 부분 분산 인증기관 기법을 제안한다. 기존의 유·무선 환경에서는 안전한 인증 서비스를 제공하기 위하여 CA(Certificate Authority)를 통한 인증기법을 사용하지만 애드 혹 환경에서는 고정된 인프라가 존재하지 않으므로 일반적인 CA를 통한 인증기법은 사용할 수 없어서 이동 노드 간 협력을 통한 분산 인증기법을 사용한다. CA의 역할을 여러 노드들에게 분산시키고 이들의 협력을 통한 비밀키를 생성하므로 인증 프로세스에 참가하는 노드들의 신뢰도 여부가 중요하지만 [2]에서 제안하는 방안은 노드의 신뢰성을 고려하지 않고 다른 노드로부터 수신한 조각키의 검증 역시 고려하지 않는다.

2.2.3 완전 분산 기법

완전 분산 기법[7]은 애드 혹 네트워크가 구성되기 전에 통신에 참여할 각 노드가 중앙 관리자로부터 인증서 및 부분 비밀키 획득하는 것을 가정하고 있다. 따라서, 애드 혹 네트워크가 구성된 후 자가 초기화 알고리즘을 사용하여 중앙 관리자 없이 부분 비밀키를 관리할 수 있다. 부분 비밀키는 모든 노드에게 분배하며 임계치 암호화 기법을 사용하여 부분 인증서를 k 개 이상 취합후 완전한 인증서를 생성할 수 있다. 하지만, 네트워크 구성 이전에 부분 비밀키를 미리 CA 노드들에게 분배한다는 가정은 네트워크의 확장성에 부정적인 영향을 준다.

2.3 CGSR(Clusterhead Gateway Switch Routing)

CGSR[1]은 이동 호스트를 중심으로 1 홉 거리에 있는 노드들을 단위 네트워크(클러스터)로 구성한다. 클러스터마다 클러스터 헤드를 선출하고 두 개 이상의 클러스터와 접하고 있는 노드를 게이트웨이로 정하여 모든 통신이 클러스터 헤드 및 게이트웨이를 통해 수행되는 계층적 구조를 갖는 프로토콜이다. CGSR은 DSDV 프로토콜을 기반으로 하고 있어서 DSDV의 라우팅 특성을 상속한다. 클러스터 헤드의 선출을 위하여 휴리스틱 알고리즘을 적용하는데 대표적으로 Lowest-ID 알고리즘[10]과 Highest-Connectivity 알고리즘[11]이 있다. Lowest-ID 알고리즘은 해당 클러스터에서 가장 작은 ID 번호를 가지고 있는 이동 호스트를 클러스터의 헤드로 선택하는 반면 Highest-Connectivity 알고리즘은 해당 클러스터의 이동 호스트 중에서 연결성이 높은 노드를 헤드로 선정한다.

3. CGSR 기반 MANET에서 신뢰성 있는 종단간 통신을 위한 인증 메커니즘

3.1 인증 메커니즘 모델

CGSR 프로토콜 기반의 MANET에서 신뢰성 있는 통신을 위한 노드간 인증을 위해서 본 논문에서는 그림 1과 같은 여러 독립적인 클러스터로 구성된 네트워크 모델을 기반으로 한다. 또, 논리적으로 1 계층에 일반 노드, 2 계층에 클러스터 헤드 그리고 3 계층에 키 관리자가 배치되는 계층적 구조를 갖는다.

본 논문에서 제안하는 인증기법은 기존의 CGSR 프로토콜에 인증 메커니즘을 부가한 것으로 네트워크 구성 초기에 MANET을 구성하는 멤버들은 CGSR 프로토콜을 사용하여 네트워크를 구성한다. 본 논문에서는 클러스터 헤드가 n 개 이하인 경우에 이미 통신이 아무런 문제 없이 진행 된 경우 이 클러스터 헤드들은 믿을 수 있는 노드라 가정하며 능동공격만을 고려한다. 네트워크 내의 클러스터 헤드가 n 개 이상이 되는 경우에 클러스터 헤드간 인증, 키 관리자 선출 및 종단간 노드 인증을 시작한다. n 의 결정은 [9]에서 제안하는 알고리즘에 의해 결정된다.

각 클러스터는 하나의 클러스터 헤드와 하나 혹은 그 이상의 일반 노드로 구성된다. 클러스터 헤드는 각 클러스터를 대표하여 독립적으로 클러스터를 관리한다. 특히, 네트워크 내의 모든 클러스터 헤드는 주소 자원 및 경로 정보를 공유한다. 본 논문에서는 주소 할당에 대한 문제는 고려하지 않으며 통신 이전에 각 노드는 충돌이 없는 IP 주소를 이미 할당 받았다고 가정한다.

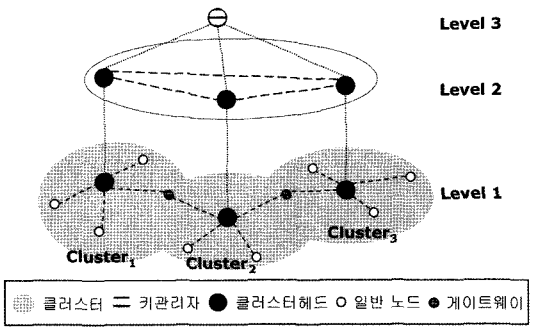

그림 1 CGSR을 기반으로 한 인증 모델

네트워크 내에서 인증프로세스를 시작하면 키 관리자를 선출하는데 키 선출을 위해서 클러스터 헤드 선출 알고리즘을 사용할 수 있다. 선출된 키 관리자는 네트워크 내의 클러스터 헤드에게 키 분배를 하고 이 과정에서 키 관리자와 클러스터 헤드간 그리고 클러스터 헤드간 신뢰성 여부를 확인한다. 이들간의 관계가 성공적으로 증명되면 클러스터 헤드와 해당 클러스터 내 노드간의 공개키 등록 절차를 갖는다. 일반 노드의 공개키 등록이 끝나면 통신하고자 하는 다른 노드와의 인증절차를 갖은 후 통신을 개시할 수 있다. 부인봉쇄를 위해서 노드간 교환되는 데이타 패킷은 모두 송신 노드의 비밀키로 암호화된 ID를 첨부하여 전송한다.

표 1 노드 및 표기법 정의

노드 형식	설명	노드 형식	설명	노드 형식	설명
KM	키 관리자	CH	클러스터 헤드	N	일반 노드
N'	N의 위장노드	N_{A_neig}	N_A의 이웃노드	N_{mal}	악의 노드
PK	공개키		SK	비밀키	
E_{PK}	PK를 사용하여 암호화		RV	임의의 수	
ID	노드의 ID		I_{rtg}	라우팅 정보	
$BList_N$	N의 블랙 리스트		$Message'$	변조된 메시지	
$N_A{:}p{\rightarrow}N_B$	노드 N_A가 패킷(p)을 N_B에 전달		$N_A{:}p{\Rightarrow}N_{A\_neig}$	NA가 이웃노드에게 브로드캐스팅.	
$N_A{:}p{\rightarrow}N_B{\rightarrow}N_C$	N_A가 N_B를 통해 N_C에게 패킷을 전달. N_B는 종단간 경로상 중간 노드.		$N_A{:}p{\rightarrow}N_B \mid{\rightarrow}N_C$	N_C로 향하는 패킷을 N_B가 전달하지 않음. N_B는 종단간 경로상 중간 노드.	

키 관리자, 클러스터 헤드 및 일반 노드는 모두 공개키/개인키 쌍을 생성한다. 먼저 키 관리자는 신뢰성 있는 노드간 통신을 위해 키를 생성하여 공개키를 네트워크에 알리며 모든 노드가 자신이 속한 클러스터를 관리하는 클러스터 헤드를 신뢰할 수 있도록 한다. 클러스터 헤드는 키 관리자 및 클러스터 헤드간 조각키 교환 및 시그널 교환 등을 위해서 공개키를 키 관리자와 다른 클러스터 헤드에게 알린다. 또, 자신이 관리하는 클러스터 내의 노드들에게도 클러스터 헤드 및 노드간 신뢰성 검사를 위해 공개키를 알린다. 일반 노드는 신뢰성 있는 종단간 노드 통신을 위해서 공개키를 클러스터 헤드에게 알린다. 노드간 키 공개 관계를 그림 2에서 볼 수 있다.

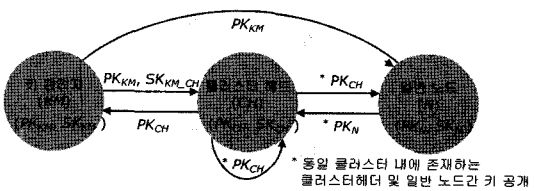

그림 2 네트워크 내의 키 공개 다이어그램

3.1.1 키 관리자 선출 및 키 분배

클러스터 헤드가 n 개 이상이 되면 아래와 같은 키 분배 및 복원 프로세스를 시작한다.

1. 클러스터 헤드 중에서 클러스터 헤드 선출 알고리즘을 사용하여 키 관리자를 선출한다.
2. 선출된 키 관리자는 공개키(PK_{KM})와 비밀키(SK_{KM})를 생성하여 공개키는 네트워크에 공개하고 서비스의 비밀키를 각 n 개의 클러스터 헤드들에게 분산시키기 위해 임계치 암호화 기법[9]을 사용한다. 비밀키 분산을 위해 키 관리자는 수식 차수가 k-1인 다항식을 선택한다. 수식 (1)에서 비밀키와 다항식의 계수 f_1, f_2, …, f_{k-1}는 키 관리자만이 알고 있다.

$$f(x) = SK_{KM} + f_1 x + f_2 x^2 + \cdots + f_{k-1} x^{k-1} \quad (1)$$

3. 클러스터 헤드(CH_i)는 선출된 키 관리자에게 임의의 수(RV_i)와 자신의 ID 및 공개키를 담아서 키 관리자의 공개키로 암호화 ($E_{PK\_KM}(RV_i|ID_{CHi}|PK_{CHi})$)하여 이를 키 관리자에게 전송한다.
4. 키 관리자는 3의 메시지 수신 후 이를 해독하여 RV_i와 2단계에서 만든 조각 키(SK_{KM_CHi})를 CH_i의 공개키로 암호화($E_{PK\_CHi}(RV_i|ID_{CHi}|SK_{KM\_CHi})$)하여 전송한다.
5. CH_i는 키 관리자가 전송한 메시지를 받은 후 자신이 보낸 RV_i와 비교하여 일치하면 해독한 조각키를 저장한다.
6. CH_i는 비밀키를 생성하기 위해서 나머지 클러스터 헤드에게 조각난 개인키를 보내줄 것을 요청하고 k 개 이상 받으면 이를 수식 (2)를 사용하여 조합하고[9] 키 관리자가 전송한 개인키와 공개키간의 오류가 없는지 확인한다. 이때 클러스터 헤드는 비밀키 교환을 위해서 서로의 공개키를 사용하여 메시지를 교환한다.

$$SK_{KM} = \sum_{i=1}^{k} l_{S_i} SK_{KM\_S_i},\ l_{S_i} = \prod_{j=1,\ j\neq i}^{k} \frac{S_j}{S_j - S_i} \quad (2)$$

3.1.2 일반 노드의 공개키 등록

클러스터 헤드와 선택된 키 관리자간의 키 분배가 성공적 끝나면, 일반 노드와 클러스터 헤드간의 신뢰성 여부를 확인하기 위해 아래와 같은 절차를 갖는다.

1. 클러스터 헤드(CH_i)는 클러스터에 자신의 공개키를 공개하고 동일 클러스터 내의 모든 노드에게 공개키 값을 요청한다. 일반 노드가 통신을 개시하고자 할 때 공개키가 등록되어 있지 않은 경우에 클러스터 헤드는 공개키를 요청할 것이다.
2. 일반 노드 N_A는 임의 값(RV_A)을 설정하여 RV_A와 자신의 ID 및 공개키를 CH_i의 공개키로 암호화 후 이를 다시 키 관리자의 공개키로 암호화하여 CH_i에게 보낸다$E_{PK_KM}(E_{PK_CHi}(RV_A|ID_{NA}|PK_{NA}))$.
3. CH_i는 3.1.1에서 산출한 비밀키와 자신의 비밀키를 사용하여 N_A로부터 받은 메시지를 해독 후 N_A의 공개키 및 ID를 저장하고 RV_A 및 CH_i의 ID를 N_A의 공개키로 암호화하여 응답한다.
4. 3에서 받은 메시지를 N_A가 해독하여 자신이 보낸 값

표 2 메시지 형식

메시지	설명	메시지	설명
S_PRIV[i]key	Service Private[*i*] Key. 키관리자가 *CH*에게 조각 비밀키 전송시 사용	*IA_REQ*	Identity Authentication Request. CH가 부분 비밀키를 요청할 때 사용.
IA_RESP	Identity Authentication Response. *IA_REQ*의 응답 메시지	*S_CH_REQ*	Search Cluster Head Request. 해당 클러스터 범위 내의 CH 탐색 메시지.
S_CH_RESP	Search Cluster Head Response. *S_CH_REQ*의 응답으로 *CH* 주소 반환	*ACK/NACK*	Acknowledgement/Negative acknowledgement. 긍정/부정 응답메시지
PK_REQ(*N*)	Public Key Request. 노드 *N*의 공개키 요청 메시지	*PK_RESP*	Public key Response. PK_REQ(N)의 응답 메시지
C_REQ	Communication Request. 통신요청 메시지	*C_Node*(*N*)	Change Node. 노드의 이동을 이웃 CH에게 알림
Inform	Information. *CH*가 클러스터에 보내는 주기적인 메시지		

과 동일하면 N_A와 CH_i은 믿을 수 있는 사이가 된다. 일치하지 않는다면 N_A는 새로운 임의 값을 선택하여 재시도하고 성공하지 않으면 네트워크 내에서 새로운 클러스터 헤드를 찾는다.

3.1.3 종단간 통신을 위한 노드간 인증

종단간 통신을 위해 일반 노드는 클러스터 헤드로부터 목적지 노드의 공개키를 얻어와 암호화에 사용하며 부인봉쇄를 위해 전자 서명을 사용한다. 즉, 목적지 노드와의 통신 이전에 상대방의 *ID* 및 공개키를 인증 단계에서 얻은 후 자신의 *ID*를 상대방의 공개키로 암호화하여 메시지 교환시에 같이 전송함으로써 위장을 방지한다. 각 노드는 상대 노드의 IP 주소와 *ID*를 가지고 있어서 수신한 메시지에서 복원한 *ID*가 자신이 가지고 있는 '주소-*ID* 테이블'과 다른 경우 이를 버린다. 노드간 인증을 위한 절차는 목적지 동일 클러스터 존재 유무에 따라 다르다. 상세한 절차는 3.8에서 다룬다.

3.2 *ID* 의 생성

인증이 끝난 후 종단간 노드가 통신을 시작할 때 메시지의 끝에 송신자의 *ID*를 수신자의 비밀키로 암호화해서 보냄으로써 수신자가 정확한 송신자로부터 온 것인지 확인할 수 있도록 하는 간단한 전자 서명 방식을 사용하여 인증과 부인 봉쇄를 만족시킬 수 있다. *ID*의 생성을 위해서는 임의의 해쉬 함수를 사용한다. 예를 들어, N_A는 통신하려는 목적지 노드(N_B)의 주소를 해쉬 함수로 입력 값으로 사용하여 산출한 값을 N_B에 대한 자신의 *ID*로 사용한다. 목적지 노드는 일반 노드 혹은 클러스터 헤드가 될 수 있다.

ID 생성시 적용된 해쉬 함수가 알려지기 전까지 송신자 및 수신자의 주소와 *ID*간의 관계를 찾아내는 것은 불가능하다. 또한 목적지 노드의 주소를 사용하여 송신자의 *ID*를 생성하여 목적지마다 다른 *ID*를 사용하기 때문에 송신자의 *ID* 유출을 최소화 시킬 수 있다.

3.3 클러스터 헤드의 양도

클러스터 헤드가 자신의 역할을 그만두려고 할 때 다음 클러스터 헤드를 선택한 후에 클러스터 헤드의 역할을 양도할 수 있다. 새로운 클러스터 헤드 선택은 CGSR 프로토콜에서의 헤드 선출 방식을 그대로 사용할 수 있다. 새로이 선출된 클러스터 헤드는 앞서 설명한 키 분배 절차를 반드시 거쳐야 한다. 또, 클러스터 내의 노드들의 공개키 획득을 위해 클러스터 헤드 및 일반 노드간 인증절차를 갖는다.

3.4 키 관리자의 양도

클러스터 헤드의 양도와 마찬가지로 키 관리자가 자신의 역할을 양도하고자 할 때 다음 키 관리자를 선택한 후에 클러스터 헤드의 역할을 양도할 수 있다. 키 관리자 선택 및 인증을 위해 3.1에서 설명한 절차를 반드시 거쳐야 한다. 키 관리자가 클러스터 헤드의 역할 역시 양도하고자 할 때 클러스터 헤드 양도 절차를 거쳐야 한다.

3.5 클러스터 헤드의 부재

클러스터 헤드는 배터리나 전기적인 문제로 인해 예기치 못한 상황에서 네트워크에서 갑자기 사라질 수 있다. 일반 노드는 주기적인 갱신 메시지를 클러스터 헤드로부터 받을 수 없기 때문에 클러스터 헤드의 부재를 인지할 수 있다. 이와 같은 경우 즉시 새로운 클러스터 헤드가 그 역할을 수행해야 한다. 새로운 클러스터 헤드의 선출은 CGSR 프로토콜에서의 헤드 선출 방식을 사용할 수 있다. 나머지 고려사항은 클러스터 헤드의 양도와 동일하다.

3.6 키 관리자의 부재

키 관리자 역시 예기치 못한 상황에서 네트워크에서 갑자기 사라질 수 있다. 이 경우, 클러스터 헤드는 주기적인 갱신 메시지를 키 관리자로부터 받을 수 없기 때문에 키 관리자의 부재를 인지할 수 있다. 이를 위해서 즉시 새로운 키 관리자가 그 역할을 수행해야 한다. 새로운 키 관리자가 선출되면 키 쌍을 생성하여 이를 네트워크에 알린 후 인증 프로세스를 시작한다. 나머지 고려사항은 키 관리자의 양도와 동일하다.

3.7 일반 노드의 진·출입

MANET을 구성하는 노드는 이동 노드이므로 각 노드는 클러스터 범위를 벗어나거나 추가될 수 있다. 일반

노드가 한 클러스터에서 다른 클러스터로 이동[3]할 때 이 노드는 단순히 자신의 클러스터만 찾으면 된다. 일반적으로 클러스터 내에 있는 노드들은 클러스터 헤드가 주기적으로 보내는 *Inform* 메시지를 통하여 자신이 그 클러스터 범위 내에 있다는 것을 인식한다. 만약 주기적인 메시지를 받지 못하면 일반 노드는 자신이 클러스터 범위에서 벗어났거나 혹은 클러스터 헤드가 그 기능을 하지 못한다고 인식하여 *S_CH_REQ* 메시지를 이용하여 자신이 포함된 클러스터의 범위를 확인한다. 이때 자신이 그 클러스터 범위 안에 있음에도 불구하고 클러스터 헤드(CH_i)로부터 *Inform* 메시지를 받지 못했다면 이는 클러스터 헤드의 기능을 하지 못한다고 판단하고 새로운 클러스터 헤드(CH_j)를 선출한다.

이와는 달리 *Inform* 메시지를 받았는데 이 *Inform* 메시지 안에 포함되어 있는 클러스터 헤드가 바뀌었다면 이는 자신이 다른 클러스터로 이동하였다고 인식한다. 일반 노드는 새로운 클러스터 헤드(CH_k)와 인증 절차를 갖은 후 자신의 공개키를 등록하여 통신을 개시한다. 새로운 클러스터 헤드는 일반 노드가 예전에 속해 있던 클러스터 헤드(CH_i)에게 노드의 이동 사실을 알린다. 이동 사실을 인지한 클러스터 헤드(CH_i)는 노드의 공개키를 삭제하고 자신의 이웃 클러스터 헤드들에게 노드의 이동 사실을 통보하며 노드의 이동 사실을 통보받은 이웃 클러스터 헤드들은 다시 이웃 클러스터 멤버 테이블을 갱신한다.

클러스터 헤드가 보내는 *N_REQ* 메시지에 노드가 응답하지 않은 경우 역시 클러스터 범위를 벗어난 것으로 간주한다. 예를 들어, $Cluster_1$에 속해 있던 일반 노드(N_B)가 클러스터 범위를 벗어난 경우, N_B는 클러스터 헤드(CH_1)가 보내는 주기적인 *N_REQ* 메시지에 응답할 수 없다. 임계값 이상의 전송 이후에도 응답이 없는 경우, CH_1은 N_B가 클러스터 범위를 벗어난 것으로 간주하여 자신이 저장하고 있는 주소 정보 및 *ID*와 공개키를 삭제하고 *C_Node*(N_B) 메시지를 사용하여 이를 이웃 클러스터 헤드(CH_2, CH_3)에게 알린다. 이때 CH_1은 상대 노드의 공개키로 암호화한 자신의 *ID*를 전송한다. 이웃 클러스터 헤드는 *C_Node*(N_B) 메시지를 받은 후 CH_1에 대응되는 *ID*를 확인 후 해당 노드 정보를 갱신한다.

3.8 일반 노드간 통신 시나리오

CGSR[1] 프로토콜에서 패킷 라우팅을 위해 통신을 개시하는 노드는 전송할 데이타 패킷을 자신의 클러스터 헤드에게 보내고 클러스터 헤드는 게이트웨이에게, 게이트웨이는 다시 이웃 클러스터헤드에게 데이타 패킷을 전송한다.

3) 클러스터 헤드의 이동은 클러스터 헤드 양도로 분류한다.

3.8.1 동일한 클러스터 내에서의 노드간 통신

동일한 클러스터 내에 존재하는 노드와의 통신을 개시하고자 할 때 먼저 종단간 노드의 인증을 위해서 다음의 절차를 따른다.

1. 그림 3에서와 같이 N_A가 같은 클러스터 안에 있는 N_B와 통신을 하고자 한다면 먼저 N_A는 자신이 속해 있는 클러스터의 헤드(CH_1)에게 N_B의 공개키를 요청한다.
2. 클러스터 헤드는 N_A의 공개키가 자신에게 등록되어 있으면 N_A를 통신에 타당한 사용자라고 인정하여 N_B의 공개키를 알려준다. 등록되어 있지 않는 경우, 클러스터 헤드는 N_A에게 공개키를 요청하고 3.1.2에서 설명한 공개키 등록 단계를 갖는다.
3. N_A는 N_B에게 통신 요청을 위해서 임의의 값(RV_1)을 N_B의 공개키(PK_{N_B})로 암호화하여 *C_REQ* 메시지를 전송한다. 이때 자신의 공개키(PK_{N_A})도 같이 전송한다.
4. N_B는 3에서의 메시지를 받고 해독 후 RV_1과 자신의 *ID*를 N_A의 공개키로 암호화 해서 전송한다.
5. N_A는 자신이 보낸 임의 값이 N_B로부터 받은 값과 동일하면 N_B의 *ID*를 저장한다.

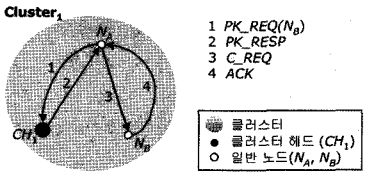

그림 3 한 클러스터 내에서의 통신

인증이 끝나면 N_A는 N_B로의 메시지 전송을 시작하는데 메시지 뒤에 자신 *ID*를 N_B의 공개키로 암호화해서 전송한다. N_B는 이 메시지를 받아서 복호화하고 *ID*가 N_A에 속한 것이 맞으면 정당한 사용자로부터 전송된 메시지임을 확인하게 되므로 메시지를 수락한다.

3.8.2 다른 클러스터에 속한 노드간 통신

다른 클러스터에 존재하는 노드와의 통신을 개시하고자 할 때 먼저 종단간 노드의 인증을 위해서 다음의 절차를 따른다.

1. N_A는 자신이 속한 클러스터를 관리하는 클러스터 헤드(CH_1)에게 목적지 N_C의 공개키를 요청한다.
2. 클러스터 헤드는 N_A의 공개키가 자신에게 등록되어 있으면 N_A를 통신에 타당한 사용자라고 인정하여 N_C의 공개키를 자신의 공개키 테이블에서 찾는다. 만약 자신의 공개키 테이블에서 N_C의 공개키를 찾을 수 없다면, 클러스터 헤드(CH_1)는 이웃 클러스터링 테이블에서 N_C가 속한 클러스터를 찾아 해당 클러스

터 헤드(CH_2)에게 N_C 공개키를 질의한다.

3. CH_2로부터 N_C의 공개키를 받으면 CH_1은 N_A에게 N_C의 공개키를 알린다.
4. 이후의 절차는 3.8.1의 절차와 동일하다. 이상의 절차를 그림 4에서 보이고 있다.

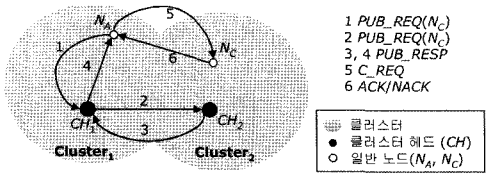

그림 4 다른 클러스터 범위에 있는 노드 간 통신

4. 성능 평가

본 논문에서 제안하고 있는 메커니즘은 클러스터링 기법을 사용하고 있어서 클러스터 헤드들만이 서비스의 비밀키 분배와 유지에 참여한다. 따라서 새로운 노드가 네트워크에 진입하는 경우에 특별한 키 재분배 없이 자신의 공개키를 클러스터의 헤드에 등록하는 것만으로 통신에 참여할 수 있어 확장성이 좋다. 또한 클러스터링 기법은 클러스터 단위로 코드 분할, 채널 액세스, 라우팅, 대역폭 할당을 하기 때문에 단일 계층으로 구성된 네트워크 보다 효율적이다.

본 논문에서 제안하고 있는 인증 메커니즘은 인증 및 부인봉쇄를 만족시킨다. 즉, 노드간 공개키 및 조각 비밀키 분배시 공개키 메커니즘을 사용하기 때문에 비밀키가 누설되기 전에는 노드간 신뢰성을 보장한다. 또, 인증이 끝난 후 종단간 노드가 통신을 시작할 때 각 노드는 특정 ID를 사용함으로써 실제 노드로부터 전송된 것임을 확인할 수 있다. 표 3에서 [2,7]에서 제안하는 기법과 본 논문에서 제안하는 기법을 비교 분석하였다.

4.1 공격 유형에 따른 방어 기법

임의의 노드 N_A, N_B, $N_C \in N$에 대해 각 공격 유형에 따른 방어 기법을 설명하기 위하여 노드의 형식 및 표기법을 표 1과 같이 정의한다. 애드 혹 네트워크에서 인증과 관련하여 나타날 수 있는 각각의 공격 유형에 대한 대처방안은 아래와 같다.

4.1.1 N_A:$p \rightarrow N_B \mid \rightarrow N_C$

N_A가 N_C로 패킷을 전송하고자 할 때 경로 상에 있는 노드(N_B)가 패킷을 전송하지 않을 수 있다. 제안하는 모델에서는 통신 요청에 대해 N_C가 긍정 혹은 부정 응답을 사용하여 통신하고자 하는 의사를 표시하였으므로 이에 근거하여 대처한다. 즉, 패킷 손실이 악의 있는 노드 때문인지 혹은 통신 시 발생된 손실인지 파악하여 N_B가 악의 노드(N_{mal})로 판명되면 이 노드를 블랙리스트($BList_{NA}$)에 올린다. N_A는 우회경로[4]를 사용하여 패킷을 재전송하고 N_{mal}로 판단된 노드를 통신에서 배제시킴으로써 이러한 공격에 대처할 수 있다.

4.1.2 N'_A:$p \rightarrow N_B$ 혹은 N_B: $p \rightarrow N'_A$

악의 있는 노드가 N_A로 가장하여 메시지를 송신하거나 혹은 수신하는 경우가 있을 수 있다. 본 논문에서 제안하는 메커니즘은 각 메시지 끝에 송신자의 ID를 송신자의 비밀키로 서명하여 보냄으로써 수신자가 송신자의 공개키로 암호화된 ID를 복호화하여 타당한 사용자로부터 온 메시지라는 것을 확인할 수 있도록 한다.

4.1.3 N_{mal}:$C\_Node \rightarrow CHis$

인증 받은 클러스터 헤드가 주기적으로 전송하는 N_REQ 메시지에 대해 클러스터 내의 어떤 노드가 N_RESP 메시지로 자신의 ID를 반환하지 않으면 3.7에서 설명한 것처럼 이 노드가 자신의 클러스터 범위를 벗어난 것으로 간주하여 노드의 공개키 값을 테이블에서 삭제하고, C_Node 메시지를 사용하여 그 사실을 이웃한 클러스터 헤드에게 알린다. 하지만 악의 있는 노드가 어떤 노드를 고립시키기 위해 이웃 클러스터 헤드에게 C_Node 메시지를 보낼 수 있다. 예를 들어 N_B가 CH_1이 관리하는 클러스터에 속해 있는 공격받는 노드이고 CH_2와 CH_3가 이웃한 클러스터 헤드라고 가정할 때 공격자 노드 N_{mal}이 C_Node 메시지를 이웃 클러스터 헤드 CH_2와 CH_3에 전송하더라도 각 클러스터 헤드와의 통신에 사용되는 CH_1의 ID를 공격자가 알 수 없기 때문에 이웃한 클러스터 헤드는 CH_1이 전송한 메시지가 아님을 알 수 있다.

표 3 완전 분산, 부분 분산 기법 및 제안 기법의 요약

	완전 분산[7]	부분 분산[2]	제안 기법
온라인 상에서의 키 분배 여부	X	X	O
클러스터헤드(서버노드)의 통신참가 여부	O	O	O
CA 필요 여부	O	O	X
키 분배 절차에서의 노드 신뢰성 고려 여부	O	O	O
키 분배 후 노드 신뢰성 고려 여부	X	X	O
주소 변조 대처 가능 여부	X	X	O
전자서명 제공 여부	X	X	O

4.1.4 N_{mal}: $Inform' \Rightarrow N_{mal\_neig}$

클러스터 헤드는 주기적으로 $Inform$ 메시지를 자신이 관리하고 있는 클러스터에 브로드캐스팅한다. 일반 노드는 이 메시지를 수신함으로써 자신이 이동을 했는지 여부를 결정하게 된다. 이때 공격자가 가짜 ID를 사용하여 $Inform$ 메시지를 생성 후 이웃 노드들에게 전송할 수 있지만 일반 노드는 클러스터 헤드와의 인증 절차에서 두 개의 공개키를 사용하여 ID와 노드의 공개키를

4) 우회 경로를 위해 N_{mal}을 배제한 종단간 최적 경로가 선택된다.

수신하기 때문에 네트워크의 비밀키를 가지고 있지 않은 노드의 경우 이 메시지를 복원할 수 없다. 일반 노드는 공격자가 생성한 *Inform* 메시지였음을 용이하게 확인이 가능하므로 이러한 종류의 공격은 쉽게 피할 수 있다.

4.2 인증 시간 분석

제안된 비밀키 관리기법과 부분 분산 및 완전 분산 비밀키 관리 기법의 노드 간 인증에 걸리는 시간을 비교 분석한다. (k, n) 임계치 암호화 기법을 사용하는데 있어 k는 전체 분산된 비밀키의 개수 n 개 중 [2/3×n] 값으로 가정한다. 네트워크 내의 클러스터 헤드의 수는 n으로 한 클러스터 내의 노드의 수를 m($0<m<10$)이라 정의한다.

부분 분산 기법에서 소요된 전체 노드의 인증시간은 비밀키를 클러스터 헤드에게 분배하는데 걸리는 시간과 n 개의 클러스터 헤드 인증시간 및 m 개의 일반 노드의 인증시간의 합으로 나타낼 수 있다(3). 클러스터 헤드의 인증시간은 각 클러스터 헤드가 부분 비밀키를 생성하는데 걸리는 시간과 2/3×n 개 이상의 부분 서명 비밀키를 모아 완전한 인증서를 생성하는데 걸리는 시간의 합으로, m 개의 일반 노드 인증시간은 클러스터 헤드가 클러스터 멤버들에게 자신이 받은 부분 서명 비밀키를 분배하는데 걸리는 시간과 2/3×m 개 이상의 부분 서명 비밀키를 모아 완전한 인증서를 생성하는데 걸리는 시간의 합으로 나타낼 수 있다. (3)에서 a_j는 각 클러스터 헤드가 부분 서명 비밀키를 생성하는데 걸리는 시간, a_k는 클러스터 헤드가 자신이 받은 부분 서명 비밀키를 클러스터 멤버들에게 분배하는데 걸리는 시간, 그리고 P는 클러스터 헤드 간 비밀키 전송 시 2/3×n 개 이상의 비밀키를 받는데 성공할 확률 또는 클러스터 헤드와 일반 노드 간에 비밀키 전송 시 2/3×m 개 이상의 비밀키를 받는데 성공할 확률이다.

$$
\begin{aligned}
&T_{TA} = T_{DK} + \sum_{i=1}^{n} T_{CHAi} + \sum_{i=1}^{m} T_{CAi},\\
&T_{CHA_i} = a_j + T_{c\_bHN} * nk\binom{n}{nk} p^{nk}(1-p)^{n-nk},\\
&T_{CA_i} = a_k + T_{c\_bHN} * mk\binom{m}{mk} p^{mk}(1-p)^{m-mk}.
\end{aligned} \tag{3}
$$

완전 분산 기법[7]에서 전체 노드의 인증시간은 오프라인상에서 키를 분배하는데 걸리는 시간과 각 노드가 부분인증서를 생성하는데 걸리는 시간, 2/3×m×n(=l) 개 이상의 부분 인증서를 모아 완전한 인증서를 생성하는데 걸리는 시간의 합으로 나타낼 수 있다. (4)에서 a_j는 각 노드가 부분 인증서를 생성하는데 걸리는 시간이고 P는 네트워크 내의 모든 노드 중 l 개 이상의 비밀키를 받는데 성공할 확률을 나타낸다.

$$
\begin{aligned}
&T_{TA} = T_{DK}^{'} + \sum_{i=1}^{l} T_i\\
&T_i = a_j + T_{c\_bHN} * lk\binom{l}{lk} p^{lk}(1-p)^{l-lk}
\end{aligned} \tag{4}
$$

제안 메커니즘에서 전체 노드의 인증시간은 키 관리자 및 클러스터 헤드 간 인증 시간, 일반 노드가 헤드가 인증된 노드임을 확인하는데 걸리는 시간의 합으로 나타낼 수 있다. (5)에서 a_j는 각 인증 요청 메시지를 헤드에게 전송하는데 걸리는 시간이고 P는 클러스터 헤드와 일반 노드 간 비밀키 전송 시 k 개 이상의 비밀키를 받는데 성공할 확률이다.

$$
\begin{aligned}
&T_{TA} = T_{imit} + \sum_{i=1}^{m} T_{CA_i}\\
&T_{CA_i} = a_j + T_{c\_bHN} * mk\binom{m}{mk} p^{mk}(1-p)^{m-mk}
\end{aligned} \tag{5}
$$

전체 네트워크 구성 노드 수를 10에서부터 100까지 증가시켜가면서 노드 증가에 따른 인증시간을 분석하였다. 한 클러스터는 10 개의 노드로 구성되었다고 가정한다. 클러스터 헤드와 일반 노드는 1 홉, 클러스터 헤드 간의 거리는 2 홉으로 전파 시간은 각각 1*ms*, 2*ms*라 가정하였다. 링크 오류 및 메시지 충돌로 인한 재 전송은 고려하지 않았다. 서명에 걸리는 시간은 RSA를 기반으로 설정하였다. 각 기법에서 비밀키를 받는데 성공할 확률(P)은 0.98, 0.9, 0.8이라는 고정 값을 사용하였다. MANET에서 최초의 인증을 거친 노드가 이동하는 경우가 있을 수 있으나, 클러스터 헤드끼리 *V_C_Node* 메시지를 통해 정보를 공유하므로 이동성은 고려하지 않았다. 그림 5는 전체 네트워크 구성 노드 수를 증가시켰을 때 기존연구보다 제안된 메커니즘이 인증에 더 짧은 시간이 소요됨을 나타낸다. 또한 이 그래프를 통해 비밀키를 받는데 성공할 확률이 0.98에서 0.80으로 감소할수록 인증을 위한 시간이 증가함을 알 수 있다.

5. 결 론

애드 혹 네트워크는 기반구조가 없는 환경에서도 사용 가능하다는 장점을 가지지만 동적인 토폴로지, 중앙의 감시와 관리의 결여, 자원의 제약성, 무선 매체의 사용 등 MANET 고유의 특성으로 인해 기존에 사용되고 있는 유·무선망보다 보안에 취약한 단점을 갖는다. 현재 많은 인증 기법들이 애드 혹 네트워크 보안을 위해 설계되었으나 이러한 메커니즘들은 서비스 비밀키 분배에 사용되는 이동 노드들의 신뢰성 문제를 가진다. 본 논문에서는 클러스터링 기법을 기반으로 하여 인증된 노드들만이 통신에 참여할 수 있도록 하는 계층적 노드 인증기법을 제안한다. 클러스터 헤드 및 키 관리자는 CGSR

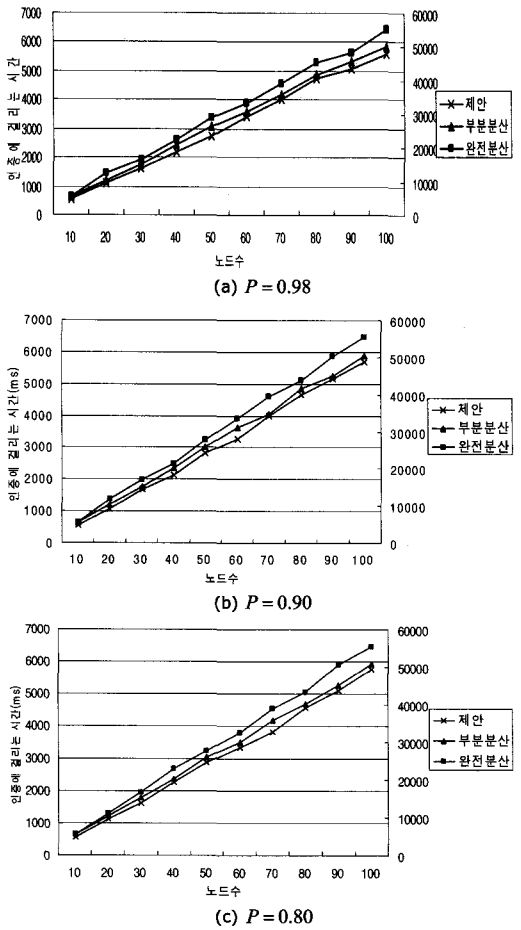

그림 5 노드 증가에 따른 인증 시간 비교

을 기반으로 선출하여 키 관리자와 클러스터 헤드 간 인증 및 클러스터 헤드간 인증, 일반 노드와 클러스터 헤드간의 인증 등의 계층적 인증절차 갖는다. 키 관리를 위해서는 부분 분산 기법[2]을 적용하고 공개키 및 *ID* 교환을 위해 공개키 암호 메커니즘을 사용하며 인증과 부인봉쇄를 위해 전자서명 기법을 사용한다. 클러스터 헤드간 인증 절차를 통해 비밀키를 분배 받는 클러스터 헤드가 신뢰할 수 있는 노드임을 증명하고, 공개키 메커니즘과 서명기법을 사용하여 보안 요구사항을 만족시켰다. 본 논문에서는 제안하는 메커니즘이 보안 요구사항을 어떻게 만족시키는지 분석하고 각 공격유형에 대한 방어 기법을 보였다. 성능평가를 위해서 제안하는 메커니즘의 인증 시간을 분석함으로써 노드증가 시에도 제안 하는 모델이 잘 동작할 수 있음을 보이고 더 나아가 확장성 측면에서 적절함을 보였다.

참 고 문 헌

[1] C. Chiang, H. Wu, W. Liu and M. Gerla, "Routing in clustered multihop, mobile wireless networks with fading channel," Proc. IEEE Singapore International Conference on Networks, Apr. 1997.

[2] L. Zhou and Z. J. Haas, "Securing ad hoc networks," IEEE Network Magazine, Vol. 13, No. 6, pp. 24-30, 1999.

[3] R. Jordan and C. T. Abdallah, "Wireless communications and networking: an overview," Antennas and Propagation Magazine, IEEE, Feb. 2002.

[4] H. Yang, H. Luo, F. Ye, S. Lu and L. Zhang, "Security in mobile ad hoc networks: challenge and solution," IEEE Wireless Communications, Feb. 2004.

[5] A. Mishra, K. Nadkarni, A. Patcha and V. Tech, "Intrusion detection in wireless ad hoc networks," IEEE Wireless Communications, Feb. 2004.

[6] S. Zhu, S. Xu, S. Setia and S. Jajodia, "LHAP: A lightweight hop by hop authentication protocol for ad hoc networks," ICDCSW'03, May 2003.

[7] J. Kong, P. Zerfos, H. Luo, S. Lu and L. Zhang, "Providing robust and ubiquitous security support for mobile ad hoc networks," ICNP'01, pp. 251-260, 2001.

[8] Y. Frankel and Y. G. Desmedt, "Parallel reliable threshold multisignature," Tech. Report TR-92-04-02, Univ. of Wisconsin Milwaukee, Apr. 1992.

[9] A. Shamir, "How to Share a Secret," Massachusetts Institute of Technology, Communication of the ACM, 22(11), pp. 612-613, 1979.

[10] A. Ephremides, J. Wieselthier and D. Baker, "A Design Concept for Reliable Mobile Radio Networks with Frequency Hopping Signaling," Proc. IEEE 75(1), pp. 56-73, 1987.

[11] M. Gerla and J. Tsai, "Multicluster, Mobile, Multimedia Radio Networks," Baltzer Journels, Vol. 1, No. 3, pp. 255-265, July 1995.

이 혜 원
정보과학회논문지 : 정보통신
제 32 권 제 1 호 참조

문 영 성
정보과학회논문지 : 정보통신
제 32 권 제 1 호 참조