애드혹 네트워크에서의 one-time 전자 서명을 이용한 라우팅 보안 메커니즘

편 혜 진[†] · 도 인 실[††] · 채 기 준[†††]

요 약

애드혹 네트워크는 기존의 유무선 네트워크의 고정된 기반시설(infrastructure) 없이 이동 호스트들만으로 구성된 무선 환경의 네트워크이다. 애드혹 네트워크의 기본 특성, 즉, 링크의 불안정성, 각 노드의 물리적 보호의 한계, 노드간 연결의 산재성, 토폴로지의 동적인 변화 뿐 아니라 악의적인 노드의 활동으로 인해 라우팅 보안에 대한 위험성은 매우 높다. 따라서 본 논문에서는 애드혹 네트워크에서 경로 탐색이나 설정 과정 중 악의적인 노드가 라우팅 메시지를 변조, 위조하거나 다른 노드를 가장하여 잘못된 라우팅 정보를 네트워크에 주입시키는 공격을 방지하기 위하여 일방향 해쉬 함수를 기초로 한 one-time 전자 서명을 이용한 라우팅 보안 메커니즘을 제안한다. 제안하는 메커니즘에서 노드들은 라우팅 메시지를 서명하기 위하여 공개키 요소의 첫 세트를 반복적으로 해쉬 함수에 적용함으로써 해쉬 체인을 생성하고, 생성된 해쉬 체인으로부터 공개키 요소들을 여러 세트 유도하여 해쉬 테이블을 생성한다. 해쉬 테이블 생성 후, 노드들은 자신의 공개키 요소를 다른 노드들에게 공표하고 라우팅 메시지를 전송할 경우 one-time 전자 서명을 포함한다. 이러한 one-time 전자 서명은 라우팅 메시지를 인증하고 메시지에 무결성을 제공한다. 제안하는 라우팅 보안 메커니즘은 이동성이 높은 네트워크 환경에서는 보안을 고려하지 않은 라우팅 메커니즘에 비해 라우팅 오버헤드가 좀더 높아지지만, 경로를 탐색하고 설정하는 과정에서 악의적인 노드의 공격에 대하여 훨씬 높은 안전성을 제공함을 시뮬레이션을 통해서 확인할 수 있다.

키워드 : 애드혹 네트워크, 라우팅 보안, one-time 전자 서명

Secure Routing Mechanism using one-time digital signature in Ad-hoc Networks

Hyejin Pyeon[†] · Inshil Doh[††] · Kijoon Chae[†††]

ABSTRACT

In ad-hoc network, there is no fixed infrastructure such as base stations or mobile switching centers. The security of ad-hoc network is more vulnerable than traditional networks because of the basic characteristics of ad-hoc network, and current routing protocols for ad-hoc networks allow many different types of attacks by malicious nodes. Malicious nodes can disrupt the correct functioning of a routing protocol by modifying routing information, by fabricating false routing information and by impersonating other nodes. We propose a routing security mechanism based on one-time digital signature. In our proposal, we use one-time digital signatures based on one-way hash functions in order to limit or prevent attacks of malicious nodes. For the purpose of generating and keeping a large number of public key sets, we derive multiple sets of the keys from hash chains by repeated hashing of the public key elements in the first set. After that, each node publishes its own public keys, broadcasts routing message including one-time digital signature during route discovery and route setup. This mechanism provides authentication and message integrity and prevents attacks from malicious nodes. Simulation results indicate that our mechanism increases the routing overhead in a highly mobile environment, but provides great security in the route discovery process and increases the network efficiency.

Key Words : Ad-hoc Network, Routing Security, One-time Digital Signature

1. 서 론

애드혹 네트워크는 기존의 유무선 네트워크의 고정된 기반시설(infrastructure)없이 이동 호스트들 간의 통신을 통해서 이루어지는 단일 혹은 다중 홉의 무선 환경 네트워크를 말한다. 따라서 네트워크를 구성하는 모든 이동 호스트들은 데이터를 송수신하는 서비스를 제공받음과 동시에 다른 호스트들을 위해 데이터를 전달하는 라우터의 기능까지 제공해야 한다[1]. 그러나 애드혹 네트워크에는 호스트들을 관리하는 중앙 기반시설이 존재하지 않기 때문에 보안상 취약성이 기존의 네트워크에 비해 높을 수밖에 없다. 특히 애드혹 네트워크

※ 본 연구는 정보통신부 및 정보통신연구진흥원의 대학 IT연구센터(ITRC) 육성·지원사업의 연구결과로 수행되었음.
† 정 회 원 : 삼성전자(주) 정보통신총괄 무선사업부
†† 준 회 원 : 이화여자대학교 컴퓨터학과 박사과정
††† 종신회원 : 이화여자대학교 컴퓨터학과 교수
논문접수 : 2005년 3월 29일, 심사완료 : 2005년 7월 18일

는 각 노드들이 라우터로서의 역할을 동시에 수행해야하기 때문에 네트워크 내의 노드 하나만 오염시켜도 이 노드를 통해 잘못된 라우팅 정보를 쉽게 퍼뜨림으로써 전체 네트워크를 마비시킬 수 있다는 보안상의 큰 취약점을 갖는다. 또한 애드혹 네트워크는 각 노드들이 권한 부여 없이 동등하게 서로 라우팅 메시지를 주고받음으로써 네트워크에 참가하고 이로써 하나의 네트워크가 형성되는 것이기 때문에 악의적인 노드가 쉽게 애드혹 네트워크에 참여할 수 있고 그 결과 라우팅 과정 중에서 많은 공격을 수행할 수 있게 된다[2]. 따라서 애드혹 네트워크에서 경로 탐색 과정 중 악의적인 노드의 공격을 탐지하고 방어할 수 있는 라우팅 보안 메커니즘이 반드시 필요하다.

본 연구에서는 경로 탐색이나 경로 설정 과정에서 악의적인 노드가 라우팅 메시지를 변조, 위조하거나 다른 노드를 가장하여 잘못된 라우팅 정보를 네트워크에 주입시키는 공격을 방지하기 위하여 일방향 해쉬 함수를 기초로 한 one-time 전자 서명을 이용한다. 해쉬 테이블 생성 후, 네트워크에 참여하는 노드들은 자신의 공개키 요소를 다른 노드들에게 공표하고 이 후 라우팅 메시지를 생성할 경우 one-time 전자 서명을 포함하여 라우팅 메시지를 인증하고 메시지 무결성을 제공하도록 한다.

본 논문은 다음과 같은 순서로 구성되어 있다. 1장의 서론에 이어서 2장에서는 애드혹 네트워크의 라우팅 보안에 관련된 기존의 연구에 대해 살펴보고, 3장에서는 본 연구를 통해 제안하는 one-time 전자 서명 기법을 이용한 라우팅 보안 메커니즘에 대해 기술한다. 4장에서는 제안하는 메커니즘의 안전성과 오버헤드 분석 및 성능평가를 위한 시뮬레이션 환경과 시나리오를 설명하고 그 결과를 분석한다. 마지막으로 5장에서는 본 연구의 결론과 향후 연구 방향에 대하여 기술한다.

2. 애드혹 네트워크에서의 라우팅 보안 메커니즘 고찰

이 장에서는 애드혹 네크워크에서 라우팅 메시지를 안전하게 주고받기 위해 제안된 기존의 라우팅 보안 메커니즘에 대해 살펴본다.

먼저 대칭키를 기반으로 라우팅 보안을 제안한 Ariadne 방식은 DSR 라우팅 프로토콜[3]에 보안을 적용한 것으로 Yih-Chun Hu, Arian Perrig, David B. Johnson에 의해 제안되었다[4]. 이는 각 노드가 모든 다른 노드와의 대칭키를 사전에 갖는다는 기본 가정 하에 근원지 노드가 목적지 노드와의 대칭키를 이용하여 경로 탐색을 수행한다. 근원지 노드는 목적지 노드와의 대칭키를 이용하여 MAC(Message Authentication Code)을 생성하고 목적지 노드는 이 값을 이용하여 이를 확인한다. 라우팅 메시지의 무결성을 보장하기위해 단대단 인증을 수행하며 브로드캐스트 인 증시에는 TESLA[5]를 적용한다. 이 방식은 모든 노드 쌍이 대칭키를 갖는다는 것을 기본 가정으로 하고 있지만 실제적으로 이는 네트워크에 미치는 오버헤드가 너무 크다는 문제점을 갖는다.

이들에 의해 제안된 또 다른 보안 라우팅 메커니즘으로 SEAD(Secure Efficient Distance Vector Routing) 방식이 있는데 이는 DSDV 라우팅 프로토콜[6]을 기반으로 하여 hop-by-hop 접근 방식을 택하고 있다[7]. 이는 단방향 해쉬함수를 기반으로 근원지 노드가 해쉬 체인을 생성한 다음 이를 라우팅 정보 변경에 사용한다. 이 인증 방식은 해쉬 체인의 요소를 안전하게 배분할 수 있다는 가정을 기본으로 하고 있으나 trusted CA를 별도로 두기 힘들다는 애드혹 네트워크의 기본적 제약과 맞지 않는다는 문제점을 갖는다.

Zapata와 Asokan은 AODV 방식[8]을 기반으로 역시 PKI[9]를 이용한 보안 기법을 제안하였다[10]. PKI를 기반으로 하여 RREQ 패킷의 필드에 대한 전자 서명을 추가하고, 해쉬 체인을 사용하여 유일하게 변조가 가능한 홉 수 필드를 변조할 수 없도록 함으로써 라우팅 메시지에 보안을 추가하였는데 이 연구 역시 PKI를 기반으로 하고 있어 지나친 오버헤드를 갖는다는 단점을 갖는다.

P. Papadimitrator와 Zygmunt J. Haas에 의해 제안된 SRP (Secure Routing for Mobile ad-hoc Network)는 중간의 악의적인 노드의 행위로 인한 잘못된 Route Reply가 개시 노드까지 전달되는 과정에서 버려지거나 도착 후 무효화되는 특성을 갖는다[11]. 이 방식은 근원지 노드와 목적지 노드 사이에 보안 연계(Security Association)가 존재하며 비밀키 $K_{S,T}$를 공유한다고 가정한다. 이 메커니즘은 MAC 값을 사용하는데 이 값을 계산하는데 많은 처리과정이 필요하지 않기 때문에 비교적 간단하게 메시지의 무결성을 제공할 수 있다. 그러나 웜홀 공격[12]이 가능하며 또한 악의적인 노드가 근원지 노드로부터 받은 패킷을 복사하여 전송할 경우, 그 패킷을 전송하는 중간 노드들의 자원 낭비를 초래한다는 문제점을 갖는다.

Manel Guerrero Zapata와 N.Asokan에 의해 제안된 Secure-AODV(ad-hoc On-Demand Vector routing) 프로토콜은 라우팅 프로토콜 메시지를 보호하기 위해 AODV 프로토콜을 확장시켰다[13]. 이 메커니즘은 각각의 노드가 네트워크의 모든 노드들의 증명된 공개키들을 가지고 있다고 가정한다. 라우팅 메시지를 생성하는 노드는 메시지에 RSA 서명과 해쉬 체인의 마지막 요소를 메시지에 추가한다. 경로 응답은 목적지 노드나 목적지까지의 경로를 가지고 있는 중간 노드들에 의해서 생성되어 근원지 노드에게로 전달된다. 그러나 이 역시 공개키 기반으로 높은 프로세싱 오버헤드를 요구한다는 한계를 갖는다.

Kimaya Sanzgiri 등에 의해 제안된 ARAN이라는 메커니즘은 공개키 기반의 보안 메커니즘으로 각 노드가 자신을 인증하기 위해 전자 서명을 수행하고 다음 노드에 의해 인증받는 형태를 취한다[8]. 이 방식은 기본적으로 인증된 CA가 존재한다고 가정하고 있는데 별도의 CA를 두는 것은 일반적인 애드혹 네트워크 환경에서는 채택하기 힘든 방식이다.

Tirthankar Ghosh, Niki Pissinou, kia Makki는 애드혹 네트워크 내에 존재할 수 있는 악의적인 노드가 단독으로, 혹은 서로 결탁하여 잘못된 라우팅 정보를 만들어 내는 것을 방지

할 수 있는 메커니즘을 제안하였다[15]. 이들은 T-AODV라는 방식을 통해 AODV의 RREQ 패킷 헤더에 추가로 trust-level을 두어 이웃 노드 간에 상호 인증해줌으로써 악의적인 노드의 정보 변경을 막는 방식을 제안하였고 이를 더 발전시켜 MAC을 추가함으로써 악의적인 노드가 결탁하여 라우팅 정보를 와해시키지 못하는 메커니즘을 제안하였는데 이 연구 역시 PKI를 기반으로 하고 있으며 모든 노드가 trust level을 사전에 분배받아야 하는 등 오버헤드가 높다는 문제점을 안고 있다.

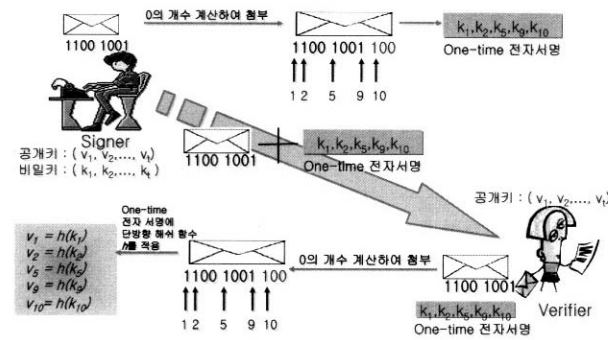
이 밖에도 애드혹 네트워크에서의 안전한 라우팅을 위한 연구는 다양하게 이루어지고 있으나 대부분의 연구가 안전이 보장된 제 삼자를 가정하거나, PKI 방식을 기반으로 하거나, 모든 노드쌍이 대칭키를 소유해야하는 등 오버헤드가 지나치고 애드혹 네트워크의 기본적 성질에 부합되지 않는 방식을 택하고 있다. 이에 본 연구에서는 오버헤드를 줄이면서도 효과적으로 애드혹 라우팅을 지원할 수 있는 방안을 제시한다.

3. One-time 전자 서명을 이용한 라우팅 보안 메커니즘

본 장에서는 안전하며 신뢰성 있는 라우팅을 위해 one-time 전자 서명 기법을 이용한 보안 메커니즘을 제안한다. 공개키 전자 서명의 사용은 비용이 높다는 점과 전자 서명을 생성하고 검증하는 과정에 소요되는 시간이 크다는 단점을 갖는다. 공개키 전자 서명을 라우팅 프로토콜에 사용할 경우 라우팅 메시지의 교환이 자주 일어나기 때문에 전자 서명의 생성과 검증에 시간적인 효율성이 고려되어야 한다[16]. 본 논문에서는 전자 서명 사용을 통한 계산 비용이나 시간 비용을 줄이기 위해 일방향 해쉬 함수(one-way hash function)를 기초로 한 one-time 전자 서명 기법을 사용한다. 제안하는 one-time 전자 서명을 이용한 라우팅 보안 메커니즘은 애드혹 네트워크의 라우팅 프로토콜이 개시되기 전에 전자 서명에 사용될 키 생성을 위해 해쉬 테이블 생성 과정이 수행되며 해쉬 테이블 생성 후에는 각 노드들은 one-time 전자 서명을 포함한 라우팅 메시지 생성을 통해 경로 탐색 및 설정을 수행한다.

3.1 One-time 전자 서명 기법

One-time 전자 서명 기법은 주어진 값을 함수에 적용하여 결과를 얻기는 쉬우나 역방향으로 계산하는 것은 거의 불가능한 공개된 함수 f를 기초로 한다. One-time 전자 서명은 Lamport에 의해 처음으로 소개되었다[17], [18]. 한 개의 비트를 서명하기 위해 비밀키로 사용할 임의의 x_1과 x_2의 값을 선택하고 각각은 0과 1을 표현한다. 비밀키로 선정된 값 x_1과 x_2 각각을 일방향 해쉬 함수에 적용하여 값 $y_1 = f(x_1)$과 $y_2 = f(x_2)$를 얻게 되고 이것을 공개키 요소로 선정하여 공표한다. 한 개의 비트를 서명하기 위해 실제 0과 1에 상응하는 x_1과 x_2의 값을 덧붙임으로써 서명하고, $y_1 = f(x_1)$과 $y_2 = f(x_2)$를 가지고 서명을 검증한다. Lamport의 연구를 동기로 많은 연구자들이 좀 더 효율적인 one-time 전자 서명 기법을 제안하였다. 그 중 Merkle은 Lamport의 방법에서 공개키의 수를 줄이는 향상된 one-time 전자 서명을 제안하였다[19], [20]. 메시지의 각 비트를 서명하기 위해 두 개의 비밀키 x_1, x_2와 두 개의 공개키 $y_1 = f(x_1)$, $y_2 = f(x_2)$를 생성하지 않고, 서명인은 메시지의 각 비트를 서명하기 위해 하나의 비밀키 값 x와 하나의 공개키 값 $y = f(x)$만을 생성한다. 서명할 메시지의 비트들 중에 비트 값이 1인 경우, 서명인은 그 비트에 해당하는 x 값을 메시지와 함께 보낸다. 그러나 만약 서명될 메시지의 비트가 0인 경우는 서명인은 메시지에 어떤 값도 보내지 않는다. 서명을 받은 증명인이 자신이 받은 x에 대하여 받지 않았다고 말하거나 서명된 메시지에서 1의 개수를 변경하는 등의 행동을 할 수 없도록 서명인은 메시지에 포함된 0의 개수를 계산하여 메시지와 함께 서명한다. n 비트의 메시지를 서명하는데 에는 0의 개수를 추가하기 위하여 $log_2 n$ 비트가 메시지에 추가된다.



(그림 1) One-time 전자 서명을 포함한 메시지의 서명과 검증 과정

3.2 제안하는 메커니즘

본 논문에서 제안하는 메커니즘은 애드혹 라우팅 프로토콜에 보안을 제공하기 위하여 위에서 설명한 one-time 전자 서명을 사용한다. 제안하는 one-time 전자 서명은 애드혹 네트워크의 각 노드들이 자신의 공개키를 네트워크의 다른 노드들에게 공표하기 위해 사용할 수 있는 공개키 기반 암호 시스템이 있다고 가정한다. 그러나 공개키 기반 전자 서명 기법의 높은 계산 비용을 줄이기 위해 본 논문에서는 공개키 기반 시스템을 라우팅 메시지를 서명하기 위해 직접적으로 사용하지는 않는다. 대신에 one-time 키 분배를 위해 한번 사용할 뿐이다. M_i를 노드가 전송하고자 하는 i번째 라우팅 메시지라 하고, 해쉬 함수 f와 h는 애드혹 네트워크에 참여하는 모든 노드들에게 알려진 일방향 해쉬 함수라고 하자. 메시지 M_i에 해쉬 함수 f를 적용하여 M_i의 해쉬 값 $f(M_i)$를 얻는다. 이 해쉬 값 $f(M_i)$는 메시지 M_i의 인증과 무결성을 제공하기 위하여 서명된다. 해쉬 함수 f의 결과물이 l비트라고 하면, $f(M_i)$를 서명하기 위해 $n(= l + [log_2 l]+ 1)$개의 one-time 공개키 요소들이 필요하다. One-time 전자 서명을 이용한 메시지의 서명과 검증 과정이 (그림 1)에 설명되어 있다.

본 연구에서는 라우팅 메시지의 해쉬 값을 얻기 위해 MD5

라는 해쉬 함수[21]를 적용한다. MD5는 임의의 길이의 입력 메시지로부터 128비트 메시지 다이제스트를 만드는 해쉬 알고리즘이다. 본 논문에서는 애드혹 라우팅 프로토콜이 하나 이상의 메시지를 서명하기 위하여 one-time 공개키 요소들이 여러 세트 필요하다. One-time 공개키 요소들을 여러 세트 유지하려면 애드혹에 참여하는 호스트들에게 메모리의 요구 사항이 지나치게 커질 수 있다. 따라서 본 논문에서는 공개키 요소의 첫 세트를 반복적으로 해쉬 함수인 MD5에 적용함으로써 해쉬 체인을 생성하고, 생성된 해쉬 체인으로부터 공개키 요소들의 여러 세트를 유도한다. 이렇게 함으로써, one-time 공개키 요소들과 비밀키 요소들은 서로 연관되어진다. 즉, 비밀키 요소들의 i번째 요소들은 공개키 요소들의 $(i+1)$번째 요소들로 사용된다.

3.2.1 One-time 키 생성을 위한 해쉬 테이블 생성

하나의 비트열 x로부터 길이 n인 해쉬 체인 $h^1(x)$, …, $h^i(x)$, … , $h^n(x)$를 만들기 위해서는 다음과 같은 절차를 수행한다. h는 MD5 해쉬 함수이고 i = 1, …, n 일 때, $h^0(x) = x$, $h^1(x) = h(h^0(x))$이고 $h^i(x) = h(h^{i-1}(x))$이다. n 비트인 k 개의 메시지를 one-time 전자 서명 기법을 이용하여 서명하기 위해 애드혹 네트워크에 참여하는 각 호스트들은 다음과 같이 해쉬 테이블을 생성한다.

① j = 1, …, n 일 때 각 호스트들은 임의로 비밀키 요소들인 x_j를 선택한다.

② 각 호스트들은 n 개의 비밀키 요소들 x_j 각각에 대해 길이 k인 해쉬 체인을 만듦으로써 해쉬 테이블을 생성한다. 해쉬 테이블의 포맷은 다음 (그림 2)와 같다.

0	$h^0(x_1)$,	$h^0(x_2)$,	….	$h^0(x_n)$
1	$h^1(x_1)$,	$h^1(x_2)$	….	$h^1(x_n)$
⋮	⋮	⋮	⋮	⋮
k	$h^k(x_1)$,	$h^k(x_2)$,	….	$h^k(x_n)$

(그림 2) 각 호스트가 생성하는 해쉬 테이블

③ 각 호스트들은 공개키 기반 암호 시스템을 사용하여 위의 해쉬 테이블의 k번째 행을 자신의 비밀키로 서명하여 브로드캐스트한다.

④ 각 호스트들은 다른 호스트들로부터 받은 서명된 $h^k(x_j)$ 값을 공개키 기반 암호 시스템을 사용하여 검증한 후 올바른 서명인 경우 j = 1, …, n 인 v_j를 저장한다. v_j는 해당 호스트들의 one-time 공개키 요소들로 사용된다.

3.2.2 One-time 전자 서명을 사용한 경로 탐색

근원지 노드가 다른 노드와 통신을 하고 싶지만 자신의 라우팅 테이블에 통신을 하고자 하는 목적지 노드까지의 라우팅 정보가 없는 경우 근원지 노드는 경로 탐색을 시작해야 한다. 근원지 노드는 자신의 이웃 노드들에게 경로 탐색 요청 메시지인 RREQ 메시지를 브로드캐스트함으로써 목적지 노드까지의 경로 탐색을 시작한다. 경로 탐색 요청 메시지는 해당 노드의 브로드캐스트 ID, hop count, 근원지 노드의 IP 주소, 근원지 노드의 sequence number, 목적지 노드의 IP 주소, 목적지 주소의 sequence number, counter, RREQ 패킷의 one-time 전자 서명을 포함한다. 경로 탐색 요청 메시지 RREQ의 구조는 (그림 3)과 같다.

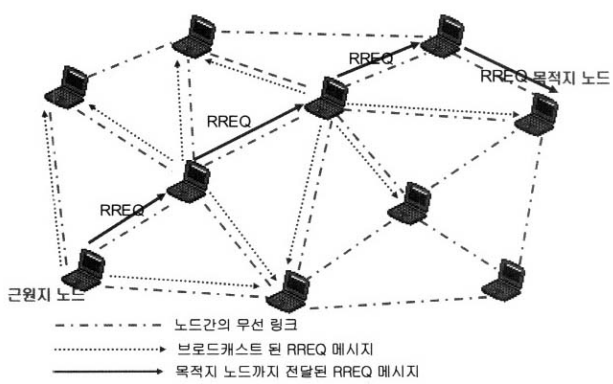
Broadcast ID	Hop Count
근원지노드의 IP주소	근원지노드의 Sequence number
목적지노드의 IP주소	목적지노드의 Sequence number
Counter	One-time 전자 서명

• **Counter** – 해당 노드가 발송한 메시지의 개수

(그림 3) 경로 탐색 요청 메시지 RREQ의 구조

근원지 노드는 자신이 보내려는 i(=counter)번째 RREQ 메시지를 서명하기 위해 Merkle의 one-time 전자 서명과 같은 방법으로 RREQ 메시지를 해쉬 함수인 MD5에 적용하고, 그 결과 $f(\text{RREQ}_i)$를 얻는다. 그 후, $f(\text{RREQ}_i)$에서 0의 개수를 계산하여 $f(\text{RREQ}_i)$에 덧붙이고 n 비트의 비트 스트링 g를 생성한다. g_j를 비트 스트링 g의 j번째 비트라고 할 때, g_j=1인 모든 j에 대하여 해쉬 테이블의 $(k-i)$번째 행에서 $h^{k-i}(x_j)$ 해쉬 값을 RREQ 메시지에 덧붙임으로써 one-time 전자 서명을 형성하고 근원지 노드는 서명한 RREQ 메시지를 이웃 노드들에게 브로드캐스트한다.

RREQ 메시지를 받은 노드는 받은 RREQ 메시지의 전자 서명을 검증하기 위해 받은 RREQ 메시지를 해쉬 함수인 MD5에 적용하여 $f(\text{RREQ}_i)$를 얻고, $f(\text{RREQ}_i)$에서 0의 개수를 계산하여 $f(\text{RREQ}_i)$에 덧붙인 후 n 비트의 비트 스트링 g를 얻는다. g_j=1인 모든 j에 대하여 $h^{i-i'}(r_j) = v_j$ 인지를 체크한다. r_j는 현재 받은 RREQ 메시지에 덧붙여져 one-time 전자 서명으로 받은 값이고, v_j는 기존의 RREQ i'번째 메시지에 대한 one-time 전자 서명 값으로 현재 RREQ 메시지를 받은 노드가 이전에 받아 저장하고 있는 one-time 전자 서명 값이다. 만일 $h^{i-i'}(r_j) = v_j$ 가 참이라면 RREQ 메시지는 RREQ를 서명한 노드로부터 전달된 것으로 인증되고, 또한 메시지의 무결성까지 보장하게 되며, 이를 통해 다른 노드인 척 가장하



(그림 4) 경로 탐색 과정

여 거짓 라우팅 정보를 유포하거나 원래의 라우팅 정보를 임의로 변조하는 것을 막을 수 있다. RREQ를 검증한 노드는 다음 메시지의 검증을 위해 현재의 v_i를 r_i로 갱신한다. (그림 4)에서와 같이 근원지 노드에서 생성된 RREQ 메시지가 목적지 노드에 도달할 때까지 경로 상에 있는 각각의 노드들은 위와 같이 이전 노드로부터 받은 메시지의 서명을 검증하고 다시 자신이 서명을 하여 메시지를 포워딩하는 절차를 반복적으로 수행한다.

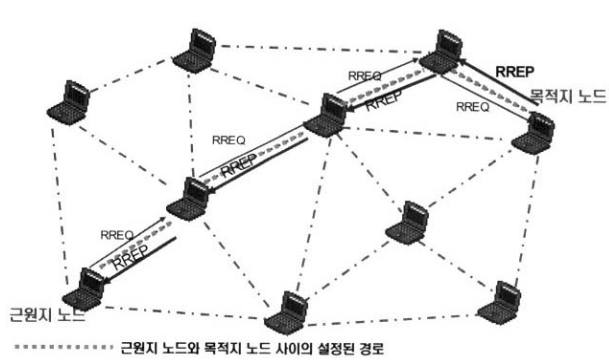
3.2.3 One-time 전자 서명을 사용한 경로 설정

근원지 노드가 보낸 경로 탐색 요청 메시지 RREQ가 목적지 노드에 도착하면 목적지 노드는 경로 탐색 응답 메시지 RREP를 생성하여 RREQ가 지나온 근원지 노드로부터 목적지노드까지 경로로 보낸다. 경로 탐색 응답 메시지의 포맷은 다음 (그림 5)와 같다.

Hop Count	Lifetime
근원지노드의 IP주소	
목적지노드의 IP주소	목적지노드의 Sequence Number
Counter	One-time 전자 서명

(그림 5) 경로 탐색 응답 메시지 RREP의 구조

목적지 노드가 i 번째 RREP 메시지를 근원지 노드까지 전달하기 위해 메시지를 서명하고 중간에 있는 각 노드가 이를 검증하는 과정은 RREP 메시지의 전달 과정에서와 동일하다. 이러한 전자 서명을 통해 메시지의 인증 및 무결성, 재생 공격을 막을 수 있다. 근원지 노드까지 경로 요청 응답 메시지가 도착하면 근원지 노드로부터 목적지 노드까지 경로가 확정되므로 근원지 노드는 인증을 거쳐 생성된 이 경로를 통해 안전하게 데이터를 보낼 수 있다. 이러한 경로 설정 과정은 (그림 6)에 나타나있다.



(그림 6) 경로 설정 과정

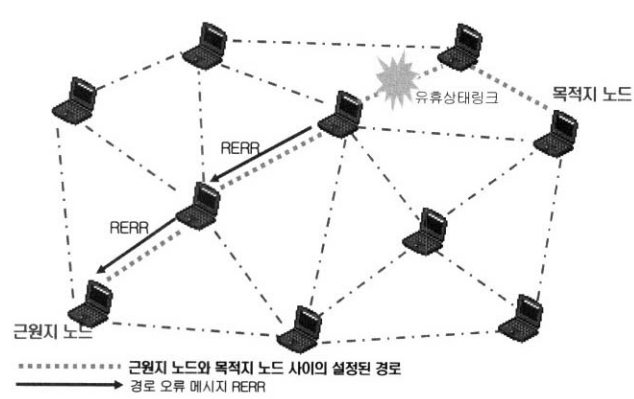
3.2.4 One-time 전자 서명을 사용한 경로 유지 및 관리

본 논문의 라우팅 프로토콜은 요구 기반(on-demand) 구동방식으로서 각 노드들은 설정된 경로가 현재 유효한 경로인지를 주기적으로 확인해야 한다. 경로의 lifetime 동안 존재하는 경로 상에 데이터 트래픽이 오고가지 않는다면, 노드는 자신의 라우팅 테이블에 간단히 그 경로가 활동하지 않는다고 체크하면 된다. 그러나 유효하지 않은 경로로부터 데이터가 전달되면 노드는 근원지 노드를 향하는 역경로를 따라 경로 오류 메시지 RERR을 생성하여 보낸다. 또한 노드들의 이동성 때문에 활동 중인 경로상의 링크가 끊어진 경우도 경로 오류 메시지를 생성하여 근원지 노드에게 보낸다. 경로 오류 메시지의 포맷은 다음 (그림 7)과 같다.

DestCount	
도착 불가능한 노드의 IP주소 (1)	도착 불가능한 노드의 Sequence number (1)
도착 불가능한 노드의 IP주소 (2)	도착 불가능한 노드의 Sequence number (2)
••••	••••
Counter	One-time 전자 서명

• DestCount – 도착 불가능한 노드의 개수
• Counter – 해당 노드가 발송한 메시지의 개수

(그림 7) 경로 오류 메시지 RERR 의 구조



(그림 8) 경로 유지 및 관리 과정

RERR 메시지가 근원지 노드에 도달할 때까지 경로 상에 있는 각각의 노드들은 RREQ, RREP 메시지와 마찬가지로 이전 노드로부터 받은 메시지의 서명을 검증하고 다시 자신이 서명을 하여 메시지를 포워딩 하는 절차를 반복적으로 수행한다. 실제로는 경로가 유효함에도 악의적인 노드가 경로가 끊어졌다고 거짓된 RERR 메시지를 위조하여 발송할 경우 이러한 악의적인 노드의 행동을 탐지하는 것은 매우 어렵다. 그러나 본 메커니즘에서는 RERR 메시지가 서명되어 발송되기 때문에 악의적인 노드가 다른 노드인 척 가장하여 경로 에러 메시지를 생성하는 공격을 막을 수 있다. 경로 유지 및 관리 과정은 (그림 8)에 나타나 있다.

4. 성능 평가

본 장에서는 제안하는 one-time 전자 서명을 이용한 라우팅 보안 메커니즘을 평가하기 위한 시뮬레이션 모델을 설명하고 그 결과를 분석한다. 4.1절에서는 제안하는 메커니즘에 대한 안전성 분석을 하고, 4.2절에서는 부가되는 오버헤드를 분석하며, 4.3에서는 시뮬레이션 환경 및 시나리오를 제시한 후, 마지막 절인 4.4절에서는 실험 결과 및 연관성을 토대로

성능 평가 및 안전성 결과 분석을 기술한다. 키 생성이나 분배 등은 오프라인으로 이루어지는 경우가 많아 기존의 공개키 방식이나 대칭키 방식과 제안 메커니즘과의 시뮬레이션 상의 비교가 용이하지 않으므로 본 성능 평가에서는 4.2절에서의 오버헤드 분석으로 이를 대신하였고, 시뮬레이션은 기본 AODV 방식에 제안 메커니즘을 적용했을 때 어느 정도의 추가적 오버헤드가 발생하는지와 이동성에 따른 변화 등을 보인다.

4.1 안전성 분석

애드혹 네트워크의 라우팅 프로토콜에 존재하는 보안상 취약점에 대하여 본 논문에서 제안하는 메커니즘이 갖는 강건성을 평가함으로써 안정성 분석을 한다.

- RREQ, RREP 메시지의 스푸핑(spoofing) 공격 : 네트워크에 참여하는 노드들은 자신의 비밀키 값을 가지고 서명을 해야 하므로 경로 탐색 과정에서 다른 노드들이 근원지 노드인 척 가장하여 행동할 수 없다. 또한 경로 설정 과정에서도 목적지 노드만이 자신의 비밀키 값을 가지고 전자 서명을 할 수 있으므로, 다른 노드가 목적지 노드를 대신하여 경로 탐색 요청에 응할 수 없다. 이처럼 본 메커니즘은 근원지 노드 또는 목적지 노드를 가장하여 경로 탐색이나 경로 설정을 할 수 없으므로 위장 공격을 방어함을 알 수 있다.
- 위조된 라우팅 메시지 : 메시지는 오직 네트워크 참여시 자신의 공개키를 다른 노드에게 공표한 노드만이 위조할 있다. 이 경우, 본 메커니즘은 악의적인 노드가 라우팅 메시지를 위조하여 보내는 것을 방어할 수 없다. 그러나 악의적인 노드가 서명한 공개키를 통해 위조된 라우팅 메시지를 보낸 노드를 알 수 있으므로 만약 어떠한 노드가 계속해서 네트워크에 거짓된 라우팅 메시지를 위조하여 보낸다면 그 노드는 추후에 경로 탐색 과정에서 제외되도록 할 수 있을 것이다. 또다른 위조의 방법으로는 각 노드가 사용한 비밀키 요소를 재사용하여 중간에서 이를 위조하는 방법이 있는데 본 연구에서 제안된 공개키 및 비밀키 요소는 일회성을 가지므로 노드가 포획되어 해쉬 테이블이 완전히 노출되지 않는 한 키의 재사용으로 인해 발생할 수 있는 보안상의 문제점을 완벽히 차단할 수 있다.
- 라우팅 메시지의 변조 : 제안한 메커니즘에서는 경로 탐색 요청 메시지 RREQ나 경로 탐색 응답 메시지 RREP가 근원지 노드와 목적지 노드까지의 설정된 경로에서 변조되는 것을 방지할 수 있다. 이는 두 메시지 모두 근원지 노드 혹은 목적지 노드에 의해 서명이 되기 때문에 중간 경로에서 통신 중 메시지의 일부가 변조되면 중간 노드에 의해 탐지되고, 변조된 메시지는 폐기되기 때문이다. 본 논문에서는 고려하고 있지 않지만, 만약 어떠한 노드가 계속해서 네트워크에 거짓된 라우팅 메시지를 변조하여 보낸다면 그 노드는 추후 경로 탐색 과정에서 제외되도록 할 수 있다.

4.2 오버헤드 분석

라우팅 메시지를 전달하는 과정에서 한 홉을 경우할 때마다 one-time 전자 서명을 수행함에 따라 그렇지 않은 경우에 비해 시간상, 공간상의 오버헤드가 발행하게 된다. 이를 분석하면 다음과 같다.

4.2.1 시간 오버헤드

먼저 one-time 전자 서명과 검증에 필요한 시간상의 오버헤드를 분석해보면 Pentium IV(2.4 GHz) 프로세서, 512MB RAM인 컴퓨터에서 Red Hat Linux 9.0의 운영체제 기반하에 MD5를 이용한 one-time 전자 서명 생성과 검증 알고리즘을 여러 번 수행하여 평균해 보면 one-time 전자서명의 생성에는 1.75ms, 전자 서명의 검증은 0.1ms의 수행 시간이 필요함을 알 수 있다. 이는 평균적으로 보았을 때 가장 간단한 PKI 기반의 전자 서명과 검증에 비해 10배 이상의 짧은 시간을 요하며 one-time 전자 서명을 사용하는 경우 새로운 해쉬 테이블을 생성하여 마지막 행의 공개키 요소를 브로드캐스트할 경우에만 PKI 기반 전자 서명을 필요로 하므로 일반적 서명 시에는 상대적으로 훨씬 짧은 시간만을 필요로 한다. 결과적으로 라우팅 설정 시의 시간을 훨씬 단축시킬 수 있다는 장점을 갖는다.

추가적으로 필요한 시간상의 오버헤드는 각 노드가 생성한 해쉬 테이블의 공개키 요소를 전부 사용하고 난 후 새로운 테이블을 생성할 때 테이블의 행 수 만큼의 해쉬함수를 적용해야 하는 경우이다. 위에 언급된 컴퓨터 환경에서 MD5를 사용하는 경우 1초에 백만회 이상의 MD5 해쉬함수 수행이 가능하므로 해쉬 테이블 생성을 위해 각 노드에 부가되는 시간상의 오버헤드는 크지 않으며 특히 테이블의 크기에 따라서는 거의 무시할 수 있는 정도이므로 최근의 일반적 노트북 사양에서는 충분히 이를 처리할 수 있다고 본다.

4.2.2 공간 오버헤드

다음으로 기억 공간 상의 오버헤드를 살펴보면, 해쉬 테이블을 저장하기 위한 기억 공간은 해쉬함수 f를 적용한 결과가 l비트이고 h를 적용한 결과가 m비트라면 $(l + [log_2 l]+ 1)$ x m 비트만큼의 one-time 공개키 요소들이 필요하며 MD5를 적용하는 경우 이는 2KB정도의 저장 공간을 필요로 하므로 k를 해쉬 테이블의 길이라 할 때 각 노드는 $2k$KB의 공간적 오버헤드를 갖는다. One-time 전자 서명은 라우팅 메시지의 서명에만 적용되므로 노드의 이동이 심하지 않은 경우 k값을 너무 크게 설정하지 않는다면 이로 인한 공간적 낭비요소는 훨씬 줄어들 수 있다. 단, 해쉬 테이블의 길이가 짧으면 새로운 경로 설정이 자주 필요한 경우 테이블이 일찍 소진되어 새로 테이블을 생성해야하는 계산상의 오버헤드가 있을 수 있으므로 이들 간의 trade-off를 고려하여 테이블의 길이를 적절하게 유지한다면 네트워크의 효율성을 더욱 높일 수 있다.

대칭키 방식의 전자 서명과 비교해 보면 본 연구에서 제안한 방식의 공간적 이득은 더욱 부각된다. 대칭키 방식의 경우 자신을 제외한 모든 노드와의 대칭키를 가져야하므로 $n(n-1)/2$개의 키가 필요하여 기억 공간의 낭비가 초래될 뿐 아니라 노드수가 증가할수록 이에 비례하여 오버헤드는 더욱

커진다. 또한 키가 노출되는 경우 이를 새로이 공유해야 하는데 trusted CA가 없는 애드혹 환경에서는 쉽지 않은 문제이다. 제안하는 방법은 기억 공간을 효율적으로 줄이면서도 노드수가 증가하여도 기억 공간이 추가적으로 필요치 않다는 장점을 가지며 각자가 해쉬 테이블을 생성하므로 CA가 없는 애드혹 환경에 적합한 방식이라 할 수 있다.

4.2.3 계산 오버헤드

본 연구에서 제안한 방법은 각 노드별로 해쉬 테이블을 별도로 생성하고 공개키 요소로 사용될 새로운 해쉬 테이블의 최종값을 브로드캐스트할 때만 공개키 계산을 수행하므로 one-time 전자 서명에 필요한 계산량보다 훨씬 높은 오버헤드를 갖는 공개키 계산량을 획기적으로 줄일 수 있다. 또한 해쉬함수의 특성을 사용하면 서명된 라우팅 메시지를 전달받은 노드가 마지막으로 받은 전자 서명값을 메시지와 함께 전달된 서명값으로 갱신하여 유지함으로써 다음 메시지를 검증할 때 필요한 해쉬 계산량을 더욱 줄일 수 있다. 특히 각 노드의 공간적 오버헤드와의 trade-off를 고려하여 해쉬 테이블의 길이를 길게 유지한다면 새로운 테이블의 공개키 요소를 브로드캐스트하는 주기를 길게 가져감으로써 공개키 계산 회수를 더욱 줄일 수 있다.

4.3 시뮬레이션 환경 및 시나리오

시뮬레이션은 Linux 9.0 / C 환경에서 GloMoSim 라이브러리를 사용하여 구현한다. <표 1>은 본 시뮬레이션에 사용된 파라미터 값들이다.

4.3.1 일반적인 경우의 효율성 측정 기준

• 패킷 전달률(Packet Delivery Fraction)

CBR 세션을 처음 시작한 근원지 노드가 생성하여 전송한 데이터 패킷이 목적지까지 전달된 비율을 측정한다. 이는 라우팅 프로토콜이 얼마나 정확한 경로 탐색을 제공하는지 측정하기 위함이다.

〈표 1〉 시뮬레이션에서 사용하는 파라미터

파라미터	수준
Area	670m X 670m / 1000m X 1000m
Speed	Uniformly distributed between 0 and 10m/s
Radio Range	250m
Placement	randomly
Movement	Random waypoint mobility
MAC	802.11
Number of nodes	20 nodes / 50 nodes
Sending Capacity	2 Mbps
Application	CBR
Simulation Time	900s

• 라우팅 오버헤드(routing overhead)

CBR 세션 동안 전달된 데이터 패킷에 대한 라우팅 메시지의 비율을 라우팅 오버헤드로 측정한다. 이는 제안 메커니즘이 데이터 패킷을 전달하기 위해 얼마나 많은 라우팅 패킷을 주고받는지를 측정하고 이를 통해 라우팅 프로토콜이 타당한 라우팅 오버헤드를 갖는지를 살펴본다.

• 평균 경로 길이(average path length)

AODV와 제안한 메커니즘 각각에 의해 설정된 경로의 평균적인 홉 수를 측정한다.

4.3.2 악의적인 노드가 존재할 경우의 효율성 측정 기준

AODV와 제안한 메커니즘을 악의적인 노드가 존재할 경우를 고려하여 실험한다. 시뮬레이션 동안 악의적인 노드는 자신을 지나가는 RREQ와 RREP 메시지의 hop count 필드를 0으로 변조함으로써 악의적인 노드 자신이 근원지 노드나 목적지 노드로부터 한 홉 떨어진 거리에 존재한다고 거짓 신고한다. 이러한 악의적인 노드의 행동의 목적은 라우팅 프로토콜의 취약점을 이용하여 경로 탐색을 수행할 때 자신이 경로상에 놓이도록 하는 것이다. 이렇게 함으로써 악의적인 노드는 근원지 노드와 목적지 노드 사이에 오고 가는 데이터 패킷을 볼 수 있고 이를 통해 메시지 변조와 같은 잠재적 공격 가능성이 드러난다고 할 수 있다. 악의적인 노드가 존재할 경우의 측정 기준은 다음과 같다.

• 평균 경로 길이(average path length)

악의적인 노드는 AODV 프로토콜을 사용할 경우 라우팅 메시지의 위조를 통해 경로 설정 시 자신을 포함하도록 할 것이고 그 결과 경로 길이가 증가할 것이다. 제안한 메커니즘에서는 라우팅 메시지의 위조가 불가능하므로 경로 길이의 변화가 없음을 확인할 수 있다. 또한 악의적인 노드의 수에 따라 경로 길이가 얼마나 증가하는지를 알아본다. 이 값은 라우팅 프로토콜의 경로 탐색 결과 긴 경로가 선택될 경우, 라우팅 오버헤드와 데이터 패킷의 통신 지연이 커진다는 점에서 중요하게 살펴보아야 한다.

• 악의적인 노드를 거쳐간 데이터 비율

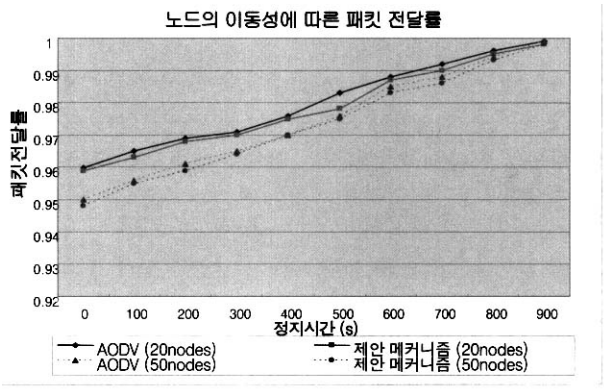
이 측정값은 AODV와 제안한 메커니즘에 대해 악의적인 노드의 분포정도에 따라 전체 데이터 패킷 중 악의적인 노드를 거쳐간 데이터 패킷이 차지하는 비율을 나타낸다. 악의적인 노드를 지나온 데이터 패킷은 악의적인 노드에 의해 잠재적으로 변조되거나 다른 노드인 척 가장하는 스푸핑 공격의 가능성을 갖는다. 그러므로 이 수치를 살펴봄으로써 악의적인 노드의 공격에 대하여 라우팅 프로토콜이 얼마나 안전한지 알 수 있다.

4.4 시뮬레이션 결과

4.4.1 패킷 전달률 (Packet Delivery Fraction)

(그림 9)는 정지 시간(pause time)에 따른 평균 패킷 전달률을 보여준다. 정지 시간이란 이동 노드들이 무작위로 어느 지점으로 이동하여 다음 목적지로 이동할 때까지 그곳에서 머무르는 시간으로 정지 시간이 0인 경우 노드는 연속적으로

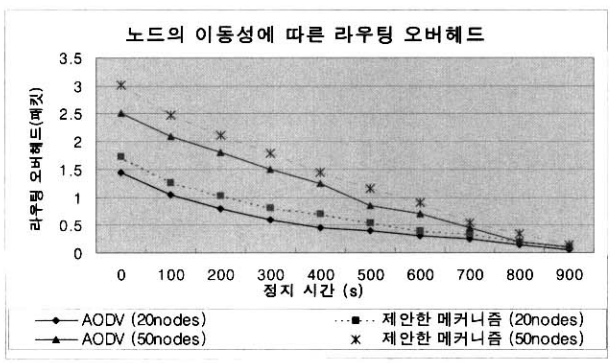
이동하는 것이고 정지 시간이 시뮬레이션 시간과 동일하면 노드가 시뮬레이션 기간 동안 이동하지 않았음을 의미한다. (그림 9)에서 볼 수 있는 것처럼 노드의 이동성이 작을수록 패킷이 안정적으로 목적지까지 도착함을 알 수 있다. 또한 제안 메커니즘은 AODV를 사용하여 라우팅을 수행 후 데이터를 주고받는 것과 거의 유사한 데이터 전달률을 갖는 것을 볼 수 있다. 이는 제안 메커니즘이 데이터 패킷을 주고받기 위해 경로를 탐색하고 설정하는 것이 매우 효율적이고 정확하다는 것을 설명해준다.



(그림 9) 노드의 이동성에 따른 패킷 전달률

4.4.2 라우팅 오버헤드 (routing overhead)

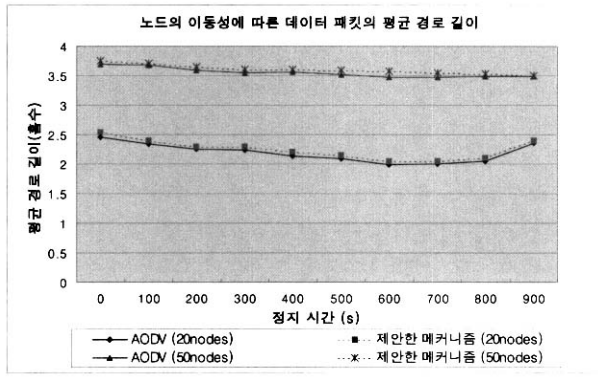
(그림 10)은 노드의 이동성에 따른 라우팅 오버헤드를 보여준다. 이는 목적지 노드까지 전달된 데이터에 대하여 경로 탐색과 설정을 위해 사용된 라우팅 메시지의 비율을 나타낸다. 결과를 살펴보면, 제안 메커니즘은 라우팅 메시지에 one-time 전자 서명을 포함하므로 기존의 AODV 라우팅 프로토콜에 비하여 경로 탐색을 위하여 더 큰 라우팅 메시지가 오고가게 된다. 이러한 점에서 AODV는 라우팅 과정 중에 적은 바이트로 구성된 라우팅 메시지를 주고받는다는 장점을 갖는다. 그러나 AODV와 제안한 메커니즘 모두 경로 탐색을 위하여 주고받는 메시지의 수는 거의 같다. (그림 10)에서 보는 바와 같이 제안한 메커니즘은 시뮬레이션 과정 중에 주고받은 데이터 패킷 수에 대하여 경로 탐색을 위해 주고받은 라우팅 패킷 수는 기존의 AODV와 거의 유사함을 알 수 있다.



(그림 10) 노드의 이동성에 따른 라우팅 오버헤드

4.4.3 평균 경로 길이(average path length)

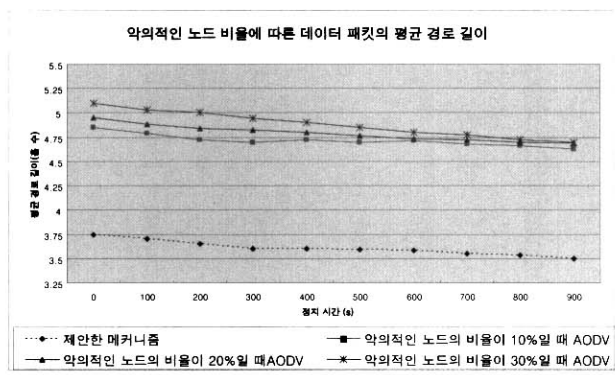
(그림 11)은 노드의 이동성에 따른 데이터 패킷의 평균 경로 길이를 보여준다. 이는 목적지 노드에 도착한 데이터 패킷이 거쳐온 홉 수의 평균값을 계산함으로써 얻은 결과이다. 제안 메커니즘과 AODV 프로토콜이 거의 흡사한 평균 경로 길이를 나타낸다. 이것은 제안한 메커니즘이 가장 짧은 최단 경로를 찾는다고 말할 수는 없지만, 목적지까지 도착한 첫 경로 탐색 요청 메시지인 RREQ가 대부분의 경우 AODV와 유사하게 최단 경로를 따라 전달되었다는 의미이다. 따라서 제안 메커니즘은 최단 경로를 찾는데 있어서 AODV와 유사한 성능을 보임을 알 수 있다.



(그림 11) 노드의 이동성에 따른 데이터 패킷의 평균 경로 길이

4.4.4 악의적 노드의 비율에 따른 평균 경로 길이

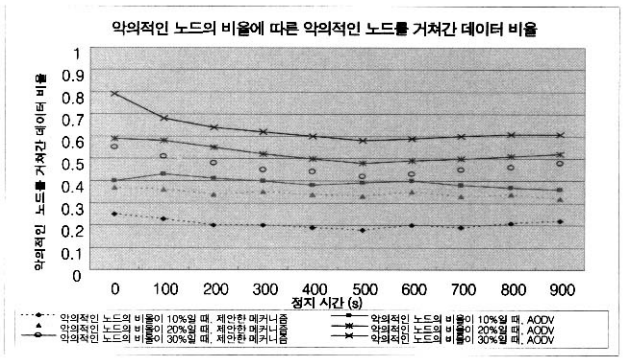
(그림 12)는 악의적인 노드 비율에 따른 데이터 패킷의 평균 경로 길이를 보여 준다. 악의적인 노드는 자신이 경로 설정 과정에서 꼭 포함되도록 함으로써 최단 경로를 선택하지 못하는 결과를 초래할 수 있다. (그림 12)에서 보는 바와 같이 제안한 메커니즘은 악의적인 노드의 비율과 상관없이 10%의 악의적인 노드를 갖는 AODV보다 더 짧은 경로를 선택함을 볼 수 있다. 따라서 제안 메커니즘은 one-time 전자 서명을 이용함으로써 악의적인 노드가 위조한 패킷을 미리 탐지하고 악의적인 노드가 네트워크에 보내는 잘못된 라우팅 정보를 걸러냄으로써 AODV 라우팅 프로토콜보다 안전함을 살펴볼 수 있다.



(그림 12) 악의적인 노드 비율에 따른 데이터 패킷의 평균 경로 길이

4.4.5 악의적 노드의 비율에 따른 악의적인 노드를 거쳐간 데이터 비율(Fraction of Data Packets which passed through malicious Nodes)

(그림 13)은 AODV와 제안 메커니즘에 대해 악의적인 노드의 분포정도에 따라 전체 데이터 패킷에 대해 악의적인 노드를 거쳐간 데이터 패킷이 차지하는 비율로서 이 값이 작을수록 악의적인 노드의 공격에 대해 안전하게 경로 탐색을 수행한다는 것을 알 수 있다. (그림 13)을 살펴보면, 제안 메커니즘에 비해 AODV는 훨씬 많은 데이터 패킷이 악의적인 노드를 통과하는 것을 보여준다. 노드들의 이동성이 없고 네트워크 내에 악의적인 노드가 10%일 경우 AODV는 36%의 데이터 패킷이 악의적인 노드를 통과한데 반해 제안한 메커니즘은 22%만이 악의적인 노드를 통과한다. 이는 악의적인 노드를 고려하지 않은 AODV 라우팅 프로토콜에서는 라우팅 과정 중에 악의적인 노드가 자신을 통과하는 RREQ와 RREP 메시지의 hop count 필드를 0으로 변조함으로써 자신을 통과하도록 공격을 수행한 결과를 보여주는 것이다. 시뮬레이션 결과 네트워크 내에 악의적인 노드의 비율이 높아질 경우 제안한 메커니즘은 AODV에 비해 더욱 안전성을 높일 수 있음을 알 수 있다.



(그림 13) 악의적인 노드를 거쳐간 데이터 비율

5. 결론 및 향후 연구

공개키 전자 서명의 사용은 비용이 비싸다는 점과 전자 서명을 생성하고 검증하는 과정에 소요되는 시간이 크다는 단점을 갖기 때문에 본 논문에서는 전자 서명을 통하여 인증과 메시지의 무결성을 제공하되 전자 서명의 생성과 검증의 효율성을 높일 수 있도록 하기위해 one-time 전자 서명을 이용한다. 제안된 one-time 전자 서명은 애드혹 네트워크에서의 라우팅 메시지를 인증하고 메시지에 무결성을 제공한다. 시뮬레이션과 보안 분석을 통해 제안 메커니즘은 기존의 보안을 고려하지 않은 라우팅 메커니즘에 비해 상대적으로 높은 라우팅 오버헤드를 갖지만, 경로를 탐색하고 설정하는 과정에서 악의적인 노드의 공격에 대항하여 안전성을 가짐으로써 네트워크의 보안을 향상시킴을 알 수 있다.

향후 연구에서는 악의적인 노드가 존재할 때 이들의 행동을 효과적으로 탐지하는 방법 및 이러한 노드를 네트워크에서 고립시켜 전체 네트워크의 효율성을 더욱 높이는 방안에 관한 연구와, 노드가 포획되어 해쉬 테이블이 노출되었을 때 이로 인한 보안상의 위협을 효과적으로 차단할 수 있는 방안에 관한 연구가 필요할 것으로 보인다.

참 고 문 헌

[1] C. E. Perkins, "ad-hoc Networking," New York, Addison-Wesley, 2001.

[2] Yih-Chun Hu, Adrian Perrig, "A survey of secure wireless ad hoc routing," Wireless Networks, Vol.11, pp.21-38, January, 2005.

[3] D. B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," In Proc. IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'94), IEEE Press, 1994, pp.158-163.

[4] Yih-Chun Hu, Adrian Perrig, David B. Johnson, "Ariadne : A Secure On Demand Routing Protocol for Ad Hoc Networks," Wireless Networks, Vol.11, pp.21-38, January, 2005.

[5] A. Perrig et al., "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," In Proc. IEEE Symp. Security and Privacy, IEEE Press, 2000, pp.56-73.

[6] C. E. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," In Proc. SIGCOMM '94 Conf. Communications Architectures, Protocols and Applications, ACM Press, 1994, pp.234-244.

[7] Yih-Chun Hu David B. Johnson Adrian Perrig, "SEAD : Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks," In Fourth IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'02), June, 2002.

[8] C.E. Perkins and E.M. Royer, "Ad-Hoc On-Demand Distance Vector Routing," In Proc. of 2nd IEEE Workshop Mobile Computing Systems and Applications(WMCSA'99), IEEE Press, 1999, pp.90-100.

[9] R.C. Merkle, "Protocols for Public Key Cryptosystems," In Proc. IEEE Symp. Research in Security and Privacy, IEEE Press, 1980, pp.122-133.

[10] C. Perkins and E. Royer, "Ad hoc On-Demand Distance Vector Routing," In Proc. IEEE Workshop on Mobile Computing Systems and Applications, 1999.

[11] P. Papadimitratos and Z. J. Haas, "Secure Routing for Mobile ad-hoc Networks," SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002), San Antonio, TX, January, pp.27-31, 2002.

[12] Y. C. Hu, A. Perrig, and D.B. Johnson, "Packet Lea-shes :

A Defense against Wormhole Attacks in Wireless Ad HocNetworks," In Proc. of 22nd Ann. Joint Conf. IEEE Computerand Communications Societies (INFOCOM 2003), IEEE Press, 2003, pp.1976-1986.

[13] Manel Guerrero Zapata and N. Asokan, "Securing ad- hoc Routing Protocols," In Proc. of ACM Workshop on Wireless Security (WiSe 2002), pp.1-10, September, 2002.

[14] Kimaya Sanzgiri et al, "A Secure Routing Protocol for Ad hoc Networks," In Proc. of the 10th IEEE inter- national Conference on Network Protocols(ICNP). 2002.

[15] Niki Pissihou, Tirthankkar Ghosh and Kia Makki, "Collaborative Trust-Based Secure Routing in multihop Ad Hoc Networks," LNCS 3042, April, 2004.

[16] K. Zhang, "Efficient protocols for signing routing messages," In Proc. of the 1998 Internet Society (ISOC) Symposium on Network and Distributed System Security, San Diego, California, March, 1998.

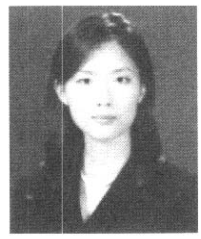
[17] W. Diffie and M. Hellman, "Net Directions in Cryptography," IEEE Trans. on Information Theory, IT-22, pp.644-654, November, 1976.

[18] L. Lamport, "Construction digital signatures from one-way function," Technical Report SRI-CSL-98, SRI International, October, 1979.

[19] R. C. Merkle, "A Digital Signature Based on a Conventional Encryption Function," In Proc. of CRYPTO'87, LNCS 293, pp.369-378, 1987.

[20] R. C. Merkle, "A Certified Digital Signature," In Proc. of CRYPTO'89, LNCS 435, Springer Verlag, pp.218-238, 1990.

[21] Ronald Rivest : The MD5 Message Digest Algorithm, RFC1321, April, 1992, ftp://ftp.rfc-editor.org/in-notes/rfc1321.txt.



편 혜 진

e-mail : hyejin.pyeon@samsung.com

2002년 이화여자대학교 컴퓨터학과(학사)

2004년 이화여자대학교 컴퓨터학과(석사)

2003년~현재 삼성전자(주) 정보통신총괄 무선사업부

관심분야 : 네트워크 보안, 애드혹 네트워크

도 인 실

e-mail : isdoh@ewhain.net

1993년 이화여자대학교 전자계산학과(학사)

1995년 이화여자대학교 전자계산학과(석사)

1995년~1998년 삼성SDS

2002년~현재 이화여자대학교 컴퓨터학과 박사과정

관심분야 : 네트워크 보안, 애드혹 네트워크, 센서 네트워크, 유비쿼터스 컴퓨팅

채 기 준

e-mail : kjchae@ewha.ac.kr

1982년 연세대학교 수학과(학사)

1984년 미국 Syracuse University 컴퓨터학과(석사)

1990년 미국 North Carolina State University 컴퓨터공학과(박사)

1990년~1992년 미국 해군사관학교 컴퓨터학과 조교수

1992년~현재 이화여자대학교 컴퓨터학과 교수

관심분야 : 네트워크 보안, 인터넷/무선통신망/고속통신망 프로토콜 설계 및 성능분석, 센서 네트워크, 홈 네트워크, 유비쿼터스 컴퓨팅