情報保護學會論文誌
第15卷 第6號, 2005. 12

Ad hoc 네트워크에서 GRID-Location Aided Routing 프로토콜을 이용한 분산 CA 구성

임 지 형,[1†] 강 전 일,[1] 고 재 영,[2] 한 광 택,[2] 양 대 헌[1‡]
[1]인하대학교 정보통신대학원, [2]한국전자통신연구원

Distributed Certificate Authority under the GRID-Location Aided Routing Protocol

Jihyung Lim,[1†] Jeonil Kang,[1] JaeYoung Koh,[2] KwangTaek Han,[2] DaeHun Nyang[1‡]
[1]INHA University Graduate School of IT&T,
[2]Electronics and Telecommunications Research Institute

요 약

Ad Hoc 네트워크는 미리 구성된 기반시설 없이 네트워크를 구성할 수 있으며, Ad hoc 네트워크에 참여하는 모바일 노드는 네트워크에 자유롭게 참여 할 수 있다. 이러한 모바일 노드의 자유로운 참여는 새로운 라우팅 경로를 찾기 위하여 잦은 계산을 요구하며, 악의적인 사용자에 의한 잘못된 정보의 유포는 심각한 보안 문제를 발생시킨다. 후자의 보안 문제를 해결하기 위해서는 네트워크에 참여하는 모바일 노드에 대한 인증이 필요하다고 할 수 있다. 인증기관 구성 방법 중 단일 CA 방식을 사용할 경우, CA에 대한 공격은 전체 네트워크의 붕괴를 가져올 수 있으나, 분산 CA 방식을 채택하게 될 경우 여러 노드를 공격해야 완전한 정보를 얻을 수 있기 때문에 단일 CA 방식에 비해 안전하지만 자원적인 측면의 문제가 있다. 일례로 분산 CA를 구성하기 위한 Secret Sharing 방법의 경우 네트워크 크기가 커질수록 전체 노드와 인증서 발급과정을 수행하기 때문에 지연문제가 발생한다. 이 논문에서는 이러한 문제를 해결하기 위한 방법으로 단계별 인증서 발급 방법을 제안하고, 제안된 방법에 대한 시뮬레이션 결과를 보여준다.

ABSTRACT

Ad hoc network is the network which can be considered without a pre-constructed infrastructure, and a mobile node can join the network freely. However, the participation of the mobile nodes to the ad hoc network brings up much burden of re-computation for new routes, because it leads to losing the connection frequently. And, also, it causes serious security problem to be broadcasted wrong information by the malicious user. Therefore, it needs authentication against the mobile nodes. To make that possible, we have two methods: single CA and distributed CA. In the case of CA method, the wireless network can be collapsed owing to expose the CA, but still the distributed CA method is a little more safe than previous one because it needs attacks toward a lot of CAs to collapse the network. We can consider Secret Share scheme as the method that constructs the distributed CA system, but it is weak when the network size is too large. In this paper, we suggest hierarchical structure for the authentication method to solve this problem, and we will show the results of simulation for this suggestion.

Keywords : *Ad hoc network security, Grid-Authentication*

Ⅰ. 서 론

Ad hoc 네트워크는 미리 구성된 기반시설 없이 네트워크를 구성할 수 있는 특징을 가지고 있다. 네

접수일 : 2005년 8월 12일 ; 채택일 : 2005년 11월 30일
† 주저자 : bolter@seclab.inha.ac.kr
‡ 교신저자 : nyang@inha.ac.kr

트워크 내의 모바일 노드는 무선 통신 장비를 장착하고 있으며, 상호 노드 간 통신 서비스를 제공하고 다른 노드의 패킷을 중계해 주는 역할을 한다. 이러한 모바일 노드는 각기 다른 계산 능력과 제한된 배터리 용량을 가지고 있으며, 네트워크에 자유롭게 참여할 수 있는 특징을 가지고 있다. 그러나 네트워크에 자유롭게 참여할 수 있는 특징은 라우팅 경로에 대한 잦은 단절로 말미암아 경로의 재계산을 수행해야 하는 부담을 가져온다. 또한 모바일 노드간의 인증 절차 없이 데이터를 전송함으로써 데이터를 중간에서 가로채거나 데이터의 위·변조, 메시지 재전송, 잘못된 라우팅 경로 유포 등의 보안 문제를 발생시킬 수 있다. 따라서 Ad hoc 네트워크는 이러한 보안 문제에 대한 안전성을 확보하기 위해서 가용성(Availability), 기밀성(Confidentiality), 무결성(Integrity), 인증(Authentication), 부인거부(Non- repudiation) 등의 특성을 갖추어야 한다.[1]

위에서 언급하였던 보안 문제를 해결하기 위해서 지금까지 다양한 연구가 이루어졌으며, 대부분의 연구는 노드 사이에서 안전하게 키를 교환하는 방법과 안전한 경로를 통한 데이터의 전송하는 방법 등에 초점을 맞추고 있다.[2-5] 이러한 방법들 중에서, 분산된 CA로부터 인증서를 획득하는 Secret Sharing 기법[6-8]은 다수의 노드들에게 비밀 키를 분배하여 운영하는 방식이다. 이 방법에서, 노드가 전송하려고 하는 메시지는 다른 노드의 부분 비밀 키(Partial Private Key)로 서명 받아야만 한다. 여기에서 노드는 서명된 메시지를 얻기 위하여 전체 노드들로부터 서명을 받아야하는 것이 아니라, 단지 한계 값(threshold) 이상의 노드들로부터 메시지를 서명 받으면 된다. 이 방법을 분산 CA에 적용하기 위해서는, 노드는 자신이 사용할 공개키를 부분 비밀 키를 가진 노드들로부터 서명 받음으로써 인증서를 획득해야 한다. 하지만 이 방법을 Ad hoc 네트워크에서 직접 사용하게 될 경우 네트워크의 크기에 따라 문제가 발생할 수도 있다. 이것은 네트워크 내의 노드가 인증서를 획득하고 공유하기 위해서 네트워크 내의 모든 노드에게 인증서 관련 정보를 요청하는 것은 네트워크에 많은 부하를 가져오기 때문이다.

본 논문에서 이러한 문제점을 해결하기 위해서 그룹으로 구성된 계층 구조를 사용하는 인증서 발급 방법을 제안한다. 노드는 네트워크에 참여하기 전에, 전체 네트워크에 대한 인증서를 획득하는 것이 아닌 가장 작은 범위의 그룹에 대한 인증서 획득을 수행한다. 다른 노드에게 데이터를 전송할 때, 노드는 자신이 가지고 있는 그룹 정보 내에 종착 노드에 대한 정보를 가지고 있는지의 여부를 확인한다. 만약 자신이 종착 노드에 대한 정보를 가지고 있지 않다면, 노드는 상위 그룹에서 해당 노드에 대한 정보를 찾게 된다. 이것은 전체 네트워크에 대해서 인증서 발급을 수행하는 방법보다 조금 더 효율적이다. 2장에서는 이러한 방법에 대해서 자세히 기술하고, 3장에서는 시뮬레이션을 통하여 이 방법에 대한 효율성을 입증한다.

II. 인증서 발급 프로토콜

1. 요구사항

기본적으로 우리가 제안하는 방법에서 사용하는 라우팅 프로토콜은 GLS(Grid Location Service)[9]이다. 이 프로토콜에서 모든 노드는 "World Grid"라 불리는 네트워크에 포함되며, "ID"라는 유일한 값을 가진다. 또한, 모든 노드는 자신의 ID 이외에 통신을 원하는 상대 노드의 ID를 알고 있다고 가정한다. 노드는 다른 노드와 통신을 하기 위해서 Location Server를 가진다. 임의의 노드는 자신의 ID보다 첫 번째로 큰 ID를 가진 노드를 Location Server로 설정하며, 이것은 순환적인(cyclic) 특성을 갖는다. 송신 노드는 종착 노드를 찾기 위해서 종착 노드의 ID와 가장 가까운 ID를 가진 노드에게 종착 노드 ID의 보유 여부를 확인한다. 질의를 받은 노드가 종착 노드에 대한 정보를 알고 있을 경우 연결 요청을 수행해 주며, 그렇지 않을 경우 앞의 과정을 반복하여 종착 노드를 찾을 때까지 질의를 반복한다.

우리는 위의 방법에 인증을 위한 과정을 추가하였다. 모바일 노드는 자신이 사용할 공개 키/비밀 키가 필요하며, 이들 키 쌍에 대해 인증을 받기 위해 네트워크에 참여하기 전에 이웃 노드로부터 자신이 소속될 가장 작은 그룹에 대한 정보를 얻어온다. 만약 요청 노드가 그룹에 대한 정보에서 딜러(Dealer) 노드를 알아내면, 요청 노드는 딜러 노드로부터 그룹 내에서 자신이 사용할 공유 값(Share Value)과 그룹 목록(Group List)을 획득할 수 있다. 요청 노드는 메시지를 보내기 전에 그룹 내의 노드들로부터 인증서 역할을 하는 서명된 메시지를 획득해

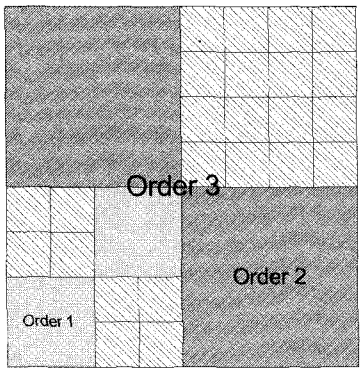


그림 1. 그룹 계층도 : 전체 네트워크를 논리적으로 분할하여 Order라는 단위로 구분

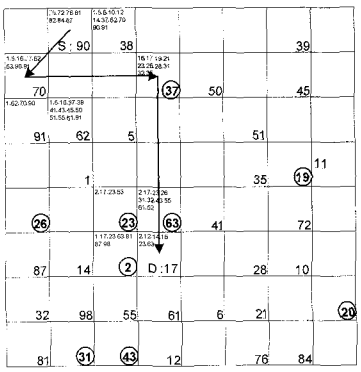


그림 2. GLS 구조에서 Location Server의 설정과 노드 탐색(ID 17기준)

야 한다. 만약 종착 노드가 그룹 내에 포함되어 있지 않다면, 송신 노드는 송신 노드와 종착 노드가 속해있는 그룹 목록을 획득해야 한다. 이 방법에서 최악의 경우, 그룹의 최대 크기는 World Grid가 될 것이다.

2. 패킷 종류와 인증 테이블

인증서 발급을 수행하기 위해서 기존 GLS에서 사용하는 패킷 이외에 추가적인 패킷이 필요하다. 이렇게 새롭게 추가된 패킷은 그룹 정보를 획득하는 과정과 인증서를 획득하는 과정을 위한 것으로 다시 구분된다.

2.1 그룹 정보 획득

- SHARE_REQUEST : 그룹에 대한 정보를 획득하기 위해 이웃 노드들에게 보내는 메시지이다.

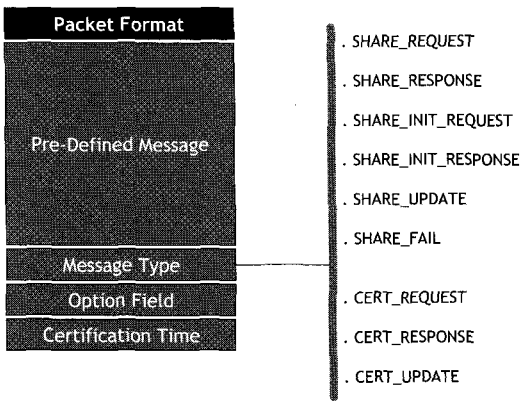


그림 3. 패킷 종류

- SHARE_RESPONSE : SHARE_REQUEST 메시지에 대한 응답 메시지로 딜러 노드의 ID를 포함한다.
- SHARE_INIT_REQUEST : 새로운 그룹을 생성하기 위해 이웃 노드에게 보내는 메시지이다.
- SHARE_INIT_RESPONSE : SHARE_INIT_REQUEST의 응답 메시지이다.
- SHARE_UPDATE : 그룹에 대한 갱신 사항을 그룹 내의 구성원에게 보내는 메시지이다.
- SHARE_FAIL : SHARE_REQUEST 메시지에 대해서 자신이 그룹에 대한 정보를 제공할 수 없을 때, 사용되는 메시지이다.

노드는 네트워크에 참여하기 전에 자신이 소속될 그룹에 대한 정보를 얻어야 한다. 그룹에 대한 정보를 획득하기 위해서, 노드는 이웃 노드들에게 SHARE_REQUEST 메시지를 전송한다. 이를 받은 이웃 노드는 자신이 가지고 있는 인증 테이블(Certification Table)을 참조하여, 요청 그룹에 대한 정보를 가지고 있다면 SHARE_RESPONSE 메시지로 응답하고, 그렇지 않다면 SHARE_FAIL 메시지로 응답한다. 요청 노드가 SHARE_RESPONSE 메시지를 받았다면, 메시지의 정보를 이용하여 그룹의 딜러 노드에게 자신을 등록해야 한다. 그런 뒤, 요청 노드는 딜러 노드로부터 SHARE_UPDATE 메시지를 받음으로써, 그룹에서 사용할 공유 값과 그룹 목록을 획득할 수 있다. 만약 요청 노드가 SHARE_FAIL 메시지를 받는다면, 이웃 노드들에게 SHARE_INIT_REQUEST 메시지를 보내어 그룹을 생성하게 된다. 만약 노드가 SHARE_INIT_REQUEST 메시

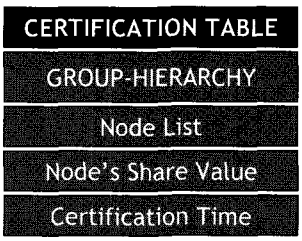


그림 4. 인증 테이블

지를 받는다면, 메시지를 이웃 노드들에게 재전송해야 하며, 최초로 SHARE_INIT_REQUEST 메시지를 보낸 노드에게는 SHARE_INIT_RESPONSE를 보내야 한다.

2.2 인증서 획득

- CERT_REQUEST : 자신이 속한 그룹의 구성원에게 자신이 생성한 인증서에 대한 서명을 요청하기 위해서 보내는 메시지이다.
- CERT_RESPONSE : CERT_REQUEST에 대한 응답 메시지로써 부분 비밀 키로 서명된 메시지를 담고 있다.
- CERT_UPDATE : 만료된 인증서에 대한 갱신을 알리기 위해서 보내는 메시지이다.

만약 노드가 그룹에 대한 정보를 획득하면, 그룹에서 사용할 수 있는 인증서를 획득해야 한다. 노드는 CERT_REQUEST 메시지에 자신이 생성한 공개키를 담아서 그룹 내의 다른 노드들에게 전송한다. 이 메시지를 그룹 내의 노드들이 받게 되면, 자신이 가지고 있는 부분 비밀 키로 메시지를 서명한 뒤, CERT_RESPONSE 메시지에 담아 요청 노드에게 돌려준다. 만약 인증서가 만료 되었다면, 노드는 CERT_UPDATE 메시지를 그룹 내의 노드에게 보냄으로써 인증서를 갱신 받아야 한다.

2.3 인증 테이블(Certification Table)

모든 노드는 인증서를 유지하기 위해서 그림 4와 같은 인증 테이블이 필요하다.

"GROUP-HIERARCHY"는 x, y축 좌표에 기반한 그룹의 ID 이고, 그룹에 대해서 유일한 값을 가진다. "Node List"는 "GROUP-HIERARCHY"로 구분된 그룹에 속한 노드의 ID값의 집합이다. 각 노드는 이 값을 통해서 그룹 내의 노드들에게 인증서를 요청하며, 그룹 내의 다른 노드들과 통신을 수행할 수 있다. "Node's Share Value"는 그룹의 딜러 노드로부터 획득하는 값으로, 노드는 이 값을 가지고 그룹 내의 다른 노드가 인증서에 대한 서명 요청을 해 올 때 수행하게 된다. "Certification Time"은 인증서를 획득한 시간을 기록한 값이다.

3. 프로토콜

네트워크에 참여하는 노드는 네트워크에 참여하기 전에 자신이 사용할 공개 키/비밀 키 쌍을 생성한다. 이후, 이 프로토콜은 크게 두 부분으로 작동한다. 첫 번째 부분은 그룹으로부터 공유 값을 획득하는 과정이고, 두 번째 부분은 이 정보로부터 그룹 내의 다른 노드들에게 인증서를 서명받는 과정이다. 노드는 다른 노드와 통신을 하기 위해서 그룹 내에서 사용할 인증서를 획득해야 하며, 통신하고자 하는 노드 역시 동일 그룹에 속해 있어야 한다. 이러한 작업을 수행하기 위해서, 각 노드는 네트워크에 참여할 때, 먼저 자신의 Order 1 크기의 Grid에 대해서 "그룹 정보 획득 과정"과 "인증서 획득 과정"

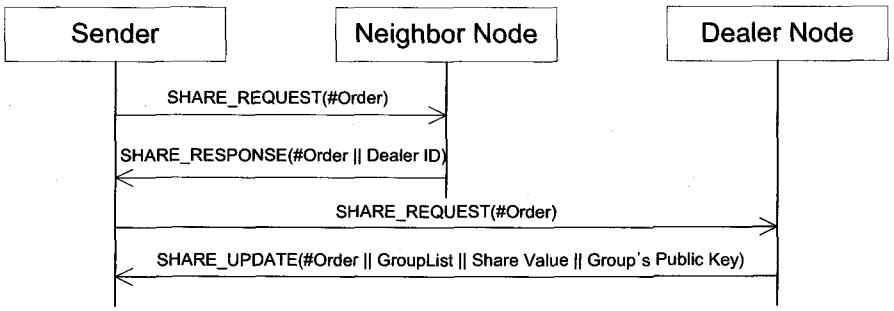


그림 5. 이웃 노드가 딜러의 정보를 제공할 수 있는 경우

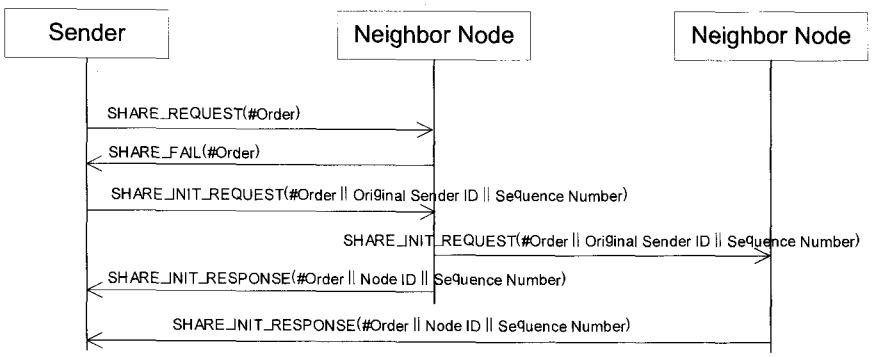


그림 6. 노드가 그룹에 대한 정보를 제공할 수 없는 경우

을 수행한다. 만약 통신하려는 노드가 자신이 가진 그룹에 포함되어 있지 않다면 송신 노드는 종착 노드가 포함된 상위 그룹에 대한 정보를 획득한 뒤, 통신을 수행해야 한다. 가장 큰 그룹은 World Grid가 된다.

3.1 그룹 정보 획득 과정

노드는 인증서를 획득하기 전에 그룹에 대한 정보를 획득해야 한다. 이 정보는 인증 테이블 내에 기록된다. 만약 이웃 노드가 요청에 대한 응답을 수행한다면, 그림 5 또는 그림 6의 역할을 수행한다.

Step 1. 노드가 그룹에 대한 정보를 가지고 있지 않다면 노드는, 자신이 속한 그룹에 대한 정보를 얻어내기 위해서 이웃 노드들에게 SHARE_REQUEST(#Order) 메시지를 브로드 캐스트 한다.

Step 2. SHARE_REQUEST 메시지를 받은 노드는 자신의 x, y 좌표를 계산하고, 자신의 인증 테이블을 검사하여 요청 메시지에 대한 응답을 수행할 수 있다. 만약 자신이 요청에 대해서 응답할 수 있는 경우라면, SHARE_RESPONSE (#Order || dealer's ID)를 돌려준다. 그렇지 않다면 SHARE_FAIL(#Order) 메시지를 전송해준다.

Step 3. 이 후, 각 노드는 다음의 일을 수행해야 한다.

a. 만약 노드가 SHARE_FAIL(#Order) 메시지를 받는다면, 인증 테이블 내의 GROUP-HIERARCHY 항목에 자신의 #Order와 x, y축에 기반한 값을 생성하여 입력하고, 자신의 노드 ID를 Node List Field에 추가한다. 그 뒤, SHARE_INIT_REQUEST (#Order || original sender's ID || Sequence No.) 메시지를 이웃 노드들에게 보내고, 동일한 메시지에 대한 응답을 막기 위해서 버퍼 테이블에 전송 내역을 기록한다. 그렇지 않다면 노드는 메시지를 버린다.

b. 만약 노드가 SHARE_INIT_REQUEST 메시지를 받는다면, 해당 메시지를 이전에 처리한 적이 있는지를 버퍼 테이블에서 검사한다. 만약 자신이 처리한 적이 없었던 메시지라면, 자신이 요청 메시지의 #Order에 소속되어 있는지를 검사한다. 만약 자신이 동일한 #Order 내에 속해 있는 노드라면 노드는 자신의 인증 테이블내의 해당 GROUP-HIERARCHY의 Node List에 original sender's ID를 추가한다. 그런 뒤 노드는 이웃 노드들에게 자신이 받은 SHARE_INIT_REQUEST 메시지를 전송하고, 최초 그룹을 요청한 노드(Original Sender)에게 SHARE_INIT_RESPONSE (#Order || node's ID || Sequence No.)를 전송한다.

c. 만약 노드가 SHARE_INIT_RESPONSE 메시지를 받는다면, 메시지내의 #Order, node's ID, and Sequence No. 값의 적합성 여부를 검사해야 한다. 만약 적합한 메시지라면, 노드는 Certification Table내의 해당 GROUP-HIERARCHY의 Node List에 node's ID를 추가한다. 만약 노드가 딜러노드라면, 노드는 SHARE_INIT_RESPONSE 메시지를 보낸 노드에게 SHARE_

UPDATE(#Order || Node List || Share Value) 메시지를 전송한다.

d. 만약 노드가 SHARE_RESPONSE 메시지를 받는다면, 노드는 메시지로부터 딜러 노드의 ID를 알 수 있다. 그런 뒤 노드는 딜러노드에게 SHARE_REQUEST 메시지를 보내게 된다.

그룹 내에 새로운 노드가 참여하게 될 때, 딜러노드는 SHARE_UPDATE(#Order || Node List || Share Value || Group's Public Key) 메시지를 그룹내의 노드들에게 전송함으로써 Node List의 갱신을 알리게 된다.

3.2 인증서 획득 과정

그룹 정보 획득 과정 절차 이후에, 노드는 다음의 인증서 획득 과정을 수행해야 한다.

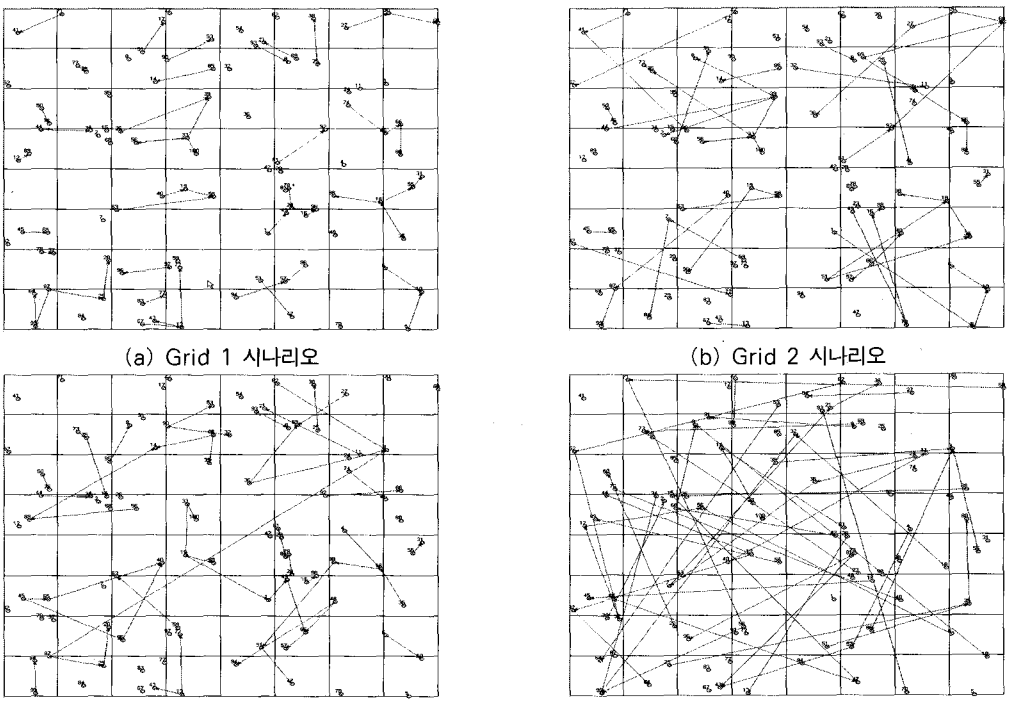
Step 1. 노드의 공개키가 담긴 메시지를 서명받기 위해서, 노드는 CERT_REQUEST (#Order || message)를 인증 테이블내의 요청 그룹에 대한 Node List Field에 포함된 노드들에게 전송한다.

Step 2. 만약 노드가 CERT_REQUEST 메시지를 받는다면, 노드는 요청 그룹과 관련된 부분 비밀 키를 사용해서 CERT_REQUEST 메시지의 message 부분을 서명하고, CERT_RESPONSE(#Order || $Group_{sk}$[message])를 돌려준다.

노드는 자신의 인증서가 만기되었을 때 해당 그룹내의 노드들에게 CERT_UPDATE(#Order || message) 메시지를 보냄으로써 인증서를 갱신할 수가 있다.

III. 시뮬레이션과 결과

시뮬레이션을 위해서 NS-2라는 이산 네트워크 시뮬레이션을 사용하였고, 2가지 경우의 시나리오를 사용하였다.[10-11] 하나는 노드의 이동이 없는 네트워크(Non-Mobility) 모델이고, 다른 하나는 노드의 이동이 있는 네트워크(Mobility) 모델이다. 또한 전체 네트워크("Entire")가 인증에 참여하는 방법과 네트워크를 그룹으로 구분("Grid")하여 인증에 참여하는 방법을 비교했다. "Grid 1" 방법은



(a) Grid 1 시나리오 (b) Grid 2 시나리오

(c) Grid 3 시나리오 (d) Grid Random 시나리오

그림 7. 통신에 사용된 Grid 시나리오

order 1 내의 노드만이 통신하는 형태이다. "Grid 2"는 order 1(50%)과 order 2(50%)의 노드들이 통신하는 형태이다. "Grid 3"은 order 1(30%)과 order 2(30%) 그리고 order 3(40%)의 노드들이 통신하는 형태이다. "Grid Random"은 모든 order 내의 노드들이 랜덤하게 통신하는 형태이다. 그림 7은 통신에 사용된 시나리오에 대한 그림이다. 이 시뮬레이션에는 100개의 노드를 시뮬레이션 시간 400초 동안 진행하여 결과를 분석하였다.

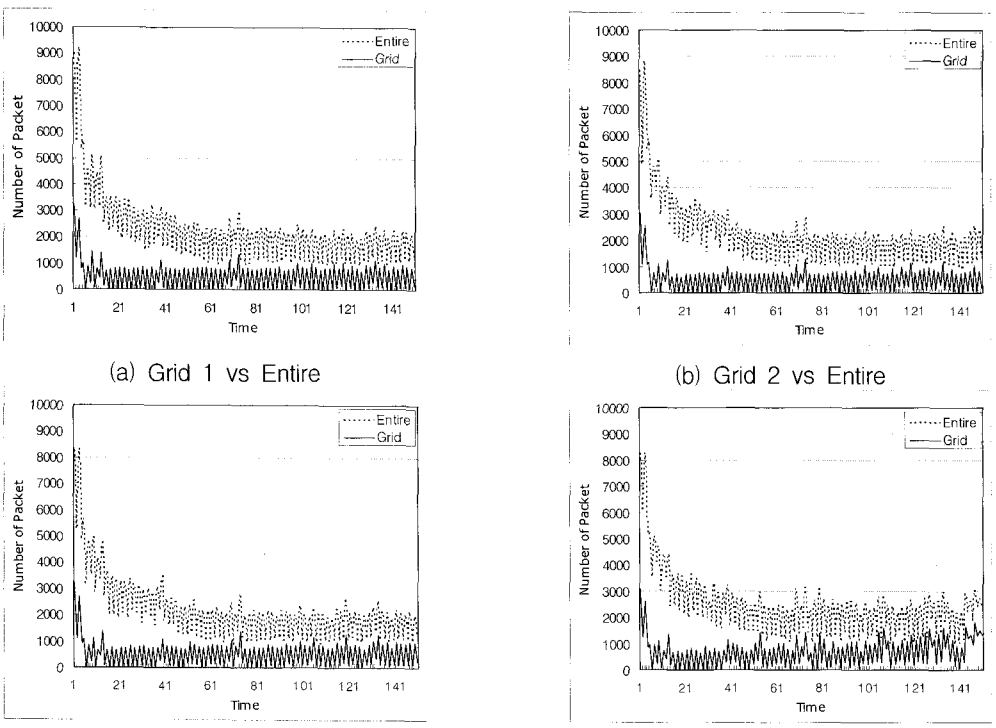
1. 노드의 이동이 없는 네트워크 모델

1.1 전체 컨트롤 패킷량

시간에 대한 패킷량은 그림 8과 같다. Entire 방법의 경우, 초기 발생되는 데이터량이 많다는 것을 볼 수 있는데 이것은 초기 노드가 데이터를 전송하기 전 자신의 이웃 노드들과 주고받는 인증서 관련 패킷량이 많기 때문이다. 반면 Grid 방식의 경우 노드의 네트워크 진입 시 노드가 Order 1내의 노드와 인증서 발급을 수행하므로 사용되는 인증서와 관련된 패킷량이 적다. 또한, 데이터를 전송하기까지의 인증서 발급에 드는 시간이 Entire 방식에 비하여 Grid 방식이 짧다는 것을 알 수 있다. 이것은 Entire 방식이 노드는 데이터를 전송하기 전 네트워크 내의 전체 노드들과 인증서 발급과정을 수행해야 하는 반면, Grid 방식의 경우 통신하고자 하는 노드가 속한 최소 크기 그룹과의 통신만 필요하기 때문이다. Grid 방식에서 Grid 1, 2 그리고 3 방식의 패킷량의 차이가 근소하다는 것을 알 수 있다. 초기 네트워크에서 노드들은 인증서에 대한 정보를 가지고 있지 않기 때문에, 데이터를 전송하기 전에 인증서에 대한 정보를 얻기 위해서 인증서 관련 메시지를 전송하게 된다. 이 때, 인증서 발급과 관련된 패킷의 대부분이 발생되며, 인증서를 얻은 이 후 좀 더 큰 그룹에 대한 인증서를 얻기 위해 보내는 메시지는 몇몇 통신에 참여하는 노드만이 수행하기 때문에 Entire 방식보다 패킷의 양이 많지 않다는 것을 알 수 있다.

1.2 인증서 발급 관련 패킷량

시간에 따른 인증서 관련 패킷량은 그림 9와 같다. 인증서에 관련된 패킷량의 대부분은 SHARE_



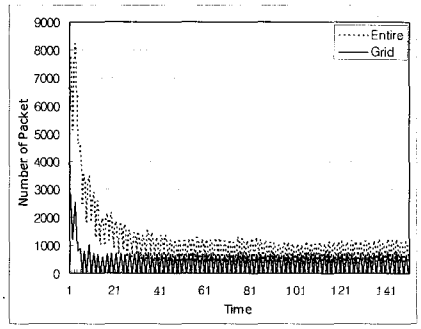
(a) Grid 1 vs Entire

(b) Grid 2 vs Entire

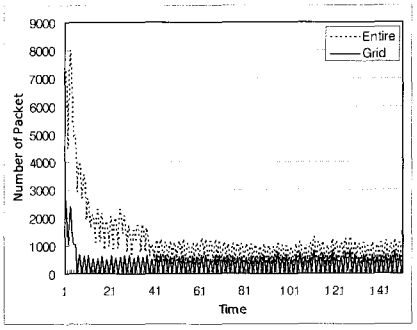
(c) Grid 3 vs Entire

(d) Grid Random vs Entire

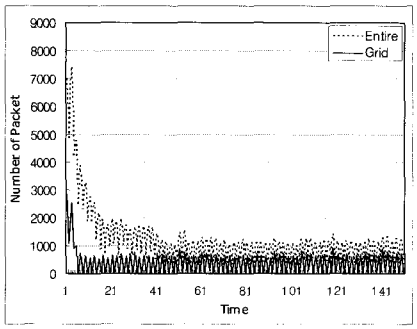
그림 8. 전체 컨트롤 패킷량 비교



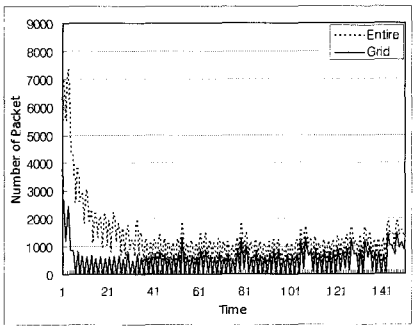
(a) Grid 1 vs Entire



(b) Grid 2 vs Entire

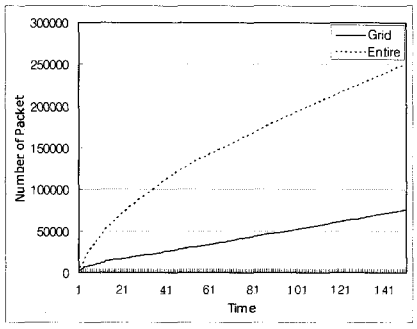


(c) Grid 3 vs Entire

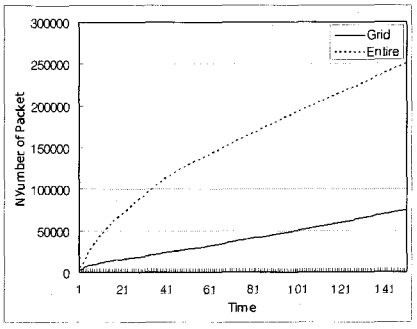


(d) Grid Random vs Entire

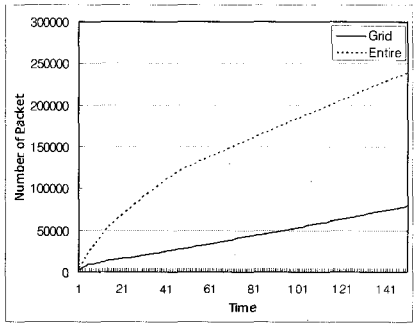
그림 9. 인증서 발급 관련 패킷량 비교



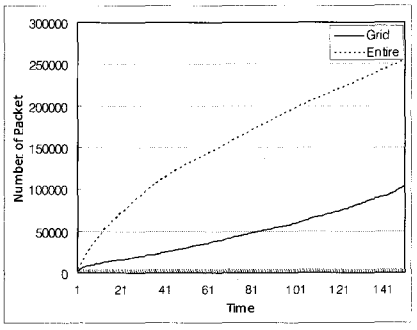
(a) Grid 1 vs Entire



(b) Grid 2 vs Entire



(c) Grid 3 vs Entire



(d) Grid Random vs Entire

그림 10. 누적 컨트롤 패킷량 비교

REQUEST, SHARE_RESPONSE, SHARE_INIT_REQUEST, SHARE_INIT_RESPONSE 이다. 인증서 요청에 대한 패킷을 전송한 이후부터 노드들은 이들 인증서를 얻기 위한 패킷을 전송한다. Grid 방식의 경우 약 5~6초 사이의 인증서를 얻기 위한 시간을 사용하고 있으나, Entire 방식의 경우 30~40초의 요청 시간을 요구한다.

1.3 누적 컨트롤 패킷량

그림 10은 시간에 따른 누적된 전체 패킷량을 보여준다. 시간이 진행 될수록 Grid 방식과 Entire 방식의 사용되는 패킷량이 큰 차이를 보이는 것을 알 수 있다.

2. 이동성

이동성이 적용된 시나리오와 그렇지 않은 시나리오의 패킷 차이는 예상외로 그렇게 많이 차이가 나지 않는다. 이것은 다음의 이유 때문이다. 초기 네트워크에서, 노드는 그룹에 대한 정보가 없기 때문에 해당 그룹에 대한 정보를 생성하고, 이 그룹으로부터 인증서를 획득하기 위해 많은 패킷을 보내게 된다. 이러한 그룹이 생성되어진 뒤, 노드의 이동에 의한 이동된 위치의 그룹 인증서 요청과 상위 그룹에 대한 인증서 요청은 전체 패킷량에 그렇게 큰 영향을 미치지 않는 것으로 보인다.

3. 인증서 발급 지연

인증서 발급 지연은 데이터 통신 전에 인증서를 획득하는 시간을 측정한 값이다. 인증서의 획득은 노드가 통신하고자 하는 그룹 목록내의 노드들의 부분 비밀 키로 서명된 메시지를 50% 이상 받은 경우를 데이터 통신이 가능한 시점으로 측정하였다. Table 1.에서 Entire 방식이 Grid 방식보다 훨씬 더 많은 지연을 가지는 것을 볼 수 있다. 이것은 두 가지 이유에 기인한다. 첫 번째로, 그룹 정보를 획득하는 시간이 인증서를 획득하는 시간보다 빠르기 때문이다. 두 번째로, 노드가 인증서를 획득하는 동안 딜러노드가 변경되었다면, 노드는 변경된 딜러노드로부터 새로운 그룹 정보를 획득한 뒤, 인증서를 다시 획득해야 하기 때문이다.

Ⅳ. 결 론

이 논문에서는 네트워크에 인증서 발급단계에 계층구조를 두어 분산 CA를 효율적으로 구성하는 방법에 대해서 논의하고 있다.

기존 Secret Sharing 방법은 노드들이 부분 비밀 키를 통하여 분산 CA를 구성하는 방법을 보여주고 있다. 네트워크의 크기가 큰 경우에 이러한 방법을 사용하면 노드는 인증서를 발급 받기 위해서 많은 지연시간을 가지게 된다. 이것은 노드가 네트워크 참여함과 동시에 발생하게 되고, 네트워크에서 통신을 하지 않는 경우라도 발생되어지므로 네트워크 자원을 낭비하는 결과를 가져온다. 이 논문에서 제안한 단계별 인증서 발급을 통한 분산 CA의 구성은 위의 문제점을 감소하여 인증서를 획득하는 방법을 보여주고 있다.

이 방법을 사용하게 될 경우, 초기 네트워크 구성 시 전체 노드들이 참여하여 인증서를 구성하는 방법보다 좀 더 빠르게 노드가 구성되는 것을 볼 수 있었다. 또한 이 후, 원하는 네트워크에 대한 구성 시 그렇게 많지 않은 패킷량을 사용해서 원하는 크기의 그룹을 구성하는 것을 볼 수 있었다. 이러한 인증서 구성 방법은 여타 다른 무선 네트워크에서 PKI(Public Key Infrastructure)를 구성하는데도 응용할 수 있을 것이다.

참 고 문 헌

[1] C-K Toh, "Ad Hoc Mobile Wireless Networks Protocols and Systems", Prentice Hall, Jan. 2004.

표 1. Entire와 Grid 방식에 대한 평균 인증서 발급 지연 비교 (unit : sec)

	Grid 1	Grid 2		Grid 3			Grid Random		
	Order 1	Order 1	Order 2	Order 1	Order 2	Order 3	Order 1	Order 2	Order 3
Average for Grid	2.79	2.62	29.89	2.59	31.91	41.62	2.63	36.04	39.08
Average for Entire	113.29	119.02		100.31			94.86		

[2] Stephen Carter and Alec Yasinsac, "Secure Position Aided Ad hoc Routing Protocol", Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02), Nov 3-4, 2002.

[3] S. Yi, P. Naldurg, and R. Kravets, "Security-Aware Ad hoc Routing for Wireless Networks", The Second ACM Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc'01), 2001.

[4] Jean-Pierre Hubaux, Levente Buttyan, Srdjan Capkun, "The Quest for Security in Mobile Ad Hoc Networks", Proceedings of the 2001 ACM International Symposium on Mobile ad hoc networking \& computing, 2001.

[5] Baruch Awerbuch, David Holmer, Cristina Nita-Rotaru and Herbert Rubens, "An On-Demand Secure Routing Protocol Resilent to Byzantine Failures", In ACM Workshop on Wireless Security (WiSe), Atlanta, Georgia, September 28 2002.

[6] Shamir A., "How to Share a Secret", Communication of the ACM, pp. 612-613, Nov. 1979.

[7] Jiejun Kong, "Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks", IEEE 9th International Conference on Network Protocols (ICNP'01), pp.251-260, 2001.

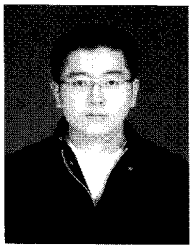
[8] T.P. Pedersen, "A threshold cryptosystem without a trusted party", Advances in Cryptology, Proceedings of the CRYPTO'91, pp. 522-526, 1991.

[9] Jinyang Li, "A Scalable Location Service for Geographic Ad Hoc Routing", 6th ACM International Conference on Mobile computing and Networking (MobiCom'00), pp.120-130, Aug. 2000.

[10] The Network Simulator(ns-2), http://www.isi.edu/nsnam/ns/

[11] The Grid Ad Hoc Networking Project, http://pdos.csail.mit.edu/grid/sim/

〈著者紹介〉



임 지 형 (Jihyung Lim) 학생회원
2004년 2월: 인하대학교 컴퓨터 공학과 졸업
2004년 3월~현재: 인하대학교 정보통신대학원 석사과정
〈관심분야〉 Ad Hoc 네트워크 보안, 암호이론, 무선센서네트워크 보안

강 전 일 (Jeonil Kang) 학생회원
2003년 2월: 인하대학교 컴퓨터 공학과 졸업
2004년 3월~현재: 인하대학교 정보통신대학원 석사과정
〈관심분야〉 RFID 보안, 인증프로토콜, 무선센서네트워크 보안

고 재 영 (JaeYoung Koh) 정회원
1984년 2월: 전북대학교 전자공학과 졸업
1992년 8월: 전북대학교 전자공학과 석사
1998년 8월: 전북대학교 전자공학과 박사
1984년 3월~2000년 1월: 국방과학연구소 선임연구원
2000년 2월~현재: 한국전자통신연구원 책임연구원

한 광 택 (KwangTaek Han) 정회원
1998년 2월: 고려대학교 전산학과 졸업
2001년 8월: 고려대학교 전산학과 석사
2004년 9월~현재: 고려대학교 정보보호대학원 박사과정
2000년 4월~현재: 한국전자통신연구원 선임연구원

양 대 헌 (DaeHun Nyang) 정회원
1994년 2월: 한국과학기술원 과학기술 대학 전기 및 전자 공학과 졸업
1996년 2월: 연세대학교 컴퓨터 과학과 석사
2000년 8월: 연세대학교 컴퓨터 과학과 박사
2000년 9월~2003년 2월: 한국전자통신연구원 정보보호연구본부 선임연구원
2003년 2월~현재: 인하대학교 정보통신대학원 조교수
〈관심분야〉 암호이론, 암호프로토콜, 인증프로토콜, 무선 인터넷 보안