Ad hoc 네트워크에서 공개키 기반구조를 이용한 신뢰적인 인증 메커니즘

봉진숙° 윤미연 신용태

숭실대학교 컴퓨터학과

bearbong@cherry.ssu.ac.kr°, myyoon@cherry.ssy.ac.kr, shin@computing.ssu.ac.kr

A Secure Authentication Mechanism using PKI in Ad Hoc Networks

Jinsook Bong°, Miyoun Yoon, Yongtae Shin

School of Computing, Soongsil University

요 약

이동 ad-hoc 네트워크(mobile ad-hoc network: MANET)는 유선 기반 망 혹은 액세스 포인트 없이 이동 단말기들로 구성된 망이다. 기존의 기반망을 사용하지 않으므로 유선 기반망이 구축되어 있지 않은 곳이나 유선 기반망이 파괴된 지역에 사용한다. 그러나 MANET에서는 노출된 매체와 동적인 토폴로지, 중앙의 감시와 관리 결여, 자원의 제약성 등과 같은 이유로 인해 유선망보다 더욱 많은 보안 문제가 발생한다. 본 논문에서는 이동 ad hoc 네트워크에서 노드의 신분 보장을 제공하기 위하여 기밀성에 중점을 둔 공개키 기반 구조에서의 노드 간 인증 기법을 제안한다.

1. 서 론

이동 ad-hoc 네트워크(mobile ad-hoc network: MANET)[1][2]는 유선 기반 망 혹은 액세스 포인트 없이 이동 단말기들로 구성된 망이다. 기존의 기반망을 사용하지 않으므로 통신에 참가하는 모든 노드들은 호스트 역할과 라우터 역할을 겸한다. 이동 ad-hoc 네트워크는 이러한 특징들 때문에 유선 기반망이 구축되어 있지 않은 곳, 지진, 홍수, 재난으로 유선 기반망이 파괴된 지역에 사용한다.

그러나 MANET에서는 노출된 매체와 동적인 토폴로지, 중앙의 감시와 관리 결여, 자원의 제약성 등과 같은 이유로 인해 수동적 공격인 도청으로부터 해서 능동적 공격인 DoS 공격까지 다양한 공격 유형들이 존재한다. 또한 이동 ad hoc 네트워크에서는 노드의 신분이 서로에게 불확실한 경우가 많으며 멀티 홉 방식에 의해 라우팅을 할 경우 중간 노드에 의해 발생될 수 있는 데이터 보안 문제가 존재한다. [3][4][5]. 이러한 보안문제로부터 Ad hoc 네트워크를 안전하게 하기 위해서는 가용성, 기밀성, 무결성, 인증, 부인봉쇄 등과 같은 보안 요구사항들을 만족시켜야한다.

본 논문에서는 이동 ad hoc 네트워크에서 노드의 신분 보장을 제공하기 위하여 기밀성에 중점을 둔 공개키 기반 구조에서의 노드 간 인증 기법을 제안한다.

2. 관련연구

2.1 비밀키 공유기법(Secret Sharing)

비밀키 공유기법은 임의의 비밀키를 여러 사용자들이 나누어 갖도록 함으로써 단일 사용자가 자신의 키만으로는 원래의 비밀키를 복원할 수 없도록 하는 기법이다.

2.1.1 Threshold Cryptography[6]

키 관리 서비스에서 비밀키는 n개의 서버에 나누어서 관리한다. 서비스는 공개키/비밀키 쌍인 K/k를 갖는다. 공개키 K는 네트워크의 모든 노드들에게 알리고, 비밀키 k는 n개의 서버 S1, S2, … Sn에 나누어준다. 각각의 서버 Si는 자신의 공개키/비밀키 쌍인 Ki/ki을 갖는다.

(n, t+1)의 threshold cryptography 스킴은 비밀키를 n개 부분으로 나누어 각각의 서버에게 할당하고 그 중 t+1개의 서버에서 올바른 서명이 오면 그 부분 서명들을 조합하여 비밀키를 생성한다. 그러나 공유된 정보가 조합된다 해도 비밀키는 알려지지 않는다. 일반적으로 서명이 정확하기 위해서는 서비스의 공개키의 입증이 필요하다. 이 기법은 t+1개 이상의 서버들이 동시에 타협되어야만 잘못된 서명이 생성될 수 있으므로 잘못된 서명이 생성될 확률은 낮다.

2.1.2 검증 가능한 Secret Sharing[7]

만일 임의의 부분키 소유자가 악의적인 생각으로 다른 부분키 소유자들이 비밀키를 복원하지 못하도록 엉뚱한 값을 전송한다고 가정 할 때, 원래의 비밀키를 복원할 수 있다고 기대했던 사용자는 전혀 다른 값을 추출하게 되므로 이런 일을 방지하기 위해 임의의 부분키 소유자는 전송된 부분키가 유효한 값인지 검증할 필요가 있다.

1) 임의의 dealer는 자신의 부분키들을 전송하기에 앞서 임의의 수 g를 선택한다. 이 g에 $f(x)$의 공계수(a_{k-1}, a_{k-2}, … a1, a0)들을 멱승하여 g^{ak-1}, g^{ak-2}, … ,g^{a1},g^{a0} 를 구한다. 이 dealer는 이 값들을 공표한다.

2) 각 노드는 자신이 임의의 부분키들을 수신할 때, 이 부분키 값의 유효성을 다음과 같이 계산해서 검증한다.

$(g^{ak-1})^{idik-1} \cdot (g^{ak-2})^{idik-2} , \cdots \cdot (g^{a1})^{id1} \cdot g^{a0}$

$$= g^{ak-1idik-1 + ak-2idik-2 + \cdots + a1idi + a0} = g^{Si}$$

3. 공개키 기반구조를 사용한 노드간 인증기법

ad hoc 네트워크와 유선망이 혼재해 있는 망에서 공개키 기반구조를 사용한 노드간 인증기법을 제안하고자 할 때 다음과 같은 사항을 고려한다.

3.1 가정

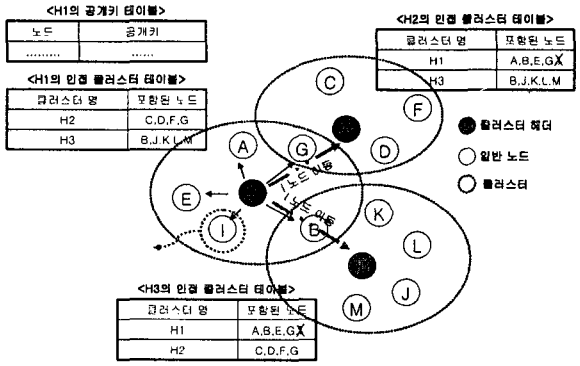
- ad hoc 네트워크의 노드는 유선망에 접속하여 CA를 찾을 수 있다.
- ad hoc 노드들은 글로벌한 IP주소를 갖는다.
- 초기에 통신을 위한 노드들의 집합을 가정한다.

3.2 초기화

모든 노드가 signal을 통해 이웃노드들을 탐색하여 홉수가 1인 노드의 범위를 클러스터 범위로 지정한다. 클러스터가 구성되면 클러스터의 헤더를 선출하게 되는데 클러스터 안에 있는 모든 노드는 자신의 존재여부와 함께 메모리용량에 대한 정보를 브로드 캐스팅 한다. 보통의 클러스터링 기법에서는 클러스터 구성의 효율성을 위하여 휴리스틱 알고리즘[8][9]을 적용하는데 여기에서는 메모리 용량을 고려하여 가장 큰 저장 용량을 가진 노드를 이 클러스터의 헤더로 결정한다.

3.3 클러스터 헤더의 인증

클러스터에서 헤더가 선출되면 클러스터 헤더는 유선망에 접속하여 CA에게 자신을 인증해달라고 요청한다. 이때 인증과정은 유선망에서의 인증과정과 동일하다. 또한 CA는 자신에게 인증을 요청한 클러스터 헤더들에게 Threshold Cryptography[6]을 사용하여 서비스를 위한 비밀키를 각 n개의 헤더(H_1, H_2 ⋯ H_n)들에게 분산 시키고 공개키는 공개한다. 통신하고자 하는 노드는 임의 값 하나를 설정하여 그 임의 값을 서비스 공개키로 암호화하여 자신의 클러스터 헤더에게 보내게 되고 클러스터 헤더는 비밀키 조합을 통하여 그 임의 값을 해독하여 통신을 원하는 노드에게 보내게 되고 통신을 원하는 노드는 그 값을 자신이 보낸 값과 비교하여 보고 일치하면 클러스터 헤더가 인증받았음을 확인하게 된다.



[그림1] 노드 이동 시 인접 클러스터링 테이블의 변화

다른 노드들은 자신이 포함되어 있는 클러스터의 헤더가 CA로부터 인증 받았다는 사실이 확인되면 자신의 공개키를 클러스터 헤더에게 보내고 클러스터 헤더는 클러스터 범위 내에 있는 노드들의 공개키 저장 테이블과 이웃 클러스터 헤더의 클러스터링 테이블을 유지한다. [그림1]에서와 같이 클러스터링 테이블은 이웃노드 탐색을 통하여 주기적으로 갱신된다.

<메시지 정의>

A_REQ	Authentication Request message. 유선 CA에게 보내는 인증 요청 메시지
A_RESP	Authentication Response message 유선 CA가 보내는 인증 응답 메시지
S_PRIV[i] key	Service Private[i] Key 유선 CA가 보내는 서비스 비밀키 조각 중 i번째 조각
HA_REQ	Header Authentication Request message 클러스터 내부에 있는 노드가 자신의 클러스터 헤더가 인증 받은 노드인지를 확인하기 위한 메시지
IA_REQ	Identity Authentication Request message 자신이 인증 받은 클러스터 헤더라는 것을 증명하기 위해 부분 비밀키를 요청하는 메시지
IA_RESP	Identity Authentication Response message 부분 비밀키를 포함하고 있는 클러스터 헤더 인증 응답 메시지
ACK	Acknowledgement message 클러스터가 인증 받은 클러스터라는 것을 확인했다는 메시지
CT_REQ	Clustering Table Request message 이웃의 클러스터링 테이블 요청 메시지
CT_RESP	Clustering Table Response message CT_REQ의 응답으로 자신의 클러스터링 테이블을 보냄

Algorithm1. 클러스터 헤더의 인증(클러스터 헤더)

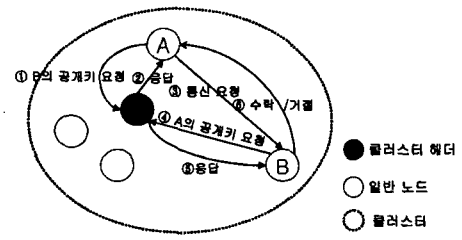
```
Send   A_req message to wired CA
Wait   A_resp message from wired CA AND S_priv[i] key

if (HA_req message from other nodes)
{
  Send    IA_req message to other cluster headers
  Wait    IA_resp message from more than
          t+1 cluster headers
  for (i=1; i<=n; i++)
    Combine    S_priv[i] key
  Process    deciphering
  Send    deciphered random value
  Wait    Ack message AND public key
  Store    node_ID AND public key
}

Send   T_req message to other cluster headers
Wait   T_resp message from other cluster headers
```

3.4 노드간 통신시의 인증기법

3.4.1. 한 클러스터 내에서의 통신

[그림 2]에서와 같이 노드 A가 같은 클러스터 안에 있는 노드 B와 통신을 하고자 한다면 먼저 A는 자신이 속해있는 클러스터의 헤더에게 B의 공개키를 요청한다. 클러스터 헤더는 A의 공개키가 자신에게 등록되어 있으면 A를 통신에 타당한 사용자라고 인정하여 B의 공개키를 알려준다. 그러면 A는 B의 공개키를 이용하여 B에게 통신을 요청한다. B는

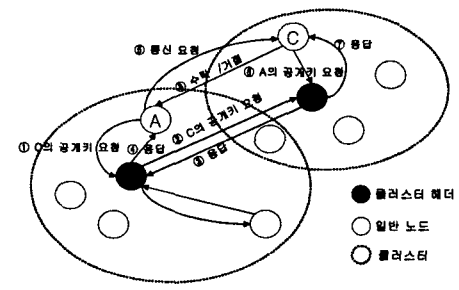
A가 자신과 통신하고자 하는 상대방인지 확인하고자 할 것이다. B는 클러스터 헤더에게 A의 공개키를 요청하고 헤더는 역시 B의 공개키가 자신에게 등록되어있는지 확인하고 등록되어 있으면 A의 공개키를 B에게 알려준다. B는 랜덤 값을 생성하여 A의 공개키로 암호화해서 A에게 보낸다. A는 자신의 비밀키로 그 값을 복호화한 후에 그 값을 다시 B의 공개키로 암호화하여 보낸다. 그 값을 받은 B는 자신이 보낸 랜덤 값과 비교하여 보고 두 값이 일치하면 A가 자신과 통신하고자 하는 노드라고 인식하고 통신을 수락하며, 일치하지 않으면 통신을 거절한다.



[그림 2] 한 클러스터 내에서의 통신

3.4.2. 다른 클러스터 범위에 있는 노드 간 통신

[그림 3]에서와 같이 노드 A가 다른 클러스터 범위에 있는 노드 C와 통신을 하고자 한다면 먼저 A는 자신이 속해있는 클러스터의 헤더에게 C의 공개키를 요청한다. 클러스터 헤더는 A의 공개키가 자신에게 등록되어 있으면 A를 통신에 타당한 사용자라고 인정하여 C의 공개키를 자신의 공개키 테이블에서 찾는다. 만약 자신의 공개키 테이블에서 C의 공개키를 찾을 수 없다면 클러스터 헤드는 이웃 클러스터 헤더의 클러스터링 테이블에서 C를 찾아서 C가 속해있는 클러스터 헤더에게 A가 C와 통신하고 싶어함을 알리고 C의 공개키를 얻어 와서 A에게 응답한다. 그러면 A는 C의 공개키를 이용하여 C에게 통신을 요청한다. C는 A가 자신과 통신하고자 하는 상대방인지 확인하고자 할 것이다. C는 클러스터 헤더에게 A의 공개키를 요청하고 헤더는 역시 A가 포함되어 있는 클러스터 헤더를 찾아서 A의 공개키를 얻어와 C에게 알려준다. C는 랜덤 값을 생성하여 A의 공개키로 암호화해서 A에게 보낸다. A는 자신의 비밀키로 그 값을 복호화한 후에 그 값을 다시 C의 공개키로 암호화하여 보낸다. 그 값을 받은 C는 자신이 보낸 랜덤 값과 비교하여 보고 두 값이 일치하면 A가 자신과 통신하고자 하는 노드라고 인식하고 통신을 수락하며, 일치하지 않으면 통신을 거절한다.



[그림 3] 다른 클러스터 범위에 있는 노드간 통신

3.4.3. 클러스터 헤더의 이동

클러스터 내에 있는 다른 노드들은 클러스터 헤더가 주기적으로 보내는 signal을 통하여 자신이 그 클러스터 범위 내에 있다는 것을 인식한다. 만약 주기적으로 오던 신호가 오지 않게 되면 다른 노드들은 자신이 클러스터 범위에서 벗어났거나 혹은 클러스터 헤더가 fail되었다고 인식하여 signal을 이용하여 이웃노드들을 탐색하게 된다. 이때 자신이 그 클러스터 범위 안에 있음에도 불구하고 signal을 받지 못했다면 이는 클러스터 헤더가 fail되었다고 판단하고 자신의 존재여부와 메모리 용량에 관한 정보를 브로드캐스팅 하여 새로운 클러스터 헤더를 선출한다. 클러스터 헤더는 자신이 이동하고자 하면 그 사실을 자신의 클러스터 범위 안에 있는 노드들에게 알리고 새롭게 헤더를 선출하게 한다.

4. 결론 및 향후과제

본 논문에서는 ad hoc 네트워크에서 공개키 기반 구조를 이용한 노드간 인증 메커니즘을 제안하였다. 이 메커니즘은 헤더가 유선망에 접속하여 안전한 CA에 접속하여 인증을 받으므로 ad hoc 네트워크 내에서 노드들끼리만 행해지는 인증 메커니즘 보다 더욱 신뢰할 수 있다. 또한 헤더들끼리 서로 상호 인증을 통해 그 헤더가 믿을 수 있는 노드라는 것을 다른 노드들에게 증명함을 통하여 인증 받지 못한 헤더로 가장하는 공격에 대해 대비할 수 있다. 그러나 이 메커니즘은 유선망과의 연동을 필수적 요소로 꼽고 있기 때문에 순수 ad hoc 망에서는 사용할 수 없는 단점을 갖는다.

앞으로의 향후 과제로 순수 ad hoc 망에서 사용가능한 인증메커니즘에 대한 연구가 수행되어야 할 것이다.

5. 참고문헌

[1] The mobile ad-hoc networks (MANET) working group, http://www.ietf.org/html.charters/manet-charter.html.

[2] C. -K Toh, "Ad Hoc Mobile Wireless Networks" Prentice Hall PTR 2002.

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

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

[5] S. Zhu, S. Xu, S. setia, S. Jajodia "LHAP: A lightweight ho-by-hop authentication protocol for ad-hoc networks"

[6] L. Zhou and Zygmunt J.Haas, "Securing ad hoc networks", *IEEE Network Magazine, Vol13, No6,* pp.24~30, November/December1999

[7] 송지은 " Ad Hoc 네트워크 환경에서 협력적인 인증키 관리기법" 전북대대학원 석사학위논문

[8] A. Ephremides, J. E. Wieselthier and D. J. Baker, "A design concept for reliable mobile radio networks with frequency hopping signaling", Proc. IEEE 75, 1987

[9] Mario Gerla and Ching-Chuan Chiang, "Multicluster, mobile, multimedia radio network, ACM_Baltzer Journal of Wireless Networks, Vol1, No.3, 1995