논문 04-29-7C-13

한국통신학회논문지 '04-7 Vol.29 No.7C

Ad-Hoc 네트워크에서 선행 키 분배 없는 단 대 단 키 설정 방안

정회원 왕 기 철*, 방 상 원**, 정 병 호***, 조 기 환*

A Peer-to-Peer Key Establishment Scheme without Pre-distributing Keys in Ad-Hoc Networks

Gicheol Wang*, Sangwon Bang**, Byungho Chung***, Gihwan Cho* ***Regular Members***

요 약

임의의 두 노드사이에 전송되는 데이터를 보호하기 위해서 단 대 단 키 설정은 필수적이다. 그러나 동적으로 위상이 변화하고, 동일한 권한을 갖는 노드로 구성되는 Ad-Hoc네트워크 환경에서 선행 키 분배는 비현실적인 가정이다. 본 논문은 Ad-Hoc네트워크에서 미리 키를 분배하지 않고 두 노드사이에 단 대 단 키를 설정하는 방안을 제안한다. 제안하는 방안은 Diffie-Hellman 키교환 방법을 기반으로 한다. 제안하는 방안은 키를 교환하는 과정에서 임의의 해쉬체인 값들을 이용하여 교환되는 Diffie-Hellman값의 위조를 방지한다. 따라서 제안하는 단 대 단 키 설정 방안은 man-in-the-middle공격에 대해 해쉬함수의 안전성 만큼 안전하다. 실험결과는 제안된 방안이 선행 키 분배 방법에 비해 키 설정 과정에서 전송되는 메시지 수를 크게 감소 시킴을 보여준다. 또한 실험결과를 통해 제안된 방안이 상대적으로 확장성이 높은 것으로 평가되었다.

Key Words : peer-to-peer key establishment, Ad-hoc network, key pre-distribution

ABSTRACT

In order to protect an exchanged data, it is indispensable to establish a peer-to-peer key between the two communicating nodes. Pre-distributing keys among the nodes is unrealistic in Ad-Hoc network environment because of the dynamic nature of its network topology and the equal authority of its nodes. This paper presents a peer-to-peer key establishment scheme without pre-distributing keys in Ad-Hoc networks. The proposed scheme is based on the Diffie-Hellman key exchange protocol. Main idea is to prevent the falsification of Diffie-Hellman values using some elements of a hash chain. As a result, it is as safe as the underlying hash function against a man-in-the-middle attack. Simulation results have shown that the proposed scheme dramatically reduces the number of messages, and has relatively higher scalability, as compared with the key pre-distribution based scheme.

Ⅰ. 서 론

Ad-Hoc 네트워크는 AP(Access Point)나 기지국과 같은 기반구조 없이 이동 단말들 로만 구성된 자율적이고 독립적인 네트워크이다. 지금까지 Ad-Hoc 네트워크는 구성이 단순하고 융통성 있으며, 일시적인 필요에 의한 임시 네트워크의 구성에 용이하기 때문에, 전쟁터에서의 통신, 재난구조 상황에서의 통신, 그리고 교실이나 회의실에서 즉석통신과 같은 다양한 분야로의 응용이 논의되어 왔다.

* 전북대학교 컴퓨터 통계정보학과 분산/이동컴퓨팅 연구실({gcwang,ghcho}@dcs.chonbuk.ac.kr)
** 송원대학 컴퓨터 정보과(swbang@songwon.ac.kr), *** 한국 전자통신 연구원 무선LAN 보안연구팀(cbh@etri.re.kr)
논문번호 : 040153-0419, 접수일자 : 2004년 4월 19일

Ad-Hoc 네트워크는 향후 유비쿼터스 컴퓨팅[1]을 실현하기 위한 개인과 개인간 통신, 개인과 사물간 통신, 사물과 사물간의 통신을 위한 기반망의 역할을 수행할 것으로 기대된다.

그러나 Ad-Hoc 네트워크가 널리 사용되기 위해서는 먼저 네트워크 상의 각 노드들에게 안전한 통신을 보장할 수 있어야 한다. Ad-Hoc 네트워크는 무선 매체를 사용하기 때문에 유선 네트워크에 비해 훨씬 더 많은 보안상 위험을 지닌다[2]. 즉, Ad-Hoc 네트워크에서는 멀티 홉 방식에 의해 라우팅 되기 때문에 악의적인 중간 노드에 의한 데이터의 무결성 및 기밀성 문제가 발생한다. 특히, 신뢰할 수 없는 매체를 이용해서 통신해야 하므로 암호키에 크게 의존하게 된다. 따라서 키 사이에 신뢰할 수 있는 관계를 형성하고, 이를 노드들 간에 안전하게 분배하는 것이 요구된다. 통신 주체간에 키가 한 번 설정되면, 이를 이용하여 기밀성, 인증, 무결성, 그리고 부인봉쇄와 같은 보안 서비스의 제공이 가능하다.

Ad-Hoc 네트워크는 인증기관이나 키 분배 서버와 같은 신뢰기관이 존재하지 않으므로, 네트워크에 참여한 노드들끼리 자율적이고 협력적인 방법으로 키의 분배 및 관리를 수행하여야 한다. 또한, Ad-Hoc 네트워크를 구성하는 노드들은 유선 네트워크에 비해 가용자원이 빈약하므로, 심각한 계산부하 및 통신부하를 유발하지 않는 키 분배 및 관리 방법이 요구된다.

현재 Ad-Hoc 네트워크에서의 키 관리를 위한 기존의 연구들은 네트워크내의 모든 노드가 동일한 그룹 키를 설정하기 위한 연구나 인증기관의 기능을 각 노드에게 분산시켜 효율적으로 공개키 관리를 수행하기 위한 연구들이 대부분이다. 반면에 Ad-Hoc네트워크에서 단 대 단 키를 설정하기 위한 연구들은 아직까지는 미흡한 실정이다.

최근에 제안된 Ad-Hoc 네트워크를 위한 단 대 단 키 설정방법[3]은 네트워크의 형성 이전에 미리 키풀(Key pool)서버로부터 임의의 키들을 분배 받고 이들을 이용하여 통신주체간에 단 대 단 키를 설정한다. 이 방법은 네트워크 내의 노드들이 단일 도메인에 속하는 센서 네트워크에 적합하며, 일반적인 Ad-Hoc 네트워크에 적용할 경우 여러 가지 문제점이 야기된다. 즉, 이러한 방법은 공유되는 키들과 노드의 수가 미리 정해지므로, 소속기관이 다른 Ad-Hoc 네트워크간 협동작업이 불가능하다. 또한 임의의 노드가 네트워크에 새로 참여하고자 할 경우에, 이 노드는 미리 키 분배 서버로부터 임의의 키들을 off-line으로 획득하여야 한다. 따라서 미리 키들을 분배받는 방법은 네트워크의 융통성 및 확장성을 크게 감소시킨다.

본 논문에서는 Ad-Hoc 네트워크에서 미리 임의의 키들을 분배받지 않고 두 노드간에 단 대 단 키를 설정하는 방법을 제안한다. 이를 위해 제안하는 방법은 잘 알려진 Diffie-Hellman 키 교환 프로토콜에 기반 하여 두 노드간에 단 대 단 키를 설정한다. 이때, man-in-the-middle공격을 방지하기 위해 제안하는 방법은 Lamport의 해쉬체인[4]을 이용하여 교환되는 Diffie-Hellman 값들을 위조하지 못하도록 한다.

이를 위해 제안하는 방법은 키설정의 주체들이 자신의 키 체인 중에서 임의로 연속되는 세개의 값을 선택하고 역순으로 공개한다. 즉, 키설정의 주체들은 키설정 메시지를 두 단계로 나누어 전송하되 1단계 메시지에는 세번째 키체인 값과 두번째 키체인 값으로 암호화된 자신의 Diffie-Hellman값을 포함한다. 이때 세번째 키체인 값은 1단계에서는 인증을 위해 사용되고, 2단계 메시지에 포함되는 첫번째 키체인 값의 정확성을 판단하기 위한 근거로 사용된다. 따라서 임의의 공격자가 man-in-the-middle 공격을 성공적으로 수행하기 위해서는 1단계 및 2단계 메시지를 모두 위조해야 한다. 그러나 해쉬함수의 단방향성으로 인해 이러한 두 단계 메시지를 모두 위조하기는 어렵다.

본 논문은 다음과 같이 구성된다. 2장에서는 Ad-Hoc네트워크에서의 키 관리 방법에 관한 기존 연구들에 대해 간략히 기술한다. 3장에서는 본 논문에서 제안하는 미리 키를 분배하지 않고 단 대 단 키를 설정하는 방법을 자세히 기술한다. 4장에서는 제안하는 방법의 안전성 및 효율성을 입증하기 위해 안전성 분석 및 실험결과를 제공하고 5장에서는 결론을 내린다.

Ⅱ. 관련 연구

1. 분산 인증기관 (Distributed CA)기법

(k, n) 부분 분산 인증기관 기법[5][6][7]은 특정 노드 집합이 인증기관의 기능을 수행하는 방법이다. 만일 임의의 노드가 자신의 인증서 발행을 요구하면, 이러한 특정 노드들이 각각 자신의 부분 비밀키로 서명된 부분 인증서를 그 노드에게 전송한다. 그 클라

이이언트 노드는 이러한 부분 인증서 k 개를 병합함으로써 자신의 실제 인증서를 획득하게 된다.

부분 분산 인증기관 기법[5][6]은 인증기관의 기능을 수행할 수 있는 노드를 일부 노드로 제한 시키고 이들을 적절히 분산시킨 방법이다. 이 방법은 각 클라이언트 노드가 자신의 인증서를 얻기 위해 최소한 인증기관 의 기능을 수행하는 노드 k개를 검색해야 하므로 많은 통신 오버헤드를 유발한다. 인증서를 갱신하거나 취소하는 경우에도, 이 방법은 또한 인증서의 동기화를 위한 추가적인 통신이 필요하다.

반면에, 완전 분산 인증기관 기법[7]은 네트워크내의 모든 노드들이 인증기관의 기능을 수행하도록 하는 기법이다. 따라서 이 방법은 인증서를 요청하는 노드들에게 높은 가용성을 제공하고 인증서 발행 시 네트워크 전체에서 발생하는 트래픽의 양을 감소시킨다. 그러나 이 방법은 모든 호스트가 부분 비밀 키를 보유하고 있으므로, 일부의 특정 노드만 부분 비밀 키를 가진 부분 분산 기법에 비해 많은 부분 비밀 키들이 위험에 노출된다. 따라서 이 기법에서 임계치 파라미터 k값은 공격자가 짧은 시간 내에 k개의 부분 비밀 키를 얻을 수 없도록 충분히 큰 값이어야 한다. 그러나 k값이 커지면 그에 따라 통신부하가 커지고 가용성이 저하 된다.

두 가지 분산 인증기관 기법은 네트워크가 분할되었다가 다시 병합되는 경우 동기화 문제를 일으킨다. 예를 들어, 단일 네트워크가 두개의 네트워크로 분할되었을 때, 각각 부분 키를 수정했다고 가정해보자. 이후에 다시 두개의 네트워크가 단일 네트워크로 통합된다면, 이때 인증기관의 기능을 수행하는 노드들은 서로 다른 비밀 키의 부분 키 값들을 보유하게 된다.

2. 클러스터 구조 기반의 그룹 키 관리 기법

일반적으로 Ad-Hoc 네트워크에서 클러스터 구조는 클러스터 헤드들이 다수의 멤버 노드들과 직접 연결되게 한다. 따라서 클러스터 헤드들은 자신들이 가진 정보를 멤버들에게 한번의 방송으로 즉시 전달할 수 있다. 따라서 클러스터 구조에서 각 클러스터 헤드는 클러스터를 대표하는 지역 방송자 역할을 수행할 수 있다[8].

참고문헌 [9]에서는 센서 네트워크에서 모든 노드들이 같은 그룹 키를 공유하는 방법이 제안되었다. 이 방법은 먼저 클러스터를 구성하고 몇몇 클러스터 헤드들을 임시 키 관리자(Potential Key Manager)들로 선정한다. 임시 키 관리자들은 그들의 키들을 생성하고 모든 클러스터 헤드들에 전파한다. 각 클러스터 헤드들은 그 키들 중에서 하나만을 임의의 기준에 따라 그룹 키로 선정한다. 그룹 키가 일치된 후에, 각 클러스터 헤드는 그 그룹 키를 자신의 멤버들에게 전송함으로써 모든 노드가 동일한 키를 보유하게 된다. 따라서 임의의 두 노드 간에 교환되는 데이터는 두 노드사이에 존재하는 모든 노드들에게 노출된다. 또한 이 기법은 단지 임의의 노드에 대한 개별인증이 아닌 그룹에 대한 멤버여부에 대해서만 인증을 수행한다. 더구나 초기 그룹 키로부터 대부분의 임시 키들이 유도되므로, 초기 그룹 키가 노출되면 전체 시스템의 보안이 크게 훼손된다.

3. 확률적인 키 공유기법 기반의 단 대 단 키 설정 방법

참고문헌 [3]에서는 확률적인 키 공유 기법을 이용한 단 대 단 키 설정 방법이 제안되었다. 이 방법에서 단 대 단 키를 설정하고자 하는 소스노드는 요구되는 보안 수준에 따라 적절한 부분키의 개수를 정하고, 설정된 경로와 proxy 요청메시지의 방송을 통해 이 수 만큼의 proxy노드를 획득한다. 이때 proxy노드들은 소스노드는 물론 목적노드와도 공유된 키들을 가지는 노드들이다. 소스노드는 임의의 단 대 단 키를 생성하고 이를 부분키의 수만큼 분할한 후에 이들을 각각의 proxy노드들을 통해 목적노드에게 전송한다. 즉, 소스노드는 임의의 proxy노드와 공유된 키들을 이용하여 부분키를 암호화해서 proxy노드로 전송하고, proxy노드는 이를 해독하여 다시 목적노드와 공유된 키들로 부분키를 암호화하여 전송한다. 따라서 임의의 악의적인 노드는 임의의 두 노드간의 단 대 단 키를 얻기 위해서는 부분키들의 암호화에 사용된 모든 초기 분배 키들을 획득해야 한다.

그림 1은 초기 키 서버가 15개의 키중에서 5개씩을 각 노드들에게 분배하고 소스노드가 단 대 단 키를 설정하기 위하여 proxy노드를 획득하는 과정을 보인다. proxy노드들은 목적노드로 부터 RREP패킷에 기록된 노드 정보를 통해 획득한다. 만일 proxy노드의 수가 부분키의 수에 미치지 못할 경우에는 proxy 요청메시지를 방송하여 부족한 proxy노드를 획득한다. 그림 1에서 보는 것처럼 노드 m, w, x, y는 모두 소스 노드 및 목적 노드와 공유된 키를 최소 한 개 이상씩 가지고 있으므로, proxy 노드로

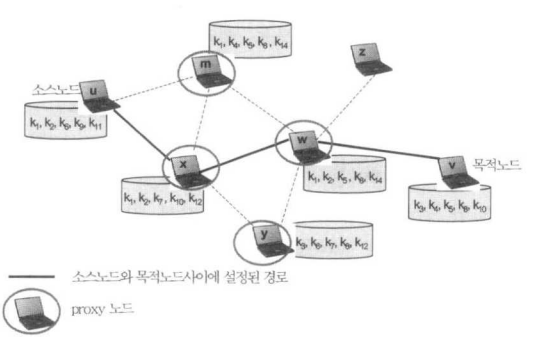


그림 1. 확률적인 키 공유 기반의 단 대 단 키 설정 방법

서 동작할 수 있다.

이 방법은 키풀에 있는 각 키가 임의의 노드에게 분배될 확률에 따라 네트워크의 보안성이 좌우된다. 즉, 키풀에 있는 각 키가 임의의 노드에게 분배될 확률이 작을수록 네트워크의 보안성이 향상된다. 또한 부분키의 개수가 많으면 많을수록 네트워크의 보안성은 크게 향상된다. 그러나 이는 많은 부분키의 전송을 위한 높은 통신부하를 유발하여 시스템의 성능을 저하시킨다. 만일 노드의 수가 증가한다면 악의적인 노드들의 공모에 의한 보안성 악화의 위험성이 높아진다. 이를 방지하기 위해서는 더 많은 proxy노드가 필요하게 되고, 이는 다시 이들 proxy노드와의 통신으로 인한 통신부하를 유발한다.

Ⅲ. 미리 키를 분배하지 않고 단 대 단 키를 설정하는 방법

본 논문에서 제안하는 단 대 단 키 설정 방법은 미리 키를 분배하지 않고 최소의 정보만으로 두 노드 사이에 단 대 단 키를 설정하기 위하여 Diffie-Hellman 키교환 방법을 기반으로 한다. 그러나 Diffie-Hellman 키교환 방법은 잘 알려진 것처럼 man-in-the-middle공격에 취약하다. 더구나 Ad-Hoc 네트워크에서 임의의 두 노드간의 경로는 멀티 홉으로 구성되므로, 이러한 공격의 위험성은 더욱 높아진다.

단순한 해결책은 공개키/비밀키 쌍을 이용하여 Diffie-Hellman값을 암호화하거나 서명하는 방법을 이용할 수 있다. 즉, 전송되는 Diffie-Hellman값들이 만일 상대방의 공개키로 암호화되면 악의적인 노드는 상대방 노드의 비밀키를 획득하기 전에는 이 Diffie-Hellman값도 알 수 없다. 만일 통신주체가 전송하는 Diffie-Hellman값을 자신의 비밀키로 암호화하면, 악의적인 노드는 그 노드의 비밀키를 획득하기 전에는 그 노드인 것처럼 위장할 수 없다. 그러나 일반적으로 이러한 공개키 연산은 많은 계산부하를 유발하고 소요시간이 길다. 따라서 제한된 전력으로 운용되는 Ad-Hoc 단말에게 이러한 공개키 연산은 부적절하다.

이에 따라 제안하는 방법은 공개키 연산을 사용하지 않고 비교적 계산부하가 작은 해쉬연산과 대칭키 연산만을 이용하여 Diffie-Hellman값을 위장하지 못하도록 한다. 이를 위해 임의의 통신주체는 해쉬체인을 생성하고 임의의 해쉬체인 값을 이용하여 자신의 Diffie-Hellman값을 암호화하고 다음 해쉬체인 값을 첨부하여 상대방 노드에게 전송한다. 상대방 노드는 첨부된 해쉬체인 값을 이용해서 통신 주체를 인증하고 같은 형식의 메시지를 구성하여 응답한다. 통신주체는 상대방 노드로부터 응답을 수신하면, 자신의 Diffie-Hellman값을 암호화하는데 사용했던 값의 이전 해쉬체인 값을 전송하여 상대방 노드가 자신의 Diffie-Hellman값을 획득하도록 한다. 이때 상대방 노드는 두번째 메시지에서 획득한 해쉬체인 값을 2번 해쉬연산을 수행해서 첫번째 메시지에서 수신했던 해쉬체인 값과 동일한지 점검한다. 즉, 이러한 검사가 성공한 경우에만 자신의 Diffie-Hellman값을 복호화하는데 사용될 해쉬체인 값을 전송하고 그렇지 않은 경우에는 더 이상 메시지를 전송하지 않는다. 따라서 임의의 공격자에 의한 위조된 Diffie-Hellman값은 수신이 거부되고, 그 공격자는 man-in-the-middle공격을 수행하는데 필요한 추가 정보를 얻을 수 없다.

제안하는 방법의 기술을 위해 모든 노드는 해쉬체인을 생성하는 해쉬함수 h와 Z_p^*의 생성자 g를 알고 있다고 가정한다. 네트워크에 진입한 임의의 노드 A는 다음과 같은 과정을 수행한다. 다음에서 x_k^A는 노드 A의 비밀키(x_0^A)가 해쉬함수 h 에 의해 k번 해쉬됨을 의미한다.

- 임의의 노드 A는 자신의 비밀키인 x_0^A와 임의의 수 n_A를 생성한다.
- 노드 A는 자신의 공개키인 $x_{n_A}^A = h^{n_A}(x_0^A)$을 생성한다.
- 노드 A는 각각 자신의 라우팅 메시지에 자신의 공개키를 포함하여 전송한다. 따라서 임의의 노드는 키 설정을 원하는 상대방 노드의 공개키를

미리 알게 된다.

이후에 임의의 노드 A와 B는 [그림 2]와 같은 과정을 통해 단 대 단 키 설정을 수행한다. [그림 2]에서 $\{d\}x^{A}_{n_A-i}$은 데이터 d를 대칭 키 $x^{A}_{n_A-i}$로 암호화 하는 것을 의미 한다. 다음은 [그림 2]를 기준으로 노드 A와 B가 단 대 단 키를 설정하는 과정을 단계별로 자세히 기술한 것이다.

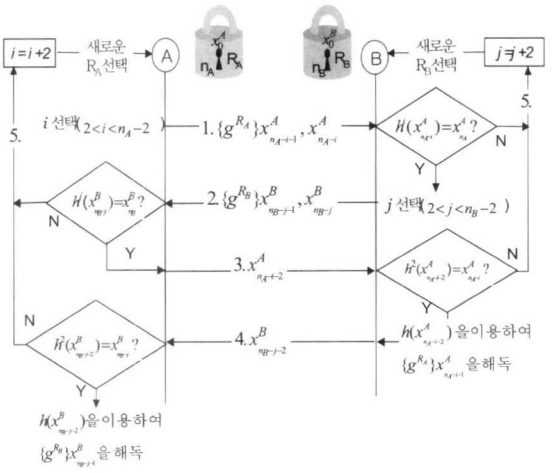


그림 2. 미리 키를 분배하지 않는 단 대 단 키 설정 방법

1. 노드 A는 임의의 수 i $(2<i<n_A-2)$를 선택하고, $x^{A}_{n_A-i-1}$ 와 $x^{A}_{n_A-i}$을 계산한다. 노드 A는 자신이 생성한 Diffie-Hellman 값(g^{R_A})을 $x^{A}_{n_A-i-1}$를 사용하여 암호화 한다. 그리고 나서 노드 A는 $x^{A}_{n_A-i}$을 첨부하여 노드 B에게 전송한다.
2. 노드 B는 수신한 $\{g^{R_A}\}x^{A}_{n_A-i-1}$를 저장하고 $h^{i}(x^{A}_{n_A-i})=x^{A}_{n_A}$ 인지 확인한다. 만일 맞다면, 노드 B는 임의의 수 j $(2<j<n_B-2)$를 선택하고, $x^{B}_{n_B-j-1}$와 $x^{B}_{n_B-j}$을 계산한다. 그 후 노드 B는 자신이 생성한 Diffie-Hellman 값(g^{R_B})을 $x^{B}_{n_B-j-1}$를 사용하여 암호화 한다. 그리고 나서 노드 B는 을 $x^{B}_{n_B-j}$을 첨부하여 노드 A에게 전송한다.
3. 노드 A는 수신한 $\{g^{R_B}\}x^{B}_{n_B-j-1}$를 저장하고 $h^{j}(x^{B}_{n_B-j})=x^{B}_{n_B}$ 인지 확인한다. 만일 맞다면, 노드 A는 암호화 키의 이전 키 체인 값인 $x^{A}_{n_A-i-2}$를 계산하고 노드 B에게 전송한다.
4. 노드 B는 $h^{2}(x^{A}_{n_A-i-2})=x^{A}_{n_A-i}$ 인지 확인한다. 만일 맞다면, 노드 B는 $h(x^{A}_{n_A-i-2})$를 이용하여 저장된 $\{g^{R_A}\}x^{A}_{n_A-i-1}$를 복호화 한다. 이후에, 노드 B는 암호화 키의 이전 키 체인 값인 $x^{B}_{n_B-j-2}$를 계산하고 노드 A에게 전송한다.

   그 후에 노드 A는 $h^{2}(x^{B}_{n_B-j-2})=x^{B}_{n_B-j}$인지 확인한다. 만일 맞다면, 노드 A는 $h(x^{B}_{n_B-j-2})$를 이용하여 저장된 $\{g^{R_B}\}x^{B}_{n_B-j-1}$를 복호화 한다.

   만일 위의 키 설정 과정 중에 임의의 검사가 실패하거나 일정 시간 내에 필요한 메시지를 모두 수신하지 못하는 경우에는 다음과 같은 과정을 수행한다.
5. 노드 A와 B는 각각 $i=i+2$ 그리고 $j=j+2$를 계산하고 새로운 Diffie-Hellman값(g^{R_A}, g^{R_B})을 생성한다. 이후에 노드 A와 B는 1단계부터 4단계를 다시 수행한다.

Ⅳ. 보안성 분석 및 효율성 분석

1. 보안성 분석

Ad-Hoc 네트워크에서 임의의 두 노드간에 단 대 단 키를 설정할 때 다양한 형태의 공격을 받을 수 있다.

먼저 가장 치명적인 문제로 서비스 거부 공격(Denial of Service)이 있다. 즉, 키설정의 소스와 목적노드 사이에 있는 중간노드들이 자신의 단말 전원을 절약하기 위해 혹은 고의적으로 패킷중계를 거부하는 형태의 공격이다. 사실 이 경우에는 키설정 방법 자체의 문제가 아니므로 침입 혹은 비정상 행위 탐지 문제와 연계되어 해결책이 고안되어야 한다. 참고문헌 [14]와 [15]에서는 중간노드에 의한 비정상 행위의 탐지 및 예방에 관한 의미 있는 해결책을 제시하였다.

재전송(replay) 공격은 이전에 합법적인 사용자로부터 전송된 메시지들을 저장했다가 다시 전송하여 합법적인 사용자인 것처럼 위장하는 공격이다. 그러나 키 설정의 주체들이 수신한 메시지를 저장하고 이전 메시지와 비교한다면, 이러한 공격은 즉시 키

설정의 주체들에게 알려지게 된다.

man-in-the-middle공격은 Ad-Hoc 네트워크에서 키설정의 주체간에 설정되는 경로는 멀티 홉으로 구성될 수 있다는 점에서 매우 중요하다. 즉, 키설정의 주체 사이에 존재하는 중간노드들에 의해 키의 무결성 및 기밀성이 훼손가능 하므로 이를 방지하기 위한 보안기법의 설계는 필수적이라 할 수 있다. 따라서 본 논문에서는 Ad-Hoc 네트워크에서 단대 단 키 설정시 man-in-the-middle공격에 대한 제안하는 방법의 안전성을 분석하였다.

임의의 악의적인 노드가 Diffie-Hellman 키교환을 수행하는 노드 사이에서 man-in-the-middle공격을 수행하기 위해서는 두 가지 조건을 만족해야 한다. 먼저 두 통신주체로부터 전송되는 Diffie-Hellman값을 모두 알아야 한다. 둘째, 악의적인 노드는 자신의 위조된 Diffie-Hellman값이 실제 통신주체로부터 전송된 값인 것처럼 위장해서 전송함으로써 통신주체들이 이 값을 신뢰하도록 해야 한다. 만일 두 가지 사항 중에 하나라도 만족되지 않으면 man-in-the-middle공격은 실패로 끝나게 된다.

제안하는 방법은 악의적인 노드로부터 전송된 위조된 Diffie-Hellman값을 식별하기 위해, 단일 노드가 전송하는 키 설정 메시지를 두 번 나누어 전송하도록 한다. 즉, 첫 번째 메시지에는 암호화된 Diffie-Hellman값과 자신의 인증을 위한 해쉬체인 값을 전송하도록 하고, 두 번째 메시지에는 자신의 Diffie-Hellman값을 암호화하는데 사용한 해쉬체인 값의 이전 값을 전송하도록 한다. 이때 첫 번째 메시지와 두 번째 메시지에서 사용된 해쉬체인 값들은 연속적인 값들이고 역순으로 노출된다. 그러나 해쉬함수는 단방향성을 가지므로 임의의 해쉬체인 값을 이용하여 이전의 값을 얻기는 매우 어렵다. 따라서 제안하는 방법의 안전성은 기본적인 해쉬함수의 안전성에 의해 좌우된다.

정리 1. 제안하는 방법은 man-in-the-middle공격에 대해 해쉬함수의 안전성 만큼 안전하다.

증명: 먼저 임의의 공격자가 키 설정을 시도하는 두 통신주체 사이에서 어떠한 정보도 없이 man-in-the-middle 공격을 시도한다고 가정하자. 예를 들어 그림 3에서 노드 A는 노드 B와 단 대 단 키를 설정하려고 한다. 이때 노드 A와 B사이에 있는 노드 C는 man-in-the-middle공격을 시도한다. 이를 위해 공격자 C는 단계 1과 단계 2에서 원래 전송된 Diffie-Hellman 값들($\{g^{R_A}\}x^A_{n_A-i-1}$, $\{g^{R_B}\}x^B_{n_B-j-1}$) 대신에 자신이 생성한 값들(즉, $\{g^{R_C}\}x^C_{n_C-i-1}$, $\{g^{R_C}\}x^C_{n_C-j-1}$)을 키설정의 주체들에게 전송해야 한다. 또한 단계 3과 4에서 공격자 C는 키설정의 주체들이 C의 Diffie-Hellman값을 얻도록 하기 위해, 단계 1과 2에서 암호화에 사용된 키의 이전 키들($x^C_{n_C-i-2}$, $x^C_{n_C-j-2}$)을 공개해야 한다. 그러나 이 키들이 공개될 때, 키설정의 주체들은 man-in-the-middle공격을 쉽게 감지할 수 있다. 이것은 키설정의 주체들이 이 키들을 이용해서 이전에 수신된 키 값들($x^A_{n_A-i}$, $x^B_{n_B-j}$)과 일치하는지 확인하기 때문이다. 즉, 그림 3에서 공격자 C에 의한 man-in-the-middle공격은 단계 3에서 알려지게 되고, 노드 B와 A는 단계 5를 수행한다. 따라서 공격자 C는 더 이상의 정보를 얻을 수 없다.

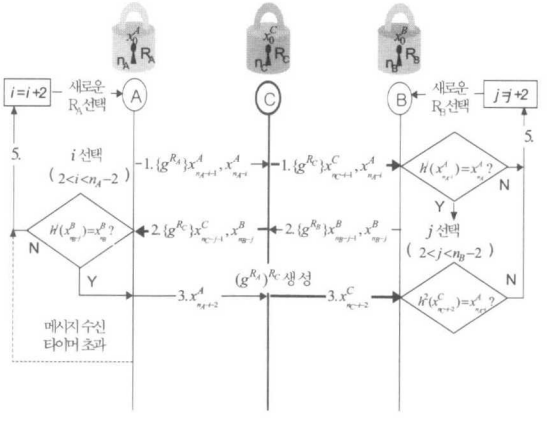


그림 3. 단 대 단 키 설정에 대한 man-in-the-middle공격

다음으로 임의의 공격자가 이전의 키 설정과정에서 전송된 정보들을 이용하여 man-in-the-middle공격을 시도한다고 가정하자. 이를 위해 공격자는 단계 4까지의 정보를 얻은 후에 메시지 전송을 거부하여 키설정이 실패하도록 한다. 이후에 키설정의 주체들은 단계 5를 통해 키 설정을 다시 시도한다. 그러나 공격자는 이전 키설정 과정에서 얻은 정보를 이용하여 man-in-the-middle공격을 수행할 수 없다. 이것은 한번 키설정이 실패하면, 이미 사용되었던 해쉬체인 값들은 더 이상 전송되는 Diffie-Hellman값의 암호화에 이용되지 않기 때문이다.

따라서 임의의 공격자가 man-in-the-middle공격을 성공적으로 수행하기 위해서는 단계 1과 단계 2에

서 공개되는 키체인 값을 이용하여 역해쉬를 취해야 한다. 임의의 공격자가 man-in-the-middle 공격을 성공적으로 수행했다고 가정하자. 즉 이것은 그 공격자가 단계 1과 단계 2에서 공개되는 $x^{A}_{n_A-i}$과 $x^{B}_{n_B-j}$을 이용하여 $x^{A}_{n_A-i-1}\,(2<i<n_A-2)$ 과 $x^{B}_{n_B-j-1}\,(2<j<n_B-2)$ 을 유도했음을 의미한다. 그러나 이것은 해쉬함수의 단방향성에 위배된다.

2. 효율성 분석

본 논문에서 제안하는 방법의 효율성을 입증하기 위해, 무선 Ad-hoc 네트워크를 위한 시뮬레이터를 CSIM 18 엔진과 C언어를 이용하여 개발하였다. 본 시뮬레이터는 일정영역 내에서 랜덤하게 이동하며 단 대 단 키 설정을 위해 주기적으로 패킷을 교환하는 호스트들을 프로세스로 모델링하였다.

초기에 100개의 노드는 1000m×1000m의 평면에서 랜덤하게 분포되었으며, 각 노드는 임의의 시간에 임의의 방향으로 랜덤하게 이동할 수 있었다. 두 노드사이의 단 대 단 키설정 주기는 2초로 설정하였고 키설정의 주체들은 랜덤하게 선정되었다. 표 1은 실험을 위한 파라미터들의 설정값을 보여준다.

표 1. 실험 파라미터들

파라미터	값
노드의 수	100, 150, 200
호스트의 이동속도	A(0~5m/s), B(5~10m/s), C(10~15m/s)
전송범위	50~500meter
단 대 단 키 설정 주기	2초
실험 영역	1000m×1000m

본 실험에서 제안하는 방법은 확률적인 키공유 기반의 단 대 단 키 설정방법[3]과 비교되었다. 전송범위가 증가함에 따라 단 대 단 키 설정동안 전송되는 평균메시지의 수가 측정되었다.

그림 4에서 보이는 바와 같이 제안하는 방법과 확률적인 키공유 기반의 방법 모두 전송범위가 증가함에 따라 키 설정시 전송되는 메시지의 수는 감소한다. 이는 전송범위가 증가함에 따라 키 설정의 주체인 두 노드 사이의 홉수가 감소되기 때문이다.

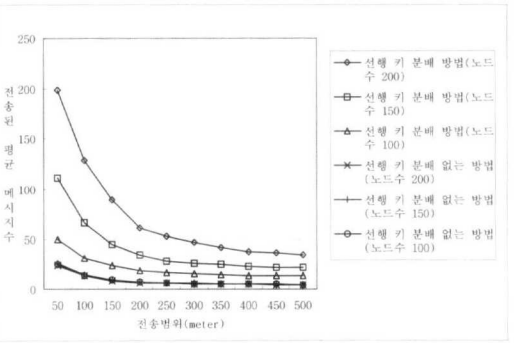


그림 4. 전송범위의 증가에 따른 평균 전송 메시지

그림 4를 통해 알 수 있는 또 하나의 사실은 제안하는 방법은 확률적인 키공유 기반의 방법에 비해 전송되는 메시지의 수가 훨씬 작다는 것이다. 이는 확률적인 키공유 기반의 방법에서 키설정의 주체들은 여러 proxy 노드와 메시지를 교환해야 하는 반면에, 제안하는 방법에서는 키 설정의 주체 사이에서만 메시지가 교환되기 때문이다. 또한 제안하는 방법은 노드의 수가 증가하여도 전송되는 메시지수의 변화가 거의 없다.

확률적인 키공유 기반의 방법과 제안하는 방법의 확장성(scalability)을 평가하기 위해 노드의 수를 점차적으로 증가시켜 가면서 전송되는 메시지의 수를 비교하였다. 이때, 확률적인 키공유 기반의 방법은 단 대 단 키의 노출확률에 따라 전송되는 메시지의 수가 변화하므로, 단 대 단 키의 노출확률을 증가함에 따라 그 방법에서 전송되는 평균메시지의 수가 측정되었다.

그림 5 에서 확률적인 키공유 기반의 방법은 노드수가 작을 때(=30) 제안하는 방법에 비해 전송되

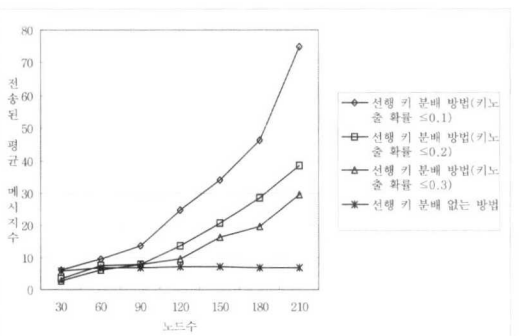


그림 5. 노드수의 증가에 따른 평균 전송 메시지

는 메시지의 수가 더 작다. 이는 미리 분배 받은 키의 노출확률이 낮아지므로, proxy 노드의 수도 감소하기 때문이다. 그러나 노드수가 증가함에 따라, 미리 분배 받은 키들의 노출확률이 높아지므로, 이들의 노출로 인한 단 대 단 키의 노출을 방지하기 위해 더 많은 proxy 노드들을 필요로 하게 된다. 따라서 키설정의 주체들과 이들 proxy노드들과의 메시지 교환이 추가되므로 전송되는 메시지의 수도 증가하게 된다. 반면에 제안하는 방법은 노드의 수가 증가하여도 전송되는 메시지의 수가 거의 일정하다. 이는 제안하는 방법에서의 전송되는 메시지의 수는 노드 수와는 상관없이 랜덤하게 설정되는 키 설정의 주체들 사이의 홉수에 의해 결정되기 때문이다. 그림 5에서 알 수 있는 또 하나의 사실은 확률적인 키공유 기반 방법에서 단 대 단 키의 노출확률을 증가시키면 두 방법사이의 평균 전송 메시지수의 차이가 점차 감소한다는 것이다. 그러나 이로 인해 네트워크의 보안 수준은 저하된다.

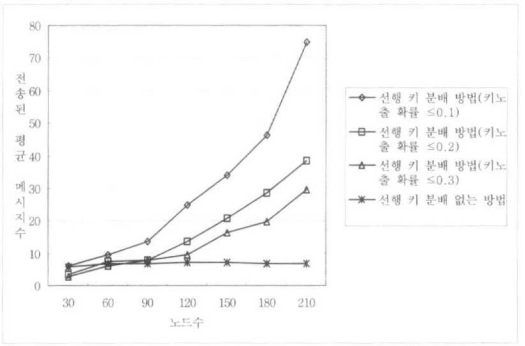


그림 5. 노드수의 증가에 따른 평균 전송 메시지

V. 결 론

본 논문에서는 Ad-Hoc 네트워크에 적절한 단 대 단 키를 설정하는 방법을 제안하였다. 제안된 방법은 두 노드간에 단 대 단 키를 설정하기 위해 임의의 키들을 네트워크 구성 이전에 분배할 필요가 없으므로, 여러 가지 이점을 제공한다. 즉, 다양한 Ad-Hoc네트워크로의 응용에 이용될 수 있고, 다른 도메인에 속한 Ad-Hoc 네트워크간에 안전한 통신을 수행할 수 있다. 또한 새로운 노드들이 네트워크 참여할 경우에도, 미리 off-line으로 키를 분배 받을 필요가 없다.

보안성 분석을 통해 제안된 방법은 임의의 공격자에 의한 man-in-the-middle공격에 대해 해쉬함수의 안전성만큼 안전하다는 것을 증명하였다. 또한 우리의 실험결과는 제안된 방법을 사용함으로써 단 대 단 키설정을 위해 전송되는 평균 메시지수가 크게 감소될 수 있음을 보였다. 더불어 노드의 수가 지속적으로 증가하여도 제안하는 방법은 단 대 단 키의 설정을 위해 전송되는 메시지의 수가 크게 변화하지 않으므로, 미리 키를 분배하는 방법에 비해 확장성이 높다는 것을 증명하였다.

참 고 문 헌

[1] M. Weiser, "The Computer for the 21st Century," *Scientific American*, pp. 66-75, 1991

[2] L. Venkatraman and D.P. Agrawa, "A Novel Authentication scheme for Ad Hoc Networks," Proc. of IEEE WCNC 2000, 2000, pp. 1268-1273

[3] S. Zhu, S. Xu, S. Setia, and S. Jajodia, "Establishing Pair-wise Keys For Secure Communication in Ad Hoc Networks: A Probabilistic Approach," Proc. of the 11th International Conference on Network Protocols(ICNP'03), 2003, pp. 326-335

[4] L. Lamport, "Password Authentication with Insecure Communication," *Communication of the ACM*, 24 (11), pp. 770-772, 1981

[5] L. Zhou and Z. J. Haas, "Securing Ad Hoc Networks," *IEEE Networks*, 13(6), pp. 24-30, 1999

[6] S. Yi and R. Kravets, "Key Management for Heterogeneous Ad Hoc Wireless Networks, "Univ. of Illinois, Urbana-Champaign, Technical Report UIUCDCS-R-2002-2290, 2002

[7] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks," IEEE ICNP 2001, 2001

[8] C. Chiang, H. Wu, W. Liu and M. Gerla, "Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel (CGSR)," Proc.of IEEE SICON'97, 1997, pp. 192-211

[9] S. Basagni, K. Herrin, E. Rosti, and D. Bruschi, "Secure Pebblenets," Proc. of the ACM Symposium on MobiHoc, 2001, pp. 156-163

[10] N. Asokan andP. Ginzboorg, "Key Agreement in Ad-hoc Networks," *Computer Communication Review*, 23(17), pp. 1627-1637, 2000

[11] J-P. Hubaux, L. Buttyan, and S. Capkun, "The Quest for Security in Mobile Ad Hoc Networks,"Proc. of the ACM Symposium on MobiHoc, 2001

[12] K. Sanzgiri, B. Dahill, B. Levine, C. Shields, E. Belding-Royer, "A Secure Routing Protocol for Ad Hoc Networks," Proc. of ICNP'02, 2002, pp. 78-87

[13] A. Weimerskirch and D. Westhoff, "Zero Common-Knowledge Authentication for Pervasive Networks," Proc. of Selected Areasin Cryptography(SAC'03), 2003

[14] S. Marti, T. J. Giuli, Kevin Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," Proc. of the 6th Int'l Conference on Mobile Computing and Networking, 2000, pp. 255-265

[15] L. Buttyan and J. P. Hubaux, "Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks," *ACM/Kluwer Mobile Networks and Applications(MONET)*, 8(5), pp. 579-592, 2001

왕 기 철(Gicheol Wang) 정회원

1997년 2월 : 광주대학교 전자계산학과 졸업
2000년 2월 : 목포대학교 멀티미디어 공학과 석사
2001년 3월~' 컴퓨터통계정보학과 박사과정
<관심분야> Ad-Hoc 네트워크, 센서 네트워크, 이동 컴퓨팅, 무선 네트워크 보안

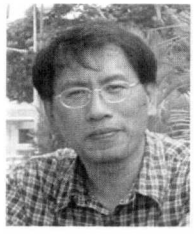
방 상 원(Sangwon Bang) 정회원

1985년 2월 : 전남대학교 계산통계학과 졸업
1988년 2월 : 전남대학교 계산통계학과 석사
1995년 2월 : 전남대학교 계산통계학과 박사
1987년~현재 : 송원대학 컴퓨터 정보과 교수
<관심분야> 이동 컴퓨팅, 컴퓨터 그래픽, 소프트웨어 공학

정 병 호(Byungho Chung) 정회원

1988년 2월 : 전남대학교 전산통계학과 졸업
2000년 2월 : 충남대학교 컴퓨터과학과 석사
2000년 3월~현재 : 충남대학교 컴퓨터과학과 박사 수료
1998년~2000년 6월 : 국방과학연구소 선임연구원
2000년 6월~현재 : 한국 전자통신 연구원 무선 LAN보안 연구팀 팀장
<관심분야> 무선 LAN보안, 무선 네트워크, Ad-Hoc 네트워크

조 기 환(Gihwan Cho) 정회원



1985년 2월 : 전남대학교 계산통계학과 졸업
1987년 2월 : 서울대학교 계산통계학과 석사
1996년 : 영국 Newcastle 대학교 전산학과 박사
1987년~1999년 :한국 전자통신 연구원 선임연구원
1997년~1999년 :목포대학교 컴퓨터과학과 전임강사
1999년~현재: 전북대학교 전자정보공학부 부교수
<관심분야> 이동 컴퓨팅, 컴퓨터 통신, 무선 네트워크 보안, 센서 네트워크, 분산처리 시스템