MANET에서 비정상 노드를 효율적으로 탐지하기 위한 보안 설계

정회원 황 윤 철*

Security Design for Efficient Detection of Misbehavior Node in MANET

Yoon-cheol Hwang* *Regular Member*

요 약

MANET(Mobile Ad hoc NETwork)은 고정된 네트워크 구조의 부재로 원거리 노드들 간의 통신은 다중 홉 경로를 통해 이루어지기 때문에 종단 노드 사이에 존재하는 노드들의 비정상적 행위를 탐지하고 예방하기가 어렵다. 그러므로 MANET의 성능과 보안 유지를 위해서는 비정상적 행위를 하는 중간노드들과 그에 오염된 노드를 찾아내기 위한 기법들이 필요하다. 그러나 기존에 제안된 기법들은 MANET를 구성하는 노드들이 우호적이며 상호 협력적인 관계라고 가정하고 비정상적 행위를 하는 노드를 식별하는 방법들만 제시해 왔고, 큰 규모의 MANET에 적용할 경우 많은 오버헤드가 발생한다. 따라서 이 논문에서는 MANET에서 구성요소간 안전한 통신을 제공하고 비정상 노드를 효율적으로 탐지 관리할 수 있는 Secure Cluster-based MANET(SecCBM)을 제안하였다. SecCBM은 동적인증을 통한 클러스터 기반 계층적 제어 구조를 이용하여 비정상 노드들을 MANET 구성 과정에서 식별하는 예방 단계와 네트워크를 구성하고 있는 노드들간 통신과정에서 발생하는 비정상 노드들을 FC 테이블과 MN 테이블을 이용해 탐지 관리하는 사후 단계로 구성하였다. 이를 통하여 MANET의 통신 안전성과 효율성을 향상시켰으며 시뮬레이션을 통한 성능평가에서 MANET에 적합한 기법임을 확인 할 수 있었다.

Key Words : MANET, Misbehavior Node, Clustering, Dynamic Authentication, Detection, Security

ABSTRACT

On a Mobile Ad hoc NETwork(MANET), it is difficult to detect and prevent misbehaviors nodes existing between end nodes, as communication between remote nodes is made through multiple hop routes due to lack of a fixed networked structure. Therefore, to maintain MANET's performance and security, a technique to identify misbehaving middle nodes and nodes that are compromise by such nodes is required. However, previously proposed techniques assumed that nodes comprising MANET are in a friendly and cooperative relationship, and suggested only methods to identify misbehaving nodes. When these methods are applied to a larger-scale MANET, large overhead is induced. As such, this paper suggests a system model called Secure Cluster-based MANET(SecCBM) to provide secure communication between components aperANET and to ensure eed. As sudh, this pand managems suapemisbehavior nodes. SecCBM consists apetwo stages. The first is the preventis pstage, whereemisbehavior nodes are identified when rANET is comprised by using a cluster-based hierarchical control structure through dynamic authentication. The second is the post-preventis pstage, whereemisbehavior nodes created during the course apecommunication amongst nodes comprising the network are dh, thed by using FC and MN tables. Through this, MANET's communication safety and efficiency were improved and the proposed method was confirmed to be suitable for MANET through simulation performance evaluation.

* 충주대학교 첨단과학기술대학 Post Doc(dolpin98@nate.com)
논문번호 : KICS2009-09-421, 접수일자 : 2009년 9월 19일, 최종논문접수일자 : 2010년 1월 14일

Ⅰ. 서 론

MANET은 유선 네트워크와 비교하여 전송속도가 낮고 신호간의 간섭이 상대적으로 심하며, 네트워크 형상이 수시로 변경된다[1],[2]. 또한, MANET를 구성하는 노드들이 유선 네트워크에 비해 소비할 수 있는 자원이 제한적이므로 기존 유선 네트워크에서 안전한 데이터 전송을 제공하기 위하여 제안된 기법들을 그대로 사용하는 것은 많은 문제가 있다[3].

또한, 다양한 환경에서 사용되는 MANET의 이동 노드들은 잦은 이동으로 인한 토폴로지 변동과 노드 상호간의 무선 링크로 인해 수동적인 공격과 능동적인 공격에 매우 취약하다. 그리고 이에 적용된 MANET의 보안 설계도 많은 취약점이 존재한다. 그래서 실제로 사용되는 대부분의 MANET은 이런 문제를 해결하기 위해 계층적 구조를 많이 사용하고 있다[3]-[5]. 그러나 대부분의 클러스터 기반의 MANET에서는 비정상 노드를 예방하기 적합한 보안 기법에 대한 언급이 미흡하고 협의된 노드들의 공격을 효율적으로 방지할 수 없다. 따라서 MANET에서 이동 노드 사이에 높은 전송효율의 통신 제공과 네트워크 자원의 효율적인 사용, 그리고 네트워크 보안을 높이기 위한 기술이 절실히 요구된다.

그리고, MANET에서는 출발 노드와 목적지 노드 사이에 있는 중간 노드의 비정상 행위는 안전한 통신을 저해하는 주요인이 된다. 그러므로 MANET에서 네트워크의 안전성과 보안성을 강화하기 위하여 구성요소들 중 비정상 행위를 하는 노드에 대하여 적은 오버헤드로 빠르게 식별하여 관리할 수 있는 효율적인 메커니즘이 필요하다. 따라서, 본 논문에서는 MANET에서 비정상 노드의 효율적인 탐지 및 관리를 위해 SecCBM을 제안한다. SecCBM은 동적인증을 통한 클러스터 기반 계층적 제어 구조를 이용하여 비정상 노드들을 MANET 구성 과정에서 식별하여 신뢰성 있는 네트워크를 형성하는 Proactive Phase와 노드들간 통신과정에서 발생하는 비정상 노드들을 FC 테이블과 MN 테이블을 이용하여 탐지 하고 관리하는 Reactive Phase로 구성하였다.

표 1. 클러스터링 기법들의 특징과 단점

구 분	방 식	특 징	단 점
노드의 식별자 기반	최고 ID 최저 ID Max-Min D-Cluster	-식별자로 클러스터 헤드 선정 -알고리즘 간단 -계산량이 적음	-노드간 균등한 부하 배분이 어려움 -클러스터 헤드의 과다한 자원 소모 -잦은 재클러스터링
노드의 연결성 기반	최고 연결도 기반	-같은 전송 범위안에서 연결도가 높은 노드가 클러스터헤드로 선정 -클러스터의 업데이트 횟수가 적음	-클러스터내의 노드 수가 제한되지 않아 과부하 발생 -클러스터 헤드의 과 다한 자원 소모 -클러스터간 균등한 부하 배분이 어려움
노드의 가중치 기반	WCA WBACA	-노드에게 가중치를 주어 클러스터 헤드로 선정 -다른 기법에 비해 업데이트 횟수가 적음	-시스템의 성능을 고려한 최적화된 방법이지만 다른 방식에 비해 계산량이 많음

본 논문은 다음과 같이 구성된다. Ⅱ장에서는 기존의 관련연구를 정리한 것으로, MANET에서의 클러스터링 알고리즘과 비정상 노드 탐지 기법에 대하여 기술한다. Ⅲ장에서는 본 논문에서 제안한 MANET에 적합한 비정상 노드 탐지 모델인 SecCBM을 제안하고, 구성 요소들간 동적 인증을 통한 신뢰성 있는 클러스터 기반 MANET을 구축하는 Proactive Phase에 대하여 기술한다. Ⅳ장에서는 구축된 클러스터 기반 MANET에서 데이터 전송과정에 발생하는 비정상 노드를 탐지 관리하는 Reactive Phase에 대하여 기술한다. Ⅴ장에서는 시뮬레이션을 통해 제안된 기법들의 성능을 측정하여 비교 분석한 결과를 기술하고, 마지막으로 Ⅵ장에서는 본 연구의 결론과 향후 연구 과제를 기술한다.

Ⅱ. 관련 연구

MANET에서 클러스터링을 통해 노드를 계층적인 구조로 관리 하는 방식 중 대표적이고 가장 많이 사용되는 방식은 노드의 ID를 이용하는 M. Gerla가 제안한 최저 식별자 기반 방식[6]과 노드의 밀도를 이용하여 단일 홉으로 구성된 계층적 구조를 만드는 방식인 최고 연결도 기반 방식이다[7]. 이러한 방식들은 가장 낮은 식별자 혹은 가장 높은 연결도를 가진 노드가 클러스터 헤드가 되며 클러스터 헤드로부터 단일 홉에 있는 노드들로 클러스터가 형성된다. 이 방식들은 토폴로지 변화가 많아질수록 클러스터를 재구성해야 하고 네트워크의 크기가 증가 할수록 클러스터의 수와 관리를 위한 트래픽 양이 증가되므로 전체적인 네트워크의 성능과 수명이 저하되는 문제점을 보인다. 이런 문제들을 해결하기 위해 [4], [8], [9]에서는 클러스터 헤드의 역할 분담을 위해 주기적인 ID 변경이나 가중치를 이용한 방법 등의 여러 가지 방안들이 제안되었다. 그러나 부하 분산을 위한 적절한 변경 주기를 선정하는 문제는 네트워크의 성능과의 관계를 고려할

경우 상호 Trade-Off관계에 있다[9],[10]. 각 클러스터링 기법들의 특징과 단점을 정리하면 표 1과 같다.

MANET에서 비정상 노드 탐지의 대표적인 방식은 Watchdog and Pathrater[11], Byzantine Fault Detection[12], 그리고 증명서를 이용한 이기적인 노드 관리 방법(Secure Mechanism to Manage Selfish Nodes)[13]를 들 수 있다. Watchdog and Pathrater 방식은 비정상 노드를 탐지하기 위해 경비견(Watchdog)과 경로 안내자(Pathrater)를 이용한 메커니즘으로 비정상 노드를 탐지하기 위해 네트워크의 모든 노드들이 자신의 주변에 있는 노드들을 서로 감시하는 방식을 사용한다.

Byzantine Fault Detection 방식은 경로의 길이가 'n'이라고 할 때, 적응적 조사 기법을 사용하여 log n개의 결함(faulty)이 발생하면 faulty 링크를 찾아낼 수 있는 기법을 이용하여 비정상 노드를 식별한다. 증명서를 이용한 이기적인 노드 관리 방법에서는 네트워크 내에 있는 이기적 노드를 탐지하는데 있어 거짓 신고하는 악의적인 노드가 있을 경우를 고려하여, 증명서를 이용해서 신고를 하도록 하는 알고리즘을 사용한다. 이 방식들은 비정상적인 노드를 식별해 내기 위한 알고리즘이지만 정상적인 노드를 악의적인 노드라고 거짓으로 신고하는 악의적인 노드에 대해서는 적절히 대응하지 못하는 문제점이 있다. 이러한 악의적인 노드에 의한 거짓 신고로 인하여 정상적인 노드를 네트워크로부터 분리시킴으로써 각각의 이동 노드들의 협력적인 동작에 의해 데이터 전달이 이루어지는 MANET의 처리율을 저하시키고 더불어 악의적인 노드를 정상적인 노드라고 잘못 판단하게 되어 네트워크의 안전성을 떨어뜨린다. 또한, 악의적인 노드는 네트워크에 계속 남아 있으면서 자신에게 오는 data를 전달하지 않고 버리는 행위와 다른 노드를 거짓 신고함으로써 점점 더 많은 노드들을 네트워크에 참여하지 못하게 만들어서 네트워크의 성능을 심각하게 저하시킨다. 따라서 이와 같이 거짓 신고하는 악의적인 노드를 찾아내어 네트워크로부터 분리시키는 것이 MANET의 성능과 안전성을 보장하는데 있어 필수적이다. 기존에 제안되었던 비정상 노드 탐지기법들의 특징과 장단점을 분석해 보면 표 2와 같다.

표 2. 비정상 노드 탐지 기법들의 특징과 장단점

기법	탐지유형	특징 및 장점	단점
Watchdog & Pathrater	이기적 악의적	-잘못된 라우팅 탐지 -메시지 전송으로 인한 오버헤드가 없음 -오버헤드가 적고 경로 설정자를 통해 믿을 만한 경로를 선택하여 전송	-Route Request메시지 전송 오버헤드가 큼 -노드에 대한 단독적인 판단 -False Positive가 많음
Byzantine Fault Detection	악의적	-적은 조사 기법으로 fault가 발생하면 fault 링크를 찾아냄 -정해진 임계치를 초과하면 중간에 있는 노드들 중 악의적인 노드 검출 -소스와 probe 사이에 on-demand로 키 교환 프로토콜을 이용	-악의적인 노드를 찾기 위한 메시지 전송에 따른 큰 오버헤드 -협의된 노드간의 협업으로 부정확한 판단이 이루어짐
증명서를 이용한 방법	악의적	-데이터 전송 과정에서 증명서를 이용해 데이터 전송을 감시 -거짓으로 신고하는 악의적인 노드의 검출이 가능	-증명서를 발행하는 시간 확인하는 시간, 그리고 저장 공간이 필요 -협의된 노드간의 협업으로 부정확한 판단이 이루어짐

Ⅲ. MANET에 적합한 비정상 노드 탐지 모델

이 장에서는 MANET의 보안 취약점을 보완하고 노드들 사이에 안전한 통신을 보장하기 위한 시스템 모델인 Secure Cluster-based MANET(이하 SecCBM)를 제안한다. SecCBM은 크게 비정상 노드를 예방하는 단계인 Proactive Phase와 비정상 노드가 발생한 후에 사용하는 조치 단계인 Reactive Phase로 구성되며, 이 장에서는 시스템 모델과 Proactive Phase에 대하여 기술하고 사후 단계는 4장에서 다룬다.

3.1 SecCBM

MANET에서 클러스터를 이용하여 구성요소 사이에 안전한 통신을 제공하고 비정상 노드를 효율적으로 탐지 관리하기 위하여 그림 1과 같은 SecCBM를 제안한다. SecCBM는 Proactive Phase와 Reactive Phase로 구성된다.

Proactive Phases는 Cluster Formation과 Dynamic Authentication으로 구성되며, Cluster Setup은 MANET의 취약점인 기반시설의 부재를 해결하기 위하여 MANET를 구성하는 이동 노드들을 클러스터 단위로 나누기 위하여 사용한다. 구분된 클러스터들은 각 클러스터에서 선출된 클러스터 헤드에 의해 관리된다. Cluster Setup은 크게 클러스터 형성과정과

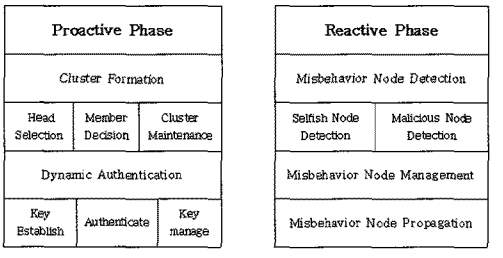


그림 1. SecCBM 구조

클러스터 관리과정으로 나누어지며 클러스터 형성과정은 클러스터 헤드선출과 멤버노드의 결정으로 이루어진다. 이 단계에서는 WCA[8]을 확장하여 클러스터가 형성된 후에 일정시간동안 클러스터를 안정적으로 유지할 수 있도록 재설계하였다[14].

Dynamic Authentication은 MANET을 구성하는 노드들간 무선 링크의 사용과 다중 홉 방식의 통신으로 인하여 통신하는 상대방을 신뢰할 수 없는 보안 취약점을 해결하기 위하여 사용되었다. 위의 두 모듈은 비정상 노드를 탐지하기 위한 기본 보안 제어구조를 형성하기 위해 사용 된다.

Reactive Phase는 Misbehavior Node Detection과 Management, Propagation으로 구성되며, 계층적 구조로 형성된 MANET에서 데이터 전송 과정에서 발생되는 비정상 노드를 탐지하고 관리하기 위해 사용된다. Reactive Phase을 통하여 데이터 전송과정 중에 MANET에 존재하는 비정상 노드를 빠르게 식별하여 대응함으로써 MANET의 안전과성능 저하를 최대한 억제되도록 하였다.

3.2 Proactive Phases

3.2.1 Cluster Formation

이 논문에서는 기존 클러스터링 알고리즘의 문제점인 최적의 클러스터 헤드 선정 문제와 재클러스터링을 해결하기 위해서 기존의 가중치 기반 클러스터링 알고리즘을 보완한 가중치 기반 분산 클러스터링 알고리즘(WBDCA)을 사용하였다[14]. 이 알고리즘은 클러스터 형성단계에서는 초기 클러스터 구성할 때 발생되는 오버헤드를 최소화하기 위해 지역적으로 클러스터를 구성하는 방식을 사용한다. 또한, 현재의 클러스터 구조를 가능한 유지하여 과다한 연산을 피하기 위해 부클러스터 헤드와 분산 게이트웨이라는 개념을 정의하여 클러스터 관리과정에서 사용한다. WBDCA는 노드의 연결성 차이, 노드들간의 거리의 합, 평균 이동 속도, 노드에 남아있는 배터리 파워 등을 기반으로 가중치를 산출한다. 이러한 가중치 산출 터리들 중 노드에 남아 있는 배터리 파워를 제외하고는 WCA에서 사용된 가중치와 동일하게 사용한다. WCA에서는 배터리 파워를 노드가 클러스터 헤드로 동작하는 동안의 누적 시간으로 산출하지워하고는하는 WBDCA에서는 실제로 노드에 남아 있는 배터리 양으로 산출함으로써 전력 소모에 따른 배터리 전력을 보다한 가중치 산관리한다. 또한 노드의 이동으로 인하여 발생되는 클러스터 재설정 오버헤드를 줄이기 위해 부클러스터 헤드와 분산 게이트웨이 노드 개념을 이용하여 시스템의 현재 구조를 최대한 유지할 수 있게 함으로써 시스템의 안정적인 관리를 제공한다. 알고리즘은 초기화 과정을 거쳐 클러스터 형성과정 그리고 클러스터 관리과정으로 이루어진다. 그림 2는 클러스터 형성 알고리즘이다.

클러스터링이 완료되면, 클러스터 멤버 노드들은 주기적으로 클러스터 헤드에게 HELLO 메시지를 전송하여 토폴로지를 관리한다. 그러나 해당 클러스터에 속한 노드가 이동하거나 클러스터 헤드가 역할을 포기했을 때는 토폴로지의 변화가 일어난다. 이런 경우에 변경된 클러스터에 대한 업데이트를 하기 위해 변경된 클러스터들에 대하여 클러스터링 관리과정이 수행된다. 첫 번째는 해당 클러스터의 헤드가 과다한 배터리 낭비로 인해 자신의 역할을 포기할 때 클러스터를 관리하는 방법이다. 매번 클러스터 헤드가 자신의 역할을 포기할 때마다 클러스터 헤드를 재선출하게 되면 그에 따른 네트워크 프로세싱 오버헤드와 불안정한 토폴로지, 자원할당 및 데이터의 손실을 초래할 수 있다. 따라서 이 논문에서는 클러스터 헤드가 선출된 다음에 클러스터 헤드가 자신의 단일 홉 내 이웃노드 중 가중치 값이 자신보다 큰 노드들 중 가장 작은 노드를 부클러스터 헤드로 선출하는 방법을 사용했다. 이를 통해 클러스터 헤드에 일정한 임계치를 적용하여 그 임계치 이상일 때만 클러스터 헤드의 역할을 수행하게 하고 임계치에 도달하면 부클러스터 헤드가

```
Initialize
Begin Setup
  MY_ch = unknown;
  {   if(W_MY_id = MIN(W_my_Z))
    {  MY_ch = MY_id;
       Broadcast Cluster(MY_id, MY_ch);
       Z = Z - {(MY_id, W_MY_id)};
    }
    while(Z!=∅)
    { on receving cluster(MY_id, W_MY_id);
      if((YOU_id⊂T_MY_ch) &&
                  (YOU_ch==unknown||W_YOU_id>W_MY_id))
      {  YOU_ch = MY_ch;
         Z = Z - {(YOU_id, W_YOU_id)};
      }
      if((W_YOU_id = MIN(W_YOU_Z))
       {     if(YOU_ch==unknown)
                 YOU_ch = YOU_id;
             Broadcast Cluster(YOU_id, YOU_ch);
             Z = Z - {(YOU_id, W_YOU_id)};
        }
    }
  }
End Setup
```

그림 2. 클러스터 형성 알고리즘

클러스터 헤드의 임무를 대행하게 함으로써 클러스터 헤드를 재선출 과정에서 발생하는 네트워크 프로세싱 오버헤드 및 선출지연, 자원 할당 및 데이터의 손실을 최소화한다. 두 번째는 클러스터 내의 해당 노드들의 이동으로 인해 클러스터 재구성 문제가 발생했을 때 클러스터 관리 방법이다. 매번 클러스터내의 해당 노드들 중의 일부가 해당 클러스터 밖으로 이동 할 때마다 클러스터를 재구성하면 네트워크에 잦은 오버헤드와 정보전송의 지연이 발생하게 된다. 따라서 이 논문에서는 분산 게이트웨이라는 역할을 정의하여 클러스터내의 해당 노드가 클러스터 밖으로 이동할 때 그 이동 노드를 관리하게 함으로써 클러스터 재구성이 발생하지 않도록 했다. 그림 3는 전체 클러스터를 유지 관리하는 알고리즘이다.

3.2.2 Dynamic Authentication

클러스터로 형성된 MANET에서 효율적인 인증을 수행하는 동적 인증 기법(Cluster based Dynamic Authentication : CBDA)을 사용한다[19],[20]. CBDA는 신뢰성 있는 통신을 위해 그림 4와 같은 여러 독립적인 클러스터로 구성된 네트워크 모델을 사용한다.

High layer에는 전체 클러스터를 관리하는 클러스터 관리자(CM)가 위치하며, 초기 클러스터 형성 과정에서 필요한 키 분배 및 키 관리 기능을 수행

```
Begin Maintenance
if(cluster-head==resign)
{ cluster-head = pre_cluster-head;
  CHid=pro_cluster-headid;
  Broadcast Cluster(pre_cluster-headid, CHid)
  on receive cluster member    // Z configure
  {   MYch = CHid;
      if(one's role change)
                role change;
      else  role maintenance;
      send to cluster-head one's weight;
   }
  Zcluster-head = { Zcluster-head -(cluster-headid, Wcluster-head_id)}
  pre_cluster-head = min (WZ)
}
if(node==walk out one's cluster){
 if(able to one's cluster-head || distribute gateway
connect)
    connect maintenance;
 else if(able to other distribute gateway)
          connect other distribute gateway;
      else if(other cluster-head)
              one's cluster-headid = other cluster-headid;
           else
               one's cluster-headid = one's cluster-headid;
}
End Maintenance
```

그림 3. 클러스터 유지 관리 알고리즘

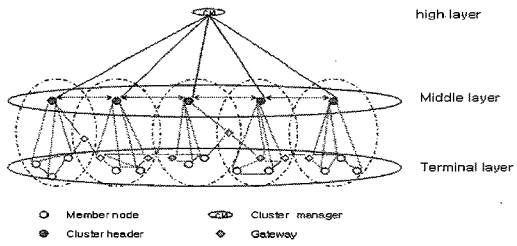


그림 4. 클러스터 기반 동적 인증 모델

한다. Middle layer에는 각각의 클러스터를 관리하는 클러스터 헤드들로 구성되며, 해당 클러스터내 멤버 노드들을 관리한다. Terminal layer에는 계층구조의 최하위 노드로 클러스터에 속하는 멤버 노드로 구성된다. CBDA의 안전성은 [15]의 Diffie-Hellman 키 분배 방식을 이용하여 이산대수의 어려움에 근거하며 노드들의 에너지 소모와 네트워크 오버헤드가 최대한 적게 발생되도록 설계하였다.

CBDA는 키 생성 및 분배, 클러스터 구성 요소 사이의 키 설립, 동적 인증, 그리고 키 관리 4단계로 구성되며, 1단계에서 비밀 분산 기법을 이용하여 부분 비밀키를 생성하고 분배한다. 2단계는 클러스터를 구성하는 노드들 사이의 키 설립이 이루어지고, 3단계에서 생성된 키들을 사용하여 클러스터 구성요소들 사이에 동적 인증이 수행된다. 4단계에서 클러스터 유지과정에서의 키 관리가 진행된다.

네트워크 내에서 인증 과정이 시작되면 클러스터 관리자가 선출되고, 클러스터 관리자는 클러스터 헤드에게 키 분배를 한다. 이 과정에서 클러스터 관리자와 클러스터 헤드 사이 그리고 다른 클러스터 헤드 사이의 인증이 그림 5와 그림 6과 같이 수행된다. 이들 사이의 관계가 성공적으로 증명되면 클러스터 헤드와 멤버 노드사이에 등록 절차가 시작되

cluster manager, cluster header, member node generate public/private key

SEND CM	Broadcast all node in cluster	
Begin	Authentication between cluster manager and cluster-head	
SEND	$CH_i \rightarrow CM$	$RV_1 \| id_{ch_i} \| GK_{ch_i} \| P_{ch_i} \| \{id_{ch_i}\} S_{ch_i} \} p_{cm}$
RECEIVE	CM	STORE id_{ch_i}, P_{ch_i}, GK_{ch_i} of CH_i
SEND	$CM \rightarrow CH_i$	$\{RV_1 \| \{k_{cm-ch_j} \| id_{cm}\} s_{cm}\} p_{ch_i}$
RECEIVE	CH_i	if (send RV_1==receive RV_1)
		{
		STORE $k_{cm-ch_j} id_{ch_i}$
		}
		else fail authentication
END		

그림 5. 클러스터 관리자와 클러스터 헤드 사이의 인증 메커니즘

Begin Authentication between cluster-head and cluster-head

SEND $CH_i \rightarrow CM$ $\{id_{ch_i} \| id_{ch_j} \| RV_2 \| \{id_{ch_i}\}s_{ch_i} \| \{RV_2 \| \{id_{ch_i}\}s_{ch_i} \| id_{ch_j}\}p_{ch_j}\}K_{CM-CH_i}$

RECEIVE CM check share key

SEND $CM \rightarrow CH_j$ $\{id_{cm} \| id_{ch_j} \| RV_3 \| \{id_{cm}\}s_{cm} \| p_{ch_i} \| \{RV_2 \| \{id_{ch_i}\}s_{ch_i} \| id_{ch_j}\}p_{ch_j}\}K_{CM-CH_i}$

RECEIVE CH_j check share key

STORE id_{ch_i}, P_{ch_i} of CH_i

SEND $CH_j \rightarrow CM$ $\{id_{ch_j} \| id_{cm} \| RV_3 \| \{id_{ch_j}\}s_{ch_j} \| \{RV_4 \| id_{ch_i} \| \{id_{ch_j}\}s_{ch_j}\}p_{ch_j}\}K_{CM-CH_i}$

RECEIVE CM if (send RV_3==receive RV_3)

SEND CH_i

$\{id_{cm} \| id_{ch_i} \| RV_2 \| \{id_{cm}\}s_{cm} \| p_{ch_j} \| \{RV_4 \| id_{ch_i} \| \{id_{ch_j}\}s_{ch_j}\}p_{ch_j}\}K_{CM-CH_i}$

else deny

RECEIVE CH_i if (send RV_2==receive RV_2)

{ STORE id_{ch_j}, P_{ch_j} of CH_j

SEND CH_j

$\{id_{ch_i} \| id_{ch_j} \| \{id_{ch_i}\}s_{ch_i} \| \{id_{ch_j} \| K_{ch_i - ch_j}\} \| RV_4 \| RV_5\}p_{ch_j}$)

else deny

RECEIVE CH_j if (send RV_4==receive RV_4)

SEND CH_i $\{id_{ch_i} \| id_{ch_j} \| \{id_{ch_i}\}s_{ch_j} \| RV_5\}K_{CH_i-CH_j}$

else deny

RECEIVE CH_j if (send RV_5==receive RV_5)

authention success

else authentication fail

END Authentication between cluster-head and cluster-head

그림 6. 클러스터 헤드 사이의 인증 메커니즘

고, 멤버 노드의 공개키 등록절차가 끝나면 통신하고자 하는 다른 노드와의 인증 절차가 그림 7과 같이 진행된 후 통신이 시작된다.

Begin Authentication between cluster-head and member node

SEND $CH_i \rightarrow M$ request member node public key $\{id_{ch_i} \| \{id_{ch_i}\}s_{ch_i} \| p_{ch_i} \| RV_6\}$

RECEIVE M STORE id_{ch_i}, P_{ch_i} of CH_i

SEND $M \rightarrow CH_i$ $\{id_m \| id_{ch_j} \| p_m \| RV_6 \| RV_7 \| \{id_m\}s_m\}p_{ch_i}$

RECEIVE CH_i if (send RV_6==receive RV_6)

{

STORE id_m, P_m of M

SEND M $\{id_{ch_j} \| \{id_{ch_i}\}s_{ch_i} \| GK_{ch_i} \| sm_M \| RV_7 \| RV_8\}p_M$

}

else deny

RECEIVE M if (send RV_7==receive RV_7)

{

validation sm_m

STORE GK_{ch_i} of CH_i

}

else fail authentication

END

그림 7. 클러스터 헤드와 멤버 사이의 인증 메커니즘

MANET의 특성상 토폴로지의 변화가 빈번히 발생하기때문에 형성된 클러스터를 유지하는 과정에서 노드의 진출입, 클러스터 관리자의 부재, 클러스터 헤드의 양도 및 부재 등으로 인한 토폴로지 변화에 적절히 대처하기 위한 키 관리 방법이 필요하다.

클러스터 관리자가 자신의 역할을 수행하는데 있어 예상치 못한 상황이 발생하거나 자원 고갈로 인해 네트워크에서 사라질 수 있다. 이런 경우 클러스터 헤드는 주기적인 갱신 메시지를 클러스터 관리자로부터 받지 못하기 때문에 클러스터 관리자의 부재를 인지할 수 있다. 재클러스터링을 통해 클러스터 관리자를 다시 선출하고 선출된 클러스터 관리자는 키 쌍을 생성하여 이를 네트워크에 알린 후 아래와 같은 과정을 수행하여 새로운 클러스터 기반 MANET을 구성한다.

step 1 : Select CM through Clustering process
step 2 : Establishment Key between Cluster configure elements
step 3 : Performance Cluster based Dynamic Authentication

클러스터 헤드가 자신의 역할을 포기하거나 자원의 한계치에 도달할 경우, 그리고 자원 고갈 문제로 인해 예기치 못한 사항에서 네트워크에서 갑자기 사라질 수 있다. 이런 상황이 전개되면 부클러스터 헤드가 클러스터 헤드의 역할을 양도 받는다. 그런 다음 해당 클러스터에 대하여 재클러스터링을 수행하여 새로운 부클러스터 헤드를 임명한다. 새로이 선출된 클러스터 헤드는 앞 절에서 기술한 키 분배 절차를 반드시 거쳐 그룹키를 클러스터 관리자에게 등록한다. 그리고 클러스터 안에 있는 노드들의 공개키를 획득하기 위해 클러스터 멤버 노드와 인증 절차를 갖는다.

step 1 : Pre_cluster-head → cluster-head
step 2 : Clustering process
step 3 : Establishment Share Key between cluster manger and cluster-head
step 4 : Authentication between cluster manger and cluster-head
step 5 : Establishment Share Key between cluster-head and member node
step 6 : Authentication between cluster-head and member

MANET를 구성하는 노드는 이동 노드이므로 각

노드는 해당 클러스터를 벗어나거나 이웃 클러스터로 진입할 수 있다. 일반적으로 클러스터 내의 노드들은 주기적으로 클러스터 헤드가 보내는 inform 메시지를 통해서 자신이 클러스터 내에 있다는 것을 인식한다. 만약에 이 메시지를 받지 못하는 경우에는 자신이 클러스터 범위를 벗어났거나 혹은 클러스터 헤드가 그 기능을 하지 못한다고 인식하여 Search Cluster Request 메시지를 이용하여 자신이 포함된 클러스터 범위를 확인한다. 이 때 자신이 클러스터 안에 있음에도 불구하고 해당 클러스터의 클러스터 헤드로부터 inform 메시지를 받지 못했다면 이는 클러스터 헤드가 기능을 하지 못한다고 판단하여 새로운 클러스터 헤드를 선출한다. 이후 키 분배 과정을 수행하고 클러스터 내 노드들의 공개키를 획득하기 위해 클러스터 헤드와 인증 절차를 갖는다. 반대로 inform 메시지를 받았는데 이 inform 메시지 안의 클러스터 헤드의 ID가 다른 클러스터의 클러스터 헤드의 ID라면, 새로운 클러스터 헤드와 인증 절차를 갖은 후 자신의 공개키를 등록하고 새로운 그룹키를 부여 받아 통신을 개시한다. 노드가 진입된 클러스터의 클러스터 헤드는 새로운 그룹키를 생성하여 해당 클러스터 멤버들에게 전송하고, 새로운 노드가 전에 속해 있었던 클러스터 헤드에게 해당 노드의 이동 사실을 알린다. 이동 사실을 인지한 이전의 클러스터 헤드는 해당 노드의 공개키를 삭제하고 자신의 그룹키을 갱신하여 자신의 클러스터 멤버 노드들에게 전송한다. 이는 노드의 진출입으로 인해 오염될 수 있는 그룹키의 무결성과 기밀성을 보장한다.

step 1 : $M_i \rightarrow CH_i$: $\{id_{m_i} \| id_{ch_i} \| p_{m_i} \| RV_1 \| CH_j \| \{id_{m_i}\}s_{m_i}\}_{p_{ch_i}}$

step 2 : Generate GK_{ch_i} and Notice CM, existence member node

step 3 : $CH_i \rightarrow M_i$: $\{id_{ch_i} \| id_m \| RV_5 \| \{id_{ch_i}\}s_{ch_i} \| GK_{ch_i} \| sm_{m_i}\}_{p_{m_i}}$

setp 4 : Authentication between cluster-head and member

step 5 : $CH_i \rightarrow CH_j$: Register M_i

step 6 : CH_j : Remove M_i

Ⅳ. 데이터 전송과정에서의 비정상 노드 탐지

이 장에서는 SecMANET의 비정상 노드 탐지 모듈을 이용하여 클러스터 기반 MANET에서 데이터를 전달하는 과정에 발생하는 비정상 노드를 효율적으로 탐지하고 관리하는 기법을 제안하고 탐지과정을 기술한다.

4.1 비정상노드 탐지 모델

MANET를 구성하는 요소들 중에는 자신의 이익을 최우선적으로 생각하는 이기적 노드(Selfish Node)와 네트워크를 마비시키거나 훼손시킬 목적으로 정상적인 패킷의 흐름을 방해하거나 파괴하고, 잘못된 정보를 생성하여 유포함으로써 적법한 노드로의 경로를 우회시켜 패킷의 전송을 지연시키거나 최적의 경로를 찾는 것을 방해하는 행위를 하는 노드와 이에 타협된 노드들인 악의적 노드(Malicious Node)도 존재한다. 이 논문에서는 이런 노드들을 비정상 노드(Misbehavior Node)라고 한다.

이런 비정상적인 노드들은 MANET에서 라우팅 프로토콜 패킷의 흐름, 즉 경로 확보 메시지의 전파와 갱신을 저하시키거나 왜곡시키고, 라우팅 프로토콜 트래픽과 제어 메시지의 파괴, 잘못된 라우팅 정보를 생성, 전파함으로써 적법한 노드로의 경로를 우회시킨다. 또한 라우팅 정보를 누설함으로써 정보의 충돌을 유도하여 라우팅을 혼란시켜 패킷을 지연시키거나 방해함으로써 패킷이 목적지 노드에 도달할 수 없도록 하는 역할을 하며 결과적으로 네트워크 성능을 저하시키거나 분해, 또는 마비시킨다.

이 논문에서 제안하는 비정상적 노드 탐지모듈의 동작은 탐지, 식별, 대응과 같이 3단계로 이루어지며 식별 단계는 비정상 노드 판별부분과 비정상 노드 관리부분으로 나누어진다. 비정상 노드의 탐지 모델은 그림 8과 같다.

비정상 노드 탐지과정은 멤버노드의 포워딩한 패킷수를 보고하는 메시지와 신고 메시지로 시작되며 보고 메시지는 클러스터 관리자나 클러스터 헤드에 의해 보고서를 보낸 노드가 이기적인 노드인지 악

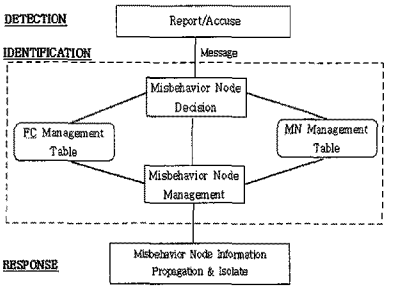


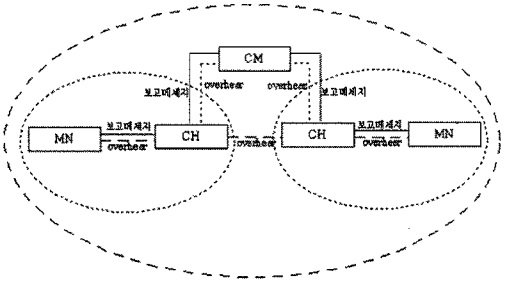
그림 8. 비정상 노드 탐지 모델

의적인 노드인지 판별한다. 판별된 노드는 비정상 노드 관리부분에서 MN 테이블로 관리한다. 협의된 노드를 고려한 악의적 노드 탐지는 데이터 전송과정에서 신고 노드의 신고 메시지에 의해 탐지되며 신고 메시지는 데이터를 전송한 소스노드에 의해 신뢰성 여부를 판단하여 악의적인 노드가 어느 노드인지 판별한다. 판별된 노드는 MN 테이블로 관리되며 최종적으로 악의적인 노드라고 판별된 노드의 정보를 네트워크의 모든 노드들에게 전파하여 악의적 노드를 격리시킨다. 그리고 이기적인 노드는 FC라는 개념을 사용하여 다른 노드들을 위해 패킷을 전달해 주는 노드에게는 일종의 보상을 해 주고 다른 노드에게 과중한 짐을 지우려는 노드에게는 약간의 불이익을 주는 방안으로 이기적인 노드가 생성되는 것을 억제한다. 이 방식은 자신이 생성한 패킷만을 보내는 노드에게는 신뢰도를 일정한 가중치만큼 감소시키고 다른 노드가 생성한 패킷을 자신을 경유하여 다른 노드에게 전달해 주는 경우에는 신뢰도를 증가시켜주는 방식을 사용한다.

4.2 비정상노드 탐지 기법

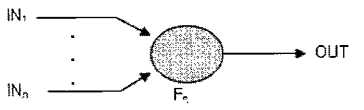
이 논문에서는 앞장에서 제안한 WBDCA 을 사용하여 형성된 계층적 MANET 구조를 이용하여 비정상 노드를 관리한다. 클러스터를 형성한 후의 비정상 노드 탐지 구조(Cluster-based Misbehavior Node Detection Architecture:이하 CMNDA)는 그림 9와 같다. 최상위 노드인 클러스터 관리자는 중간 노드인 클러스터 헤드들의 신뢰도를 관리하고 각각의 클러스터 헤드들은 자신이 관리하는 멤버들의 신뢰도 값을 각각 분산 관리한다. 그리고 각 노드에는 그림 10과 같은 구조로 이루어진 FC를 설치한다.

CMNDA를 사용하는 MANET에서는 한 노드가



CM: Cluster Manger CH: Cluster-Head MN: Member Node

그림 9. 비정상 노드 탐지 구조



Input = IN_1 + ... + IN_n OUT = Input − T_{Pn} + S_{Pn}

F_n = Input − T_{Pn}

Fn : number of Forwarding packet

T_{Pn} : number of Destination packet

S_{Pn}: number of Start packet

그림 10. Forwarding Counter의 구조

자신이 생성한 패킷을 송신하기 위해서는 다른 노드들에게 패킷을 포워딩함으로써 얻게 되는 신뢰도가 한계치 이상이어야 한다. 클러스터 관리자나 클러스터 헤드 노드는 자신이 관리해야 하는 노드들의 신뢰도를 관리하기 위해 그림 11과 같은 FC 관리 테이블을 생성하여 유지한다. 그리고 클러스터 헤드나 각 클러스터 멤버 노드들은 일정 주기 간격으로 자신이 포워딩한 패킷수를 자신의 신뢰도를 관리하는 클러스터 관리자나 클러스터 헤드에게 보고한다. 보고를 받은 클러스터 관리자나 클러스터 헤드는 이들 정보가 신뢰할 수 있는지 검증을 하고 보고 메시지의 정보를 이용하여 FC 관리 테이블의 내용을 변경한다. 그림 11의 FC 관리 테이블에 있는 보고 노드의 ID는 클러스터 관리자나 클러스터 헤드에게 보고 메시지를 보낸 노드의 ID이며 신뢰도는 보고 메시지의 정보들을 가지고 식 1에 의해 생성된 값에 일정한 가중치를 붙인 값이다.

가중치 값은 [11]에서 사용한 등급값을 준용하며 MANET에 참여하는 모든 노드의 신뢰도 초기값은 1이며 아무 일도 하지 않으면 신뢰도는 변화가 없고 다른 노드에게 패킷을 전달하는 중간 노드 역할을 할 경우 신뢰도를 높여준다. 경때 신뢰도는 기존의 값에 보고 메시지에 포함된 정보를 통해 얻뢰도 n에 보상 가중치 0.01을 곱한 값만큼 증가시켜준 얻뢰도 불일치 노드의 ID는 신뢰도SN을 가진 메시지 중중간고 보상패킷을 도 하는 도 서하는 도 뉴하는치하지 않보상노드의 ID이 그 횟수를 불일치 횟수에 기록 않으면 신뢰도거짓 보고 횟수는 보고 메시지를 허위로 보고한 보고 노드에 부여하는

보고 노드의 ID	신뢰도	보고 불일치 노드의 ID	불일치 횟수	거짓 보고 횟수

그림 11. FC 관리 테이블

카운터 값이다.

임의의 한 노드에서 이웃한 다른 노드로 패킷을 전송하면 해당 노드들을 관리하는 상위 노드에게 이 사실을 알리는 보고 메시지를 보낸다. 이 때 통신 쌍방 간에 보내온 메시지를 받은 상위 노드는 두 보고 메시지의 내용을 검증하여 일치 여부를 확인한다. 확인 결과 불일치 내용이 있을 때 거짓 보고 횟수가 증가하고 불일치 횟수도 증가된다. 그리고 거짓 정보를 포함하고 있는 보고 메시지를 보내온 노드의 신뢰도는 0.1만큼 차감한다. 한 노드의 신뢰도가 임계치 이하로 떨어지면 이 노드를 이기적 노드로 판별하여 이 노드에서 보내온 패킷을 다른 이웃 노드에서는 무시한다. 그러나 다른 노드에서 이 노드로의 패킷 전송은 허용함으로써 자발적으로 패킷 전달에 참여하도록 유도한다. 클러스터 헤드나 각 클러스터 멤버 노드들이 클러스터 관리자나 클러스터 헤드에게 보고하는 보고 메시지 형식은 그림 12와 같다.

보고 메시지의 각 필드의 의미는 다음과 같다.

- SID : 보고하는 노드의 ID
- NID : 패킷을 전달받은 이웃노드의 ID
- SN : 보고 메시지의 동기화를 위한 순서 번호
- I_{pn} : 이웃노드로부터 n으로 전송된 패킷 수
- O_{pn} : n으로부터 이웃노드로 전송한 패킷 수
- S_{pn} : 출력된 패킷 중 n이 생성하여 전송한 패킷 수
- T_{pn} : 입력된 패킷 중 n이 최종 목적지인 패킷 수
- ONS : 자신이 출력한 패킷중 전달받을 이웃노드가 생성하여 전송한 패킷 수

클러스터 매니저와 클러스터 헤드는 자신에게 전송된 보고 메시지를 통하여 보고한 노드의 포워딩한 개수를 식 1과 같이 산출한다.

$$F_n = \sum_{n \in N_n} I_{pn} - \sum_{n \in N_n} T_{pn} = \sum_{n \in N_n} O_{pn} - \sum_{n \in N_n} S_{pn} \quad (1)$$

F_n : 일정 주기 동안 노드 n이 포워딩한 개수
N_n : n 노드의 이웃노드의 집합

이 식은 한 노드에서의 총 입력 패킷수와 총 종료 패킷수간의 차이가 포워딩 패킷수와 같아야 하고 또한 총 출력 패킷수와 총 송신 패킷수간의 차가 같아야 한다는 것을 의미한다. 임의의 노드 n의 신뢰도는 F_n에 비례하여 증감한다. 이웃 노드의 패킷을 다른 이웃 노드에게 전달해 줄때는 신뢰도는 $F_n \times 0.01$ 만큼 증가시켜 주고 자신의 패킷을 생성하여 전송할 때는 $\sum S_{pn} \times 0.05$만큼 감소한다. 각 노드의 신뢰도 초기값은 1로 부여하고 신뢰도가 0.8 이하이면 이기적인 노드로 판명하여 자신이 생성한 패킷을 다른 노드에게 전송하지 못하도록 불이익을 주면서 계속 경고 메시지를 보내 다른 노드들이 생성한 패킷을 전달하는 일에 적극적으로 협조하도록 유도한다. 또한 임의의 노드가 패킷 수 Fn을 조작했는지의 여부는 임의의 노드 n으로부터 출력된 패킷수가 해당 이웃 노드들이 n으로 송신한 패킷 수에서 n이 생성하여 전송한 패킷수를 뺀 수와 같은지를 체크하여 판별한다.

SID	NID	SN	I_{pn}	O_{pn}	S_{pn}	T_{pn}	ONS

그림 12. 포워딩한 패킷수를 보고하는 메시지 형식

동일한 순서 번호를 갖는 보고 메시지를 수집한 클러스터 관리자나 클러스터 헤드들은 보고 메시지의 신뢰성을 검증하기 위해 다음과 같은 검증 절차를 수행한다. 첫 번째는 서로 이웃한 임의의 두 노드에 대하여 한 노드로부터 출력된 패킷수가 다른 노드에서 받은 입력 패킷수와 같은지를 검사한다. 두 번째는 포워딩 패킷 수가 식 1을 만족하는지 검사한다. 마지막으로 한 노드에서 생성하여 보낸 패킷 수가 이 패킷을 받는 이웃노드의 ONS와 일치하는지의 여부를 검사한다. 보고된 메시지가 위와 같은 세 가지 검사를 모두 통과하면 신뢰할 수 있는 메시지임을 확인할 수 있다. 보고 메시지가 신뢰할 수 있는 메시지임을 확인한 클러스터 관리자나 클러스터 헤드는 해당 노드의 신뢰도를 보고 메시지 내의 정보를 이용해 계산하여 기록한다. 그리고 신뢰할 수 없는 보고 메시지를 보낸 노드의 신뢰도는 0.1을 차감하고 이 노드에서 생성하여 전송한 패킷은 다른 이웃 노드들에서는 무시된다.

V. 성능 평가

제안 기법들은 GloMoSim(Global Mobile Information System Simulator) V2.03의 PARSEC [16]을 이용하여 네트워크 크기는 100 *100m, 노드의 개수는 150개, 노드의 이동성은 Random waypoint Mobility, 노드의 이동속도는 uniform하게 최소 0m/s에서 최대 20m/s로 설정하고 Network protocol은 AODV인 환경에서 평가하였다.

5.1 SecCBM의 효율성

SecCBM의 효율성은 발생되는 오버헤드로 측정되며 SecCBM의 오버헤드는 저장 공간 오버헤드와 계산 오버헤드, 통신 오버헤드를 들 수 있다.

5.1.1 저장 공간 오버헤드

SecCBM에서 하나의 클러스터에 속한 인증된 노드는 자신이 속한 클러스터의 그룹키, 클러스터 헤드의 공개키와 부분 비밀키, 자신의 개인키와 공개키를 소유한다. 만약 노드 i가 클러스터 관리자이면 자신의 공개키와 개인키, 클러스터 헤드와의 공개키를 소유하고 클러스터 헤드이면 클러스터 관리자와 다른 클러스터 헤드와의 공유키와 자신의 클러스터 멤버와의 통신을 위하여 그룹키를 소유한다. 그리고 멤버 노드는 추가적으로 다른 멤버 노드와의 통신을 위하여 세션키를 소유하게 되므로 각각의 노드 역할에 따라 소유하게 되는 키에 대한 저장 공간이 필요하다. 그리고 임의의 노드 i가 비정상적 노드를 탐지하고 관리하기 위한 저장 공간이 필요하다. 이 기적인 노드를 탐지하기 위하여 클러스터 헤드나 클러스터 관리자는 보고 메시지를 전송받아 검증결과를 저장할 FC 관리 테이블이라는 저장 공간이 필요하다. 그러나 보고 메시지나 FC 카운터에 의하여 생성된 값들은 발생 즉시 처리하기 때문에 별도의 저장 공간이 필요 없다. 악의적인 노드를 탐지하고 관리하기 위하여 모든 노드에는 MN 테이블이라는 저장 공간과 증명서를 임시로 저장할 저장 공간이 필요하다. 또한 임의의 노드 i가 클러스터 관리자이면 클러스터 헤드 리스트를 관리하기 위한 저장 공간이 필요하고, 클러스터 헤드이면 멤버 노드 리스트를 관리하기 위한 저장 공간이 필요하다.

각 키의 길이는 8byte, 테이블의 크기는 30byte로 가정하고 클러스터 헤드의 수는 N_{ch}, 멤버 노드의 수는 N_m, 이웃 노드의 수는 N_n, 이웃 클러스터 헤드는 N_{nch}이라고 정의하고 제안 SecCBM에서의 저장 공간 오버헤드는 기본 데이터 구조 이외에 발생되는 추가적인 저장 공간으로 한정한다. 임의의 노드가 클러스터 관리자이면 $((8\times N_{ch})+90)$byte의 storage 오버헤드가 발생하고, 클러스터 헤드이면 $((8\times N_m\times N_{nch})+98)$ byte, 그리고 멤버 노드이면 46byte의 저장 공간 오버헤드가 발생한다.

5.1.2 계산 오버헤드

SecCBM은 비밀분산 기법과 공개키 방식을 사용하여 안전한 클러스터 기반의 통신과 동적인증이 이루어진다[15]. 비밀 분산 기법과 공개키 방식을 사용하면 키 교환 및 암호화하는데 소비되는 시간이 적기 때문에 이동 노드에게 매우 적은 계산 오버헤드가 발생한다. RC5[17]를 SecCBM의 블록 암호화에 사용하면 8byte의 data를 가진 메시지를 암호하고 하나의 키를 가진 MAC code를 생성하는데 소요되는 시간이 2.38~3.32ms이고 이 연산에 소비되는 에너지는 30μJ이다. SecCBM에서 하나의 이동 노드가 n개의 이웃노드로부터 전송된 패킷을 처리하는데 소요되는 계산 오버헤드는 O(n)이다.

5.1.3 통신 오버헤드

SecCBM에서 노드사이에 데이터를 송수신하는데 소요되는 통신비용을 [18]에 제시된 값을 준용하여 산출한다. [18]에 따르면 36byte의 packet을 수신하는데 12.25μJ/byte가 소요되고, 송신하는 데는 16.25 μJ/byte가 소요된다. 제안하는 SecCBM에서는 클러스터 형성과정과 동적 인증과정, 그리고 비정상 노드 탐지과정에서 통신 오버헤드가 발생한다.

클러스터 형성과정에서는 클러스터 헤드를 선정하고 각각의 클러스터를 구성하기 위한 메시지를 전송하는데 통신 오버헤드가 발생한다. 또한 동적 인증과정에서는 구성 요소사이의 키 설립과 인증하는 과정에서, 비정상 노드 탐지과정에서는 보고 메시지와 신고 메시지, 증명서를 전송하기 위하여 통신 오버헤드가 발생한다. 통신에서 사용되는 모든 종류의 패킷을 8byte로 가정하며, 멤버 노드의 수는 q, 한 클러스터의 적정 노드수는 δ라고 정의하면, 클러스터 형성과정에서 클러스터 헤드의 통신 오버헤드는 초기 브로드캐스트 패킷과 Join 패킷, Accept 패킷을 송수신하는데 $(98+130\delta+130q)\mu J$이 발생한다. 또한 멤버노드가 클러스터 헤드로부터 브로드캐스트 패킷을 p개 받았다면 클러스터 형성과정에서 멤버노드의 통신 오버헤드는 $(16.25+12.25p)\mu J$이 발생한다.

동적 인증과정에서 임의의 노드가 클러스터 관리자이면 $((456\times N_{ch})+(912\times N_{nch}))\mu J$의 통신 오버헤드가 발생하고 클러스터 헤드이면 $(912+(456\times N_{nch})+(1368\times N_m))\mu J$ 통신 오버헤드가 발생한다. 그리고 멤버 노드이면 $(652+(912\times Nm))\mu J$ 통신 오버헤드가 발생한다.

5.2 WBDCA의 안정성

WBDCA의 안정성은 형성된 클러스터 구조에서 클러스터 헤드와 멤버 노드의 변화율을 가지고 평

가한다.

5.2.1 클러스터 헤드의 변화율

클러스터 헤드의 변화가 적은 클러스터링 알고리즘이 클러스터 헤드가 자주 변화는 클러스터링 알고리즘보다 안정적이다. 표 3은 이 논문에서 제안한 가중치 기반 클러스터링 알고리즘의 이동속도와 전송범위에 따른 클러스터 헤드의 평균 변화율을 나타낸다.

실험 결과는 노드의 이동 속도가 증가함에 따라 클러스터 헤드의 역할 변화가 증가하는 것을 보여준다. 노드의 이동 속도가 증가하면 네트워크 전체의 토폴로지 변화도 커지게 되므로 클러스터 헤드의 변화도 커지게 된다. 표 3에서도 노드의 이동 속도가 증가할수록 클러스터 헤드의 변화수가 증가하는 것을 보여준다. 그러나 전송범위가 커질수록 클러스터 헤드의 변화율이 작아짐을 알 수 있다.

5.2.2 노드의 재가입률

재결합수는 실험하는 동안에 이동노드들이 자신이 속한 클러스터를 벗어나 다른 클러스터에 속하는 노드의 수를 말한다. 따라서 재결합수가 적은 클러스터링 알고리즘이 형성된 클러스터를 유지하는 시간이 길기 때문에 재결합수가 많은 클러스터링 알고리즘보다 안정적이라고 할 수 있다. 이 논문에서 제안한 가중치 기반 분산 클러스터링 알고리즘의 재결합수를 이동속도와 전송범위를 다르게 해서 측정한 결과는 아래 표 4와 같다. 표 4에서는 이동속도가 작고 전송범위가 커질수록 이동노드의 재결합수가 작아짐을 보여준다.

표 3. 클러스터 헤드의 평균 변화율

전송범위 / 이동속도	150m	200m	250m
5 m/s	31.7	9.6	5.8
10 m/s	63.0	27.3	17.2
15 m/s	93.4	36.2	23.1
20 m/s	126.6	47,5	31.1

표 4. 전송범위와 이동속도에 따른 재결합수

전송범위 / 이동속도	150m	200m	250m
5 m/s	430.9	329.1	286.3
10 m/s	729.4	745.0	580.9
15 m/s	1159.8	1018.5	767.2
20 m/s	1924.6	1427.5	1038.0

그림 7. 클러스터 헤드와 멤버 사이의 인증 메커니즘

5.3 비정상 노드 탐지 기법의 안전성

제안한 비정상 노드 탐지기법의 안전성은 데이터 전송률과 비정상 노드 탐지율로 평가한다.

5.3.1 데이터 전송률

데이터 전송률은 출발 노드가 보낸 총 데이터 패킷수와 목적 노드가 성공적으로 받은 패킷수로 계산한다. 여기서 출발 노드가 중복된 데이터 패킷을 보내는 것은 수신 노드에서 데이터 패킷을 받지 못하여 재전송한 데이터 패킷으로 전송률을 측정하는데 있어 중요한 요소가 된다. 기존의 연구에서는 버려지는 패킷만을 고려하여 데이터 전송률을 측정하였는데 이 논문에서는 패킷이 어느 정도의 비율로 제대로 전달되었는지를 나타내기 위해 식 2와 같이 데이터 전송률을 산출한다. 이는 정확성과 더불어 신뢰성도 고려한 측정 기준이라 할 수 있다.

$$\text{데이터 전송률} = \frac{\sum_{i=1}^{n} \text{목적지까지전달된패킷수}}{\sum_{i=1}^{n} \text{소스노드에서생성된패킷수}} \quad (2)$$

신뢰도가 높은 노드를 통해 데이터 패킷을 전송할 때 출발노드와 목적지 노드 사이의 패킷 수신율은 일정 시간동안 송수신한 데이터 패킷을 기준으로 한다. 그림 13에서처럼 제안된 기법은 비정상적인 노드가 네트워크 내에 45%에서 50% 정도 존재하여도 적정한 수준의 신뢰성을 보장하는 것을 볼 수 있다. 이는 중간 노드에서 발생되는 비정상적 노드를 제외하고 신뢰할 수 있는 노드들을 통해서만 패킷을 전송하기 때문이다. 그러나 제안한 기법도 비정상적인 노드의 비율이 50%를 초과하면 급격히

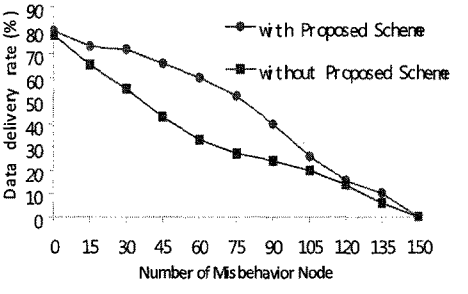


그림 13. 비정상 노드 수에 따른 데이터 전송률

분실되는 패킷 수가 증가하여 데이터 전송률이 현저히 낮아지는 것을 볼 수 있다.

5.3.2 비정상 노드 탐지율

MANET에서는 기존의 네트워크와 MANET는 상이한 특성을 가지고 있기 때문에 기존의 유선 네트워크에서 사용하였던 침입 탐지 기법을 그대로 사용할 수 없다. 그림 14에서는 제안된 기법이 비정상적인 노드 비율이 50%가 될 때까지는 어느 정도 비정상적 노드를 탐지해 냄을 볼 수 있다. 그러나 비정상적인 노드가 50%을 넘어서면 급격히 탐지율이 떨어짐을 볼 수 있다. 이는 MANET의 특성상 침식된 노드가 증가함에 따라 협의된 노드나 false positive가 증가하기 때문이다. 제안 기법이 노드에 IDS를 탑재하여 탐지한 것보다는 다소 탐지율이 떨어지지만 각각의 IDS를 설치하는데 소요되는 비용과 다른 노드에 설치된 IDS와 통신하는데 소요되는 오버헤드가 없다는 장점이 있다.

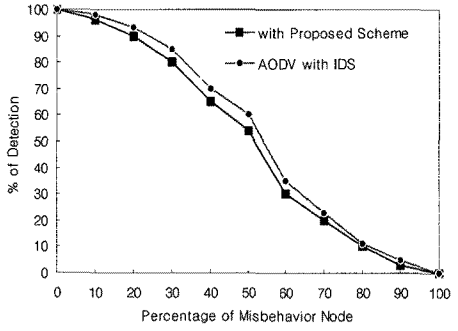


그림 14. 비정상적인 노드 비율에 따른 탐지율

VI. 결 론

MANET에서 다중 홉 경로를 통한 무선 통신은 이웃 노드를 신뢰하지 못하는 상황에서 대부분 이루어지기 때문에 보안상 매우 취약하며 종단 노드 사이에 존재하는 노드들의 비정상 행위는 신뢰성 있는 통신을 어렵게 한다. 따라서 이 논문에서는 이런 문제점을 보완하여 비정상 행위를 하는 노드들을 효율적으로 식별하고 관리하는 SecCBM를 제안했다.

SecCBM는 Proactive Security 모듈,Reactive Phase 모듈로 구성되어 있다. Proactive Security 모듈은 비정상 노드를 사전에 예방하기 위하여 사용되며 이 모듈을 통해 비정상 노드를 효과적으로 예방하고 관리할 수 있는 보안 제어구조가 형성된다. 그리고 형성된 보안 제어구조에서 데이터를 전달하는 과정에 발생하는 비정상 노드를 탐지하고 관리하기 위하여 Reactive Phase 모듈이 사용된다.

Cluster Round 모듈에서는 클러스터를 형성할 때 지역적으로 클러스터를 형성하게 함으로써 초기 발생되는 오버헤드를 감소시켰다. 그리고 부클러스터 헤드와 분산 게이트웨이라는 개념을 정의하여 관리과정에서 발생되는 재클러스터링의 횟수를 감소시켜 기존의 클러스터링 알고리즘보다 안정성을 향상시켰다. 그리고 Proactive Security 모듈에서는 클러스터가 형성되는 과정에 동적 인증기법을 사용하여 구성 요소들 사이에 신뢰 관계를 구축했다. 이를 통하여 신뢰할 수 있는 노드만이 통신에 참여할 수 있게 하였다. 그러나 자원의 제약성과 무선 통신 매체의 사용, 멀티 홉 방식의 통신 취약점으로 인하여 데이터 전송과정에서 비정상 노드가 발생한다. 따라서 이 논문에서는 사후 조치 방법으로 데이터 전송과정에 발생하는 비정상 노드를 효율적으로 탐지하고 관리하기 위하여 비정상 노드 탐지모듈이 사용되었다. 제안된 기법들은 실험을 통하여 기존 기법들보다 안전성 및 안정성, 보안성 측면에서 향상되었음을 보였다.

이 논문에서 제안된 기법들은 유비쿼터스 환경이 완벽히 구축되면 우리 일상생활에서 충분히 활용될 수 있는 기법이므로 제안된 기법들을 현실에 적용할 수 있도록 좀 더 보완하여 구현하는 연구가 지속적으로 진행되어야 한다.

참 고 문 헌

[1] M. Zapata, N. Asokan, "Securing Ad Hoc Routing Protocols," *ACM WiSe*, pp.1-10, Sep. 2002.

[2] C. C. Shen, "CLTC: A Cluster-Based Topology Control Frame w ork for Ad Hoc Networks," *IEEE Tran. on Mobile Computing*, Vol3, No. 1, pp.18-32, Jan. 2004.

[3] C. E. Perkins, Ad Hoc Networking, Addison Wesley, 2000.

[4] S. Basagni, "Distributed Clustering for Ad Hoc Networks," *Proc. of International Symposium on Parallel Architectures, Algorithms and Networks*, pp.310-315, Jun. 1999.

[5] S. Corson, J. Macker, "Mobile Ad Hoc

Networking (MANET)," Internet Draft, IETF, Oct. 1998.

[6] M. Gerla, J. T. C. Tsai, "Multicluster, mobile, multimedia radio network," *ACM/Baltzer Journal of Wireless Networks* 1, 3, pp.225-265, Jul. 1995.

[7] A. K. Parekh, "Selecting routers in ad-hoc wireless networks," *Proc. of ITS*, pp. 420-424, Aug. 1994.

[8] M. Chatterjee, S. K. DAS, "WCA: A Weighted Clustering Algori thm for mobile Ad Hoc Networks," *Cluster Computing*, Vol.5 No.2, pp. 193-204, Apr., 2002.

[9] T. Ohta, "An adaptive multihop clustering scheme for highly mo bile Ad Hoc networks," *IEEE ISADS 2003*, pp.293-300, Apr. 2003.

[10] C. Jaikaeo, C. C. Shen, "Adaptive Backbone-Based Multicast for Ad Hoc networks," *IEEE ICC2002*, pp.3149-3155, May. 2002.

[11] S. Marti, T. J. Giuli, K. Lai, M. Baker, "Mitigating routing misbe havior in mobile Ad Hoc networks," *Mobile Computing and Net working*, pp.255-265, Aug. 2000.

[12] B. Awerbuch, D. Holmer, C. N. Rotaru, H. Rubens, "An On-De mand Secure Routing Protocol Resilient to Byzantine Failures," *ACM WiSe*, pp.21-30, Sep. 2002.

[13] 나가진, 도인실, 편혜진, 채기준, "애드 혹 네트워크에서의 이기적 노드를 관리하기 위한 보안 메커니즘," *JCCI 04*, IV-E-4.1-4, pp.223-223, Apr. 2004.

[14] 황윤철, 이상호, 김진일, "모바일 에드혹 네트워크에서 안정성을 향상 시킨 분산 조합 가중치 클러스터링 알고리즘," *한국컴퓨터정보학회 논문지*, 제12권 제1호, pp.33-42, 2007.

[15] W. Diffie, M. E. Hellman, "New Directions in cryptography," *IEEE Transaction on Information Theroy*, Vol.IT-22, No.6, pp.644-654, Nov. 1976.

[16] R. Bagrodia, R. Meyer, M. Takai, Y. hen, X. Zeng, J. Martin, b. Park, H. Song, "PARSEC: A Parallel Simulation Environment for Complex System," *Computer Magazine*, Vol.31, 10, pp. 77-85, Oct. 1998.

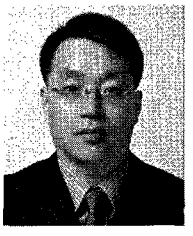
[17] R. L. Rivest, "The RC5 encryption algorithm," in: Bart Preneel (ED), Fast Software Encryption, Springer, pp.86-96, Mar. 1995.

[18] L. B. Oliveira, Hao Cho Wang, A. A Loureiro, "LHA-SP:secure pr otocols for hierarchical wireless sensor networks," *in 9th IFIP/IE EE International Symposium on Integrate Network Management*, pp.31-44, May. 2005.

[19] 황윤철, 김진일, "이동 애드혹 네트워크에서 신뢰성 있는 클러스터기반 동적 인증 기법," *멀티미디어학회 논문지* 제11권 5호, pp 661-672. 2008.

[20] 이근호, 한상법, 서혜숙, 이상근, 황종선, "이동 Ad Hoc 망을 위한 다중 계층 클러스터링 기반의 인증 프로토콜," *한국정보과학회논문지*, 제33권 4호, pp.310-323, Apr. 2006.

황 윤 철 (Yoon-cheol Hwang) 정회원



1994년 2월 한남대학교 전자계산공학과 졸업

1996년 8월 한남대학교 전자계산공학과 석사

2008년 2월 충북대학교 전자계산학과 박사

현재 충주대학교 첨단과학기술대학 박사후 연구원

<관심분야> Ad Hoc 및 센서 네트워크 보안, IDS, ITS