隨意式網路中監聽節點之偵測與隔離

蘇民揚　陳建民
銘傳大學資訊工程研究所
minysu@mcu.edu.tw　phyrexia0415@yahoo.com.tw

摘要

隨意式網路中有些攻擊是惡意節點必須處在監聽模式並與其它節點執行相同繞路協定才能進行的，針對此類攻擊，我們提出了一個偵測監聽與隔離的機制。利用在 AODV 中增加一個新的控制封包來偵測相鄰節點是否有做監聽的動作，並且透過隔離機制將監聽節點隔離在 AODV 網路之外，使其無法進行攻擊。本文使用 Ns-2 模擬軟體來評估我們設計的偵測與隔離機制，結果顯示我們的機制不但能有效偵測及隔離監聽節點，同時對網路的負擔也不大。

關鍵字：監聽、節點隔離、隨意式網路、AODV、Ns-2

Abstract

In the AD HOC, some attacks which could be achieved must be under sniffing mode and under the same routing protocol. Aimed at these attacks, we propose a detect sniffing and isolate mechanism. We add a new control packet to detect neighbor nodes are sniffing or not, and use isolate mechanism to isolate the sniffing node out of AODV network, then the sniffing node can not attack any more. In this paper, we use Ns-2 simulator to estimate our detection and isolation mechanism. From the simulation result, we can see that our detect mechanism can detect and isolate sniffing nodes effectively and will not give too much load to network.

Keyword：AD HOC Network、AODV、Sniffing、Node Isolation、Ns-2

1. 前言

無線區域網路上的攻擊方式有很多種類，因此衍生出很多種偵測攻擊的方式，例如對繞路協定攻擊做偵測[4][5][6][8][12][14]，對一般攻擊做偵測[1][9]，也有針對特定攻擊做偵測，如DoS/DDoS攻擊[10][13][15][16]、重送攻擊[11]、黑洞攻擊[3][1]等等，以上都是利用攻擊的特徵作為偵測的基礎，其偵測率也無法達到100%。而且有的僅能偵測攻擊事件發生，無法得知來源節點為何，因此無法對惡意的節點進行處罰。

許多的攻擊，例如黑洞攻擊、中間人攻擊、隧道攻擊、中斷路徑、節點隔離等都必須要經由監聽模式來取得被害者的資訊或是修改所搜集的控制封包始能達到攻擊的目的，而且這類的攻擊節點必須與其它節點執行相同繞路協定才能執行。本文就是以偵測監聽模式為作為偵測攻擊的基礎並且偵測率能夠達到100%。另外我們能夠清楚得知惡意節點為何，因此能夠將惡意節點隔離在AODV之外。先偵測出監聽節點，就能知道那些節點有可能進行攻擊，我們再將監聽節點隔離在AODV之外，如此就能早一步防止這些攻擊的發生。

本文以下之章節架構如下，第二節為相關文獻探討，第三節陳述主要之偵測與隔離的系統架構，第四節是模擬的實驗及結果，第五節為結論。

2.背景知識

2.1 AODV 中的各種攻擊

AODV 是 AD HOC 中 Table-Driven Routing Protocols 的其中一種，在 AODV 協定中，若 A 節點想傳送資料給 D 節點，而他們之間又沒有存在任何路徑的話，則 A 節點就要廣播 RREQ 封包到網路上尋找路徑，若有節點知道到 D 的路徑，則

會回覆一個 RREP 給 A，如此即可建立路徑並開始傳送資料。AODV 在設計時是假設每個節點都是正常的節點，沒有考慮安全性的問題，所以惡意節點想要攻擊 AODV 的協定或是正常的節點是相當容易的。我們將 AODV 中的攻擊歸納成下面四種類型：

■ 身份認證攻擊
■ 資源可利用性攻擊
- 丟棄封包 (Dropping of Packets)
- 黑洞攻擊 (Blackhole)
- 捏造錯誤攻擊 (Fabrication Attacks)
- 資源消耗攻擊 (Resource Consumption)
- 選擇性存在攻擊 (Selective Existence)

■ 完整性的攻擊
- 單一攻擊 (Single Attacks)
  - 偽造序號 (Forging Sequence Number)
  - 偽造跳躍數 (Forging Hop Count)
- 複合式聚集攻擊 (Aggregated Attacks)
  - 中間人攻擊 (Man-in-the-Middle Attack)
  - 隧道攻擊 (Tunneling Attacks)

隔離節點(Isolate node)
中斷路徑(Route disruption)

■ 機密與隱私攻擊

其中黑洞攻擊、偽造序號攻擊、偽造跳躍數攻擊、中間人攻擊、隧道攻擊、中斷路徑、節點隔離、以及機密與隱私攻擊等都必須要配合監聽模式才能夠進行攻擊，而且這些攻擊也都要在相同的繞路協定下才能夠運作，因此我們的偵測機制若能有效偵測出監聽的節點，並能將其隔離在 AODV 之外，就能夠有效防止這些攻擊的發生。

2.2 有線上的偵測監聽方式

正常情況下，送出偽造 MAC 位置的 ARP 封包以及 ICMP 封包不會有人回應，該封包會被網路卡的硬體過濾器過濾掉，MAC 偵測[7][17]就是利用在監聽時，硬體過濾器不會動作，而對偽造 MAC 位置的封包做出回應的原理，偵測網路上是否有主機設定在錯亂(Promiscuous)模式進行監聽的動作。如表 1，根據我們的測試，在 AD HOC 中使用 MAC 偵測時，Windows 系列的作業系統只會回應正常的 ARP 或 ICMP 封包，而填入其他的 MAC 位置都接收不到封包，不過若是在把 MAC 位置的第 8 個 bit 設成 0，其他 47 個 bit 不論設為 0 或是 1 情況下(B80)，就可以接收到偽造的 ARP 及 ICMP 封包，不過一樣不會回應。而在 Linux 的情況下就與 Windows 恰好相反，所有的正常及偽造的 ARP 與 ICMP 封包都能夠正常的接收與回應，只有在(B80)的情況下會收不到封包。也就是說，要在 AD HOC 中使用 MAC 偵測只能在 Linux 系列的作業系統中使用，在 Windows 系列則無法使用。

表 1 ARP 偵測在 AD HOC 中之表現

OSs / HW address		Windows XP		Windows 9X / Windows Me		Windows NT / Windows 2000		Linux 2.6x kernel	
		Recv.	Reply	Recv.	Reply	Recv.	Reply	Recv.	Reply
FF:FF:FF:FF:FF:FF	Br	O	O	O	O	O	O	O	O
FF:FF:FF:FF:FF:FE	B47	X	-	X	-	X	-	O	O
FF:FF:00:00:00:00	B16	X	-	X	-	X	-	O	O
FF:00:00:00:00:00	B8	X	-	X	-	X	-	O	O
Xxxx0:XX:XX:XX:XX:XX	B80	O	X	O	X	O	X	X	-
01:00:00:00:00:00	Gr	X	-	X	-	X	-	O	O
01:00:5E:00:00:00	M0	X	-	X	-	X	-	O	O
01:00:5E:00:00:01	M1	O	O	O	O	O	O	O	O
01:00:5E:00:00:02	M2	X	-	X	-	X	-	O	O
01:00:5E:00:00:03	M3	X	-	X	-	X	-	O	O

O：成立　X：不成立　-：無資料

RTT 時間是指一個封包從送出到一個主機到收到回應的時間，RTT 偵測[7]是利用在監聽時，接收的網路流量大增，導致 RTT 時間變大。只要對可疑主機測量 RTT 時間即可判斷是否正在做監聽的行為。

DNS 偵測[7][17] 利用密碼搜集監聽時共通的行為來做偵測，關鍵是密碼搜集監聽並非完全是被動的，事實上密碼搜集監聽會產生一些網路的流量，雖然很難分辨出哪些是正常的流量，哪些是監聽所產生的流量。這證明了以前所遇過的密碼搜集監聽會在它所監聽的流量做反向DNS查詢，若是這些流量是由監聽程式所產生，技巧是利用某種方式去偵測此種DNS查詢以及如何分別它與正常的DNS查詢請求的不同。

負載偵測[7][17]與 RTT 偵測的觀念相同只是負載偵測是針對 FTP 服務，對可疑主機發送一些 FTP 的流量來測量 RTT 時間，藉以判斷是否正在做監聽的動作。

在有線網路上面雖然能夠偵測出監聽的節點，但是無法將之隔離，不過無線上並沒有線路連接，因此可以將偵測出的節點隔離之。

3. 研究方法

3.1 AD HOC 中的偵測監聽方式

在本篇論文中，我們是利用分散式系統，每個節點各自偵測相臨的節點是否處於錯亂模式以及各自隔離自己黑名單中的惡意節點。我們設計了一個自我偵測的機制，在 AODV 中新增了一個封包格式 PROD (Promiscuous mode Detect) ，如圖 1，每個節點定期發出一個 PROD 封包給相臨節點，其 Status 欄位的值為 Request。當節點收到此 PROD 封包時，會測試節點本身的網路卡，若是網路卡設定在錯亂模式，則會回應另一個 PROD 封包，其 Status 欄位的值則為 Promiscuous，若網路卡設定為正常模式，其 Status 欄位的值則為 Normal。當節點收到 Status 欄位為 Promiscuous 的 PROD 的封包時，表示偵測到有相臨節點處於錯亂模式中時，則將該節點加入自己的黑名單行列中進行隔離的動作。

0	8	15
Type	Status	

圖 1. PROD 封包格式

3.2 AD HOC 中的隔離方式

我們反制的目的，是希望將惡意的節點隔離在 AD HOC 網路之外，我們的隔離方式是每個節點在收到控制封包時，比對封包標頭中的來源點以及目的地，若是與黑名單中的節點符合，則丟棄此封包。如此一來之後所建立的路徑將會繞過黑名單中的節點，且黑名單中的節點也無法建立路徑。如圖 2 所示，A 點就被隔離在 AODV 之外，這樣就

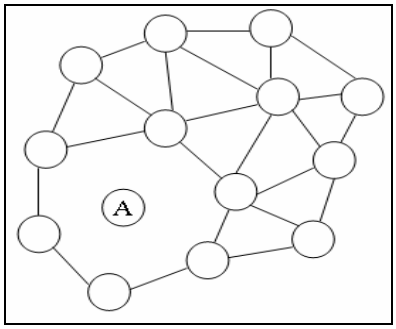

圖 2. 節點隔離圖示

達到隔離的效果。

3.3 系統流程

如圖 3，在系統開始時，每個節點先取得直周圍相臨節點的資訊，隨後開始每 5 秒發送一次 PROD 封包給周圍的相臨節點並等待回應。在收到 PEOD 封包時，會先判斷其中的 Status 欄位，若 Status 欄位的值為 Request，則測試本身的網路卡，若是網路卡設定在錯亂模式，則會回應另一個 PROD 封包，其 Status 欄位的值則為 Promiscuous，若為正常模式，其 Status 欄位的值則為 Normal。若 Status 欄位的值為 Promiscuous，表示是收到相臨節點所回應的 PROD 封包，且該節點的網路卡是設定為錯亂模式，因此將該節點加入到黑名單中，且將 Routing Table 中該節點已建立之路徑刪除，並且執行隔離的動作。若該節點已在黑名單中則不做任何事；若 Status 欄位的值為 Normal，表示該節點為正常節點，因此不做任何事。

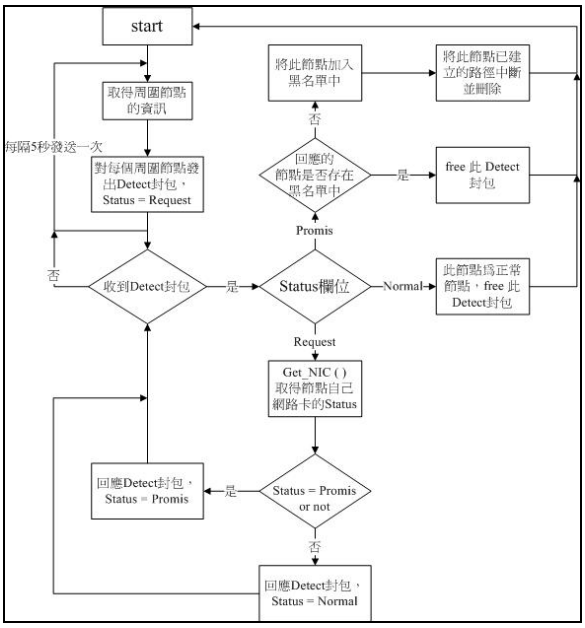

圖 3. 系統流程圖

4. 實驗結果

我們使用 Ns-2 (Network Simulation-2)來進行模擬，使用的平台為 Windows XP，CPU 速度為 3.2G，記憶體為 1G MB，環境設定如下：

30 個節點隨機分布在 600x600 公尺的範圍中，每個節點以 0~10m/s 的速度隨機移動，30 個點中任意挑出 20 個節點分為 10 組進行 UDP 連線，封包大小為 512 bytes，速率為每秒 2Kb 資料。

剩餘 10 個節點任意挑出 0、2、4、8 個節點做為監聽節點，總模擬時間為 1000 秒，實驗數據以 10 次實驗取平均值

■實驗項目及結果

我們以下列三個實驗來測試我們的偵測與隔離機制對網路的影響：

- ■ 延遲時間與監聽節點數及停留時間之關係
- ■ 延遲時間與監聽節點數及速度之關係
- ■ 封包送達率與監聽節點數及速度之關係

實驗的說明及數據分析如下：

■延遲時間與監聽節點數及停留時間之關係

在這個實驗中，我們設定每個節點以 0 ~ 10 m/s 的速度隨機移動，且到達目的地後的停留時間(pause time)分別為 0 秒、50 秒、100 秒及 150 秒，再向下一個目的地前進。計算封包延遲時間(delay time)與監聽節點多寡及停留時間之間的關係。監聽節點個數依照 0、2、4、8 的順序增加，每增加一次做 10 次實驗取其平均值。

由圖 4 可看出，封包傳送的延遲時間與停留時間並無太大的關係，主要是與監聽節點的多寡關係較明顯。由於監聽節點被偵測出來之後會遭其他的節點隔離並且將其原有的路徑刪除，因此所有經過監聽節點的路徑都必須重新發起路徑搜尋請求來找到新的路徑。而且新建立的路徑又必須繞過監聽節點，所以新建立的路徑有相當大的機會較舊有的路徑長，因此傳送封包的時間就會比較久。而且我們可以看到延遲時間成長的關係並不是線性的，這是因為在監聽節點數量少的時候，監聽節點佔所有節點的比例較低，建立路徑時經過監聽節點的機會很少，因此所造成的影響不太明顯，但是監聽節點數量多的時候，建立路徑時很容易會經過監聽節點。而且在隔離監聽節點之後，整個網路的節點都變少了，等於是節點密度降低了，路徑中斷的機會也相對提高許多，所以才會造成這種現象。

圖 5 是比較正常的 AODV 與我們提出的 AODV 的延遲時間，由圖 6 可看出原本的 AODV (normal AODV) 延遲時間比我們所提出的 AODV (m_node = 0) 要短一些，這是因為我們提出的 AODV 會定期發出 PROD 封包到網路上，雖然是用來偵測監聽節點，但是仍會造成些許的網路負擔，不過發出 PROD 封包的間隔可以視需要的安全性及網路壅塞情況而調整以減少對網路造成的負擔。由於 AODV 中存在路徑修復機制，隨著監聽節點的數量增加，在其被隔離之後，會不斷執行路徑修復機制試圖重新找回路徑，因此會廣播許多 RREQ 到網路上面，造成網路流量的增加，進而造成網路延遲時間增加。

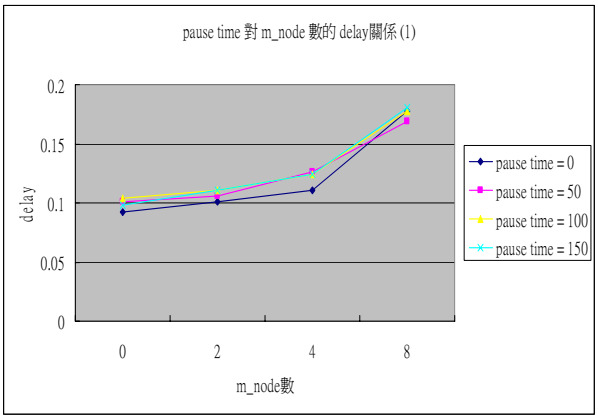

圖 4. 延遲時間與監聽節點數及停留時間之關係

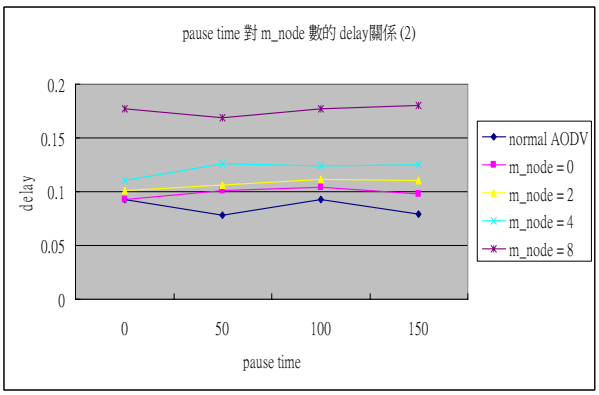

圖 5. 延遲時間與監聽節點數及停留時間之關係

■延遲時間與監聽節點數及速度之關係

在這個實驗中，我們設定每個節點的移動速度分別為 10m/s、30m/s 以及 50m/s 的順序增加，每增加一次做 10 次實驗取其平均值。停留時間為 10 秒，計算封包延遲時間與監聽節點多寡及移動速度之關係。監聽節點依照 0、2、4、6、8 的順序增加，每增加一次做 10 次實驗取其平均值。由圖 6 可看出在移動速度較低的時候(Speed = 10 m/s 及 30 m/s)，監聽節點的數量對於延遲時間的影響比較不明顯，要到 4 個以上的監聽節點存在時，延遲時間才開始明顯上升；而在快速移動時(Speed = 50 m/s)，延遲時間成長快速，這是因為快速移動造成

路徑中斷的機會增加，因此路徑必須經常重新搜尋，造成延遲時間的增加，以及被隔離的監聽節點數量增加造成有效節點減少使得延遲時間增加，兩者互相加成使得延遲時間大幅度的成長。因此在速度越快的時候，延遲的情況會更加嚴重。

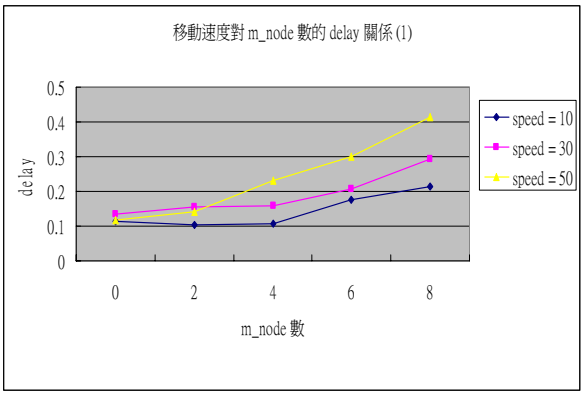

圖 6. 延遲時間與監聽節點數及速度之關係

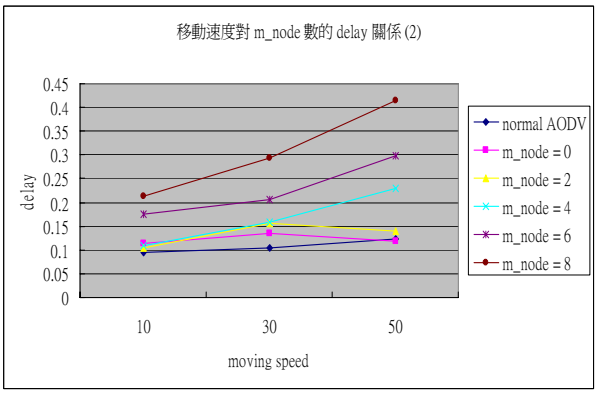

圖 7. 延遲時間與監聽節點數及速度之關係

由圖 7 可以看出來，在監聽節點的數量不多的時候(m_node < 4)，移動速度對延遲時間的影響並不大，只有在監聽節點的數量較多的時候(m_node > 4)，移動速度對延遲時間的影響才會變得比較明顯，而且可以看出有快速上升的趨勢。

■封包送達率與監聽節點數及速度之關係

設定移動速度分別為 10m/s、50m/s，每增加一次做 10 次實驗取其平均值。停留時間為 10 秒，計算封包送達率與監聽節點多寡及移動速度之關係。監聽節點依照 0、2、4、6、8 的順序增加，每增加一次做 10 次實驗取其平均值。從圖 8 我們可以看到在速度較快的時候(Speed = 50)，封包到達率明顯比速度慢的時候少了許多，而且在監聽節點數量增加的時候(m_node > 4)還有繼續下降的趨勢，而速度慢的時候幾乎都是持平，沒有什麼下降的趨勢。

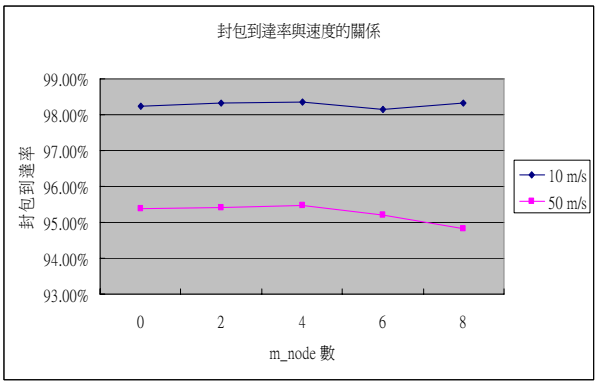

圖 8. 封包到達率與速度的關係

藉由以上的實驗我們可以看出我們所提出的 AODV 版本利用分散式的偵測系統能夠節省下利用 IDS 主機做偵測所需要負擔的硬體成本，但是所要花費的代價就是會增加控制封包的數量，使得網路流量增加，以及每個節點會多消耗許多功率在發送 PROD 封包上面，這是我們想要解決的問題，如何才能在不增加網路流量的狀況下，有效的隔離一個節點。在實驗一與實驗二中我們可以看到，延遲時間跟停留時間(pause time)長度以及移動速度快慢的關係較不密切，主要還是跟被隔離的監聽節點數量有關。

5. 結論

在本篇論文中，我們提出了一個新的 AODV 版本，目的在於能夠有效的偵測出網路中的監聽節點並且將之隔離在 AD HOC 網路之外，以保護正常節點的資訊安全。在系統架構上，由於是使用分散式偵測系統，將偵測的工作分散到網路中的每個節點上，因此節省了另外增加 IDS 主機的成本，不過因為需要廣播許多的偵測封包，也會因此增加每個節點的工作量以及功率消耗。

由於本論文的偵測機制是包含在 AODV 協定之中，因此必須與原本 AODV 協定不相容，若節點是使用原本的 AODV 協定，則在收到我們版本的 AODV 的控制封包時，將不予理會，同時也無法辨識其中的內容。若是惡意節點想要參與繞路，就必須要使用我們版本的 AODV 協定才行，而且要進行許多的攻擊之前必須要先取得被攻擊節點

的資訊或是其他的控制封包，因此就必須要利用監聽模式來擷取網路上的封包，這時惡意節點就會被我們的偵測機制給偵測出來並且隔離在 AD HOC 網路之外，所以若是能夠有效的偵測監聽節點，就能夠先一步防止攻擊的發生，使網路的安全性上升。

在未來後續研究上面有以下目標，如果有節點有特殊用途需要用到監聽模式，例如 IDS 主機，那麼此種情況就必須另外考量。此外若是有節點不誠實回報 PROD 封包的話，我們可以利用監聽節點的特性，它會持續存在於被監聽節點的廣播範圍內。正常若是在移動的情況下，並不會持續停留在其他節點廣播範圍內。我們只要找出有節點長時間持續存在於某節點的廣播範圍內的話，那麼該節點就有相當大的機會是在做監聽的動作。

參考文獻

[1] 張舜超，“一個解決隨意網路黑洞問題的分散式機制”，碩士論文，中華大學資訊工程研究所，民國九十三年六月

[2] Amitabh Mishra, Ketan Nadkarni, Animesh Patcha, and Virginia Tech, “Intrusion Detection Wireless AD HOC Networks”, *IEEE Wireless Communications*, Volume 11, Issue 1, pages 48-60, February 2004.

[3] Bo Sun, Yong Guan, Jian Chen, and Pooch U. W., “Detecting Black-Hole Attack in Mobile AD Hoc Networks”, *IEEE Personal Mobile Communications Conference*, pages 490- 495, April 2003.

[4] B. Sun, K. Wu, and U.W. Pooch, “Routing anomaly detection in mobile ad hoc networks”, *Computer Communications and Networks*, pages 25-31, Oct. 2003

[5] B. Sun, K. Wu, and U.W. Pooch, “Towards Adaptive Intrusion Detection in Mobile AD Hoc networks”, *IEEE Global Telecommunications Conference*, Volume 6, pages 3551- 3555, Dec. 2004.

[6] Chin-Yang Tseng, Poornima Balasubramanyam, Calvin Ko, Rattapon Limprasittiporn, Jeff Rowe, and Karl Levitt, “A Specification-based Intrusion Detection System for AODV”, *Proceedings of the 1st ACM workshop on Security of ad hoc and sensor networks*, pages 125-134, 2003.

[7] D. Wu and F. Wong, “Remote Sniffer Detection,” *Computer Science Division, University of California, Berkeley*, December 14, 1998.

[8] Giovanni Vigna, Sumit Gwalani, Kavitha Srinivasan, Elizabeth M. Belding-Royer, Richard A. Kemmerer, “An Intrusion Detection Tool for AODV-based AD Hoc Wireless Networks”, *Computer Security Applications Conference*, pages 16-27, Dec. 2004.

[9] Hongmei Deng; Wei Li; Agrawal, D.P., “Routing Security in Wireless AD Hoc Networks”, *IEEE Communications Magazine*, Volume 40, Issue 10, pages 70-75, Oct. 2002.

[10] I. Aad, J.P. Hubaux, and E. Knightly, “Denial of Service Resilience in AD Hoc Networks”, *Proceedings of the 10th annual international conference on Mobile computing and networking* , pages 202-215, September 2004.

[11] Ketan Nadkarni, Amitabh Mishra, “A Novel Intrusion Detection Approach for Wireless AD Hoc Networks”, *Wireless Communications and Networking Conference*, pages 831- 836, March 2004.

[12] Peng Ning , Kun Sun, “How to Misuse AODV: a Case Study of Insider Attacks Against Mobile AD Hoc Routing Protocols”, *Ad Hoc Networks*, Volume 3, Issue 6, pages 795-819, November, 2005.

[13] Gupta, V. Krishnamurthy, S. Faloutsos, M. , “Denial of Service Attacks at the MAC Layer in Wireless AD Hoc Networks”, *MILCOM 2002*, pages 1118-1123, October 2002.

[14] Weichao Wang , Yi Lu , Bharat K. Bhargava, “On Vulnerability and Protection of AD Hoc On-demand Distance Vector Protocol”, *Telecommunications, 10th International Conference*, Volume 1, pages 375-382, March 2003.

[15] Yan Xia; Ren-Fa Li; Ken-Li Li, “Intrusion Detection Using Mobile Agent in AD Hoc Networks”, *Proceedings of the International Conference on Machine Learning and Cybernetics*, Volume 6, pages 3383-3388, Aug. 2004.

[16] Yongguang Zhang, Wenke Lee, Yi-An Huang, “Intrusion Detection Techniques for Mobile Wireless Networks”, *Wireless Networks*, pages 545-556, 2003.

[17] Trabelsi, Z., Rahmani, H., Kaouech, K., Frikha M., “Malicious Sniffing Systems Detection Platform”, *Proceedings of the International Symposium on Applications and the Internet*, pages 201-207, 2004.