وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR UNIVERSITY -ANNABA
UNIVERSITE BADJI MOKHTAR -ANNABA

جامعة باجي مختار
- عنابـــة

Faculté: Sciences de l'Ingéniorat

Année 2017-2018

Département: Informatique

THÈSE

Présentée en vue de l'obtention du diplôme de **Doctorat 3ème cycle**

Détection d'intrusion distribuée dans les réseaux ad hoc mobiles

Option : Réseaux et Sécurité
Informatique

Par

Leila MECHTRI

Devant le jury

Présidente	Mme. Labiba Souici Meslati	Professeur à l'Université d'Annaba
Directrice de thèse	Mme. Fatiha Djemili Tolba	MCA à l'Université d'Annaba
Examinateur	M. Mohamed Tahar Kimour	Professeur à l'Université d'Annaba
Examinateur	M. Pascal Lorenz	Professeur à l'Université de haute Alsace, France
Examinateur	M. Azzedine Bilami	Professeur à l'Université de Batna
Invité	M. Salim Ghanemi	Professeur à l'Université d'Annaba

وزارة التعليم العالي والبحث العلمي

BADJI MOKHTAR UNIVERSITY -ANNABA
UNIVERSITE BADJI MOKHTAR -ANNABA

جامعة باجي مختار
- عنابـــة

Faculty: Engineering Sciences
Department: Computer Science

Year: 2017-2018

Thesis

Submitted in partial fulfilment of the requirements for the degree of
Doctorate 3rd cycle

Distributed intrusion detection in mobile ad hoc networks

Option: Networks and
Information Security

Presented by

Leila MECHTRI

Jury

President	Mrs Labiba Souici Meslati	Professor at University of Annaba
Supervisor	Mrs Fatiha Djemili Tolba	MCA at University of Annaba
Examiner	Mr Mohamed Tahar Kimour	Professor at University of Annaba
Examiner	Mr Pascal Lorenz	Professor at University of haute Alsace, France
Examiner	Mr Azzedine Bilami	Professor at University of Batna
Invited guest	Mr Salim Ghanemi	Professor at University of Annaba

To my family...

Acknowledgements

First, I would like to thank the members of my thesis committee: Pr. Labiba Souici Meslati, Pr. Mohamed Tahar Kimour, Pr. Pascal Lorenz, and Pr. Azzedine Bilami, for accepting to evaluate this work.

I would like to thank my supervisor Dr. Fatiha Djemili Tolba for her guidance, continuous support, and for all the interesting discussions we had during this thesis. Thank you for your encouragement and patience.

I was very fortunate to have the opportunity to work with Pr. Salim Ghanemi. I would like to thank him for his endless support, encouragement, and valuable advice.

I would like also to thank Pr. Nacira Ghoualmi-Zine, Director of the LRS Laboratory for accepting me as a member of her Laboratory.

I would like to thank Pr. Damien Magoni for his support during my internship at the LaBRI.

Finally, I am deeply grateful to my family, my parents and sisters, for their continuous love, encouragement and support all the way since the beginning.

Abstract

Mobile ad hoc networks are getting more and more involved in our daily lives. For instance, these networks are used in different applications pertaining to educational, commercial, military field and many other interesting domains. Despite the benefits obtained from the use of these networks, they are particularly vulnerable to security threats. Several approaches have been proposed to solve these problems, but the perfect solution has not yet been found. In this thesis, we are interested in the development of a new intrusion detection and response system (IDRS)to reinforce the defence systems of these networks.

As a first step, we have prepared a thorough state of the art on security and in particular intrusion detection and response in ad hoc networks. Based on the findings of this literature study, we proposed a new architecture for a distributed Intrusion Detection System (IDS). This architecture consists of a set of Local intrusion detection systems (LIDS), where the detection tasks are distributed on a set of mobile and stationary agents. Realizing that multi-agent systems are fault-prone, it seems interesting to improve the proposed approach so that to better adapt our IDS to the specificities of mobile ad hoc networks. We have, therefore, integrated a new agent replication mechanism to help the automatic and fast recovery of our IDS to a correct state in case of failure. More interestingly, the response of our IDRS is adapted and optimized so as to distinguish and respond to the detected intrusions according to their severity and their distribution over time (in case of multi-step attacks). Finally, a self-healing mechanism is added to the system to automatically recover damaged data and services.

Keywords: intrusion detection; intrusion response; mobile ad hoc network; distributed and cooperative; multi-agent system; intrusion severity; self-healing.

Résumé

Les réseaux mobiles ad hoc sont de plus en plus utilisés dans les différents domaines de notre vie (education, le domaine commercial, le domaine militaire, etc.). En dépit des avantages apportés par ces réseaux, ils sont très vulnérables aux problèmes de sécurité. Plusieurs approches ont été proposées pour résoudre ces problème, mais la solution parfaite n'a pas encore été trouvée. Dans cette thèse, nous nous sommes intéréssés au développement d'un nouveau système de détection et de réaction aux intrusions.
Dans un premier temps, nous avons préparé un état de l'art approfondi sur la sécurité et en particulier la détection d'intrusions dans les réseaux mobiles ad hoc. En se basant sur les conclusions tirées de cette étude bibliographique, nous avons proposé une nouvelle architecture pour un système de détection d'intrusions (IDS) distribué. Cette architecture consiste à intégrer un système de détection d'intrusions local, dont les tâches de détection sont distribuées sur un ensemble d'agents mobiles et stationnaires, au niveau de chaque nœud mobile dans le réseau. Vu que les systèmes multi-agents souffrent, souvent, de plusieurs problèmes critiques tels que l'échec d'un agent, nous avons adopté un nouvel mécanisme de réplication d'agents pour garantir une supervision continue de notre réseau.
La deuxième partie de cette thèse est consacrée à l'optimisation de la réaction (réponse aux intrusions) de notre IDS d'une façon à lui permettre de distinguer et de répondre aux attaques détectées selon leurs sévérités. Finalement, un mécanisme de self-healing a été ajouté au système pour permettre de récupérer d'une façon automatique les données affectées par l'intrusion(s).

Mots-clés: détection d'intrusions; résponse aux intrusions; réseau ad hoc mobile; système distribué et coopératif; système multi-agents; séverité d'intrusion; self-healing.

الملخص

شهدت السنوات الأخيرة استعمالا واسع النطاق للشبكات اللاسلكية اد هوك في عدة مجلات تمس الحياة اليومية للأفراد مثل التعليم، القطاعين الاقتصادي و العسكري و غيرها. على الرغم من هذا الانتشار الواسع و الامتيازات الكثيرة التي تتيحها إلا أنها مازالت تعاني من بعض المشاكل خاصة تلك المتعلقة بالجانب الأمني. يهتم العديد من الباحثين بتطوير أنظمة حماية خاصة بهذه الشبكات إلا أنه لم يتم التوصل إلى الحل المثالي الذي من شأنه أن يسد جميع الثغرات الأمنية نهائيا. و لهذا فإن المجال يبقى مفتوحا لمزيد من الأبحاث على أمل إيجاد حل لهذه المشكلة.

نركز في هذه الرسالة على تطوير نظام جديد للكشف عن الاختراق والرد عليه وكخطوة أولى ، قمنا بإعداد دراسة شاملة في مجال الأمن وخاصة كشف الاختراق في الشبكات اللاسلكية. بناء على نتائج هذه الدراسة ، اقترحنا بنية جديدة لنظام كشف الاختراق الموزع. تعتمد هذه البنية أساسا على توزيع مهام النظام على مجموعة من العناصر المتنقلة والثابتة. أيضا قمنا بتكييف وتحسين استجابة النظام للاختراقات المكتشفة بحيث يمكنه التمييز و الاستجابة للتدخلات وفقًا لشدتها وتوزعها عبر الوقت. وأخيرًا ، تمت إضافة آلية الاسترجاع الذاتي إلى النظام لاسترداد البيانات والخدمات التالفة تلقائيًا.

الكلمات المفتاحية: كشف الاختراق، الرد على الاختراق، شدة الاختراق، الشبكات اللاسلكية اد هوك، الاسترجاع الذاتي.

Contents

List of Figures

List of Tables

Chapter 1

General Introduction

1.1 Problem statement and motivation

Over the last few decades, Mobile Ad hoc NETworks (MANETs) have raised several challenging security-related issues. The inherent nature of the wireless medium together with the distributive structure of these networks makes them particularly vulnerable to a wide variety of security threats ranging from passive eavesdropping to active interference. Moreover, these networks are highly resource constrained in terms of network topology, memory and computational abilities, which complicated the design and deployment of security solutions. Considering these issues and the fact that not all attacks are preventable, securing MANETs by means of preventive security mechanisms such as access control, authentication, and encryption is deemed unsatisfactory.

These solutions become useless in some contexts like incidents involving insider attackers. For these reasons, intrusion detection systems (IDSs) are necessary as a second line of defence to better defend against both insider and outsider attacks. In MANET, IDSs are generally classified into four main architectures, namely, stand-alone IDSs, distributed and cooperative IDSs, hierarchical IDSs and agent-based IDSs. Contrary to stand-alone IDSs, where the detection process is performed on each node locally and independently, distributed and cooperative IDSs suggest that every node must participate cooperatively in intrusion detection and response. Hierarchical IDSs, on the other hand, are the most suitable for multi-layered networks where the network is divided into clusters. A cluster head is selected in order to collect security-related information from nodes in a cluster and to determine if an intrusion has occurred. The last architecture of MANET IDSs, referred to as agent-based IDS architecture, is based on the distribution of the intrusion detection tasks amongst a number of agents.

The use of the stand-alone architecture is impaired by the nodes' limited view and

their vulnerability to distributed attacks. Extending this architecture to a distributed and cooperative one by allowing nodes to share their detection data and to use secondhand information helps in coping with its limitations. Unfortunately, distribution and cooperation entail additional overhead of IDS communication and data sharing. In addition, cooperation requires that cooperating nodes should have an acceptable level of trustworthiness. Hierarchical IDSs, with their structured architecture, try to reduce the overhead caused by totally distributed IDSs and to better organize cooperating entities. Similarly, the deployment of this architecture is restrained by the incurred latency in detection and the difficulty and overhead incurred by nodes' mobility to maintain the established hierarchies.

Contrary to the previously discussed architectures that have been excessively used for the development of MANET IDSs, the studies that approach agent-based IDSs were quite few in the early years of IDS deployment in MANET. This is mainly due to: (i) the additional complexity involved in developing agent-based IDSs especially as this technology is known for introducing new challenges with respect to security mainly when dealing with mobile agents and (ii) the lack of experience in formulating agent-based solutions to applications. However, as they, recently, proved several advantages, software agents are attracting a lot of attention especially for their suitability for the building of distributed applications. It is for this reason that many researchers are encouraged to explore more possibilities for the application of software agents in the context of MANET intrusion detection mainly together with other architectures to cope with the issues they raise.

Regardless of the adopted architecture, intrusion detection when executed alone detects the presence of anomalies and/or some specific node misbehaviour and may also allow to identify and localize intruders but does not offer any options to stop the ongoing threat. This task is usually handled by dedicated intrusion response systems that can either be integrated within the IDS, forming what is called an Intrusion Detection and Response System (IDRS), or independently work along the IDS.
In the literature, generated responses are typically classified as either passive or active. Passive responses, consisting of the generation of alarms and reports, are not suitable for MANET environments where each node should react on its own since no form of centralized administration exists and nodes cannot rely on other nodes. Whereas, active responses consist of a predetermined set of actions (countermeasures) executed whenever an intrusion is detected to stop its spread and to locate and deter malicious node(s). Active responses may include: interrupting all communications with the intruder, discarding the intruder from the network, or executing some corrective actions.
It would be inadequate if such responses were applied to all types of intrusions in a fixed manner mainly if coupled with a high false positive alarm rate. For instance, this would

result in discarding some innocent nodes and may lead to the disruption of some network functionalities like losing connectivity, congestion, and an increased network latency. This created a need for the development of more corrective and adaptive response systems.

Since not all intrusions are predictable, some damage might be experienced before these intrusions can be detected and completely removed. For that, even if the implications of intrusions could be minimized by adaptive IDRSs, altered or deleted data is yet to be recovered. In addition, interrupted network services or connections should be brought back to function in a timely manner. For that, the network should be designed so that to survive such situations and to be able to autonomously heal any potential damages. This has led to the emergence of the so-called self-healing techniques as essential complementary techniques to achieve truly autonomous survivable networks.

Accordingly, a new highly dependable IDRS is presented in this thesis. The proposed IDRS is lightweight, autonomous, fault-tolerant, and allows a timely generation of adaptive responses. A self-healing mechanism is also integrated to attain more dependability for the supervised network.

1.2 Contributions

This thesis focuses on intrusion detection and response in MANET. The salient contributions of this thesis can be summarized in the following:

- A security-oriented literature review about MANET, multi-agent systems, and fault tolerant and survivable systems, with a great emphasis on intrusion detection and response, is presented.

- The introduction of a new lightweight agent-based IDS. The proposed IDS architecture is based on: (a) the distribution which is achieved through the implementation of a local intrusion detection system on each network node, and (b) the cooperation that is guaranteed by mobile agents.

- The extension of the proposed IDS to include an active response module. The resulting IDRS has the ability to autonomously generate adaptive responses based on both intrusions' severity level and their distribution over time. The notions of severity degree, severity index, $E^j_{min} and\ E^j_{max}$, the cumulative damage and cumulative severity degree, simple and severe responses are introduced to that end.

- The reinforcement of the proposed IDRS' fault tolerance through the integration of a novel dynamic replication mechanism.

- A recovery-oriented approach is proposed to enable the supervised network to heal itself of those potentially caused faults and damages.
- The thesis also highlights the effects of fixed response approaches on network connectivity and initiates to solving some issues related to network partitioning and remerging. The notion of DHB (Detection History Base) is introduced to help in maintaining consistency among the different IDRSs for better cooperation.
- Evaluation of the performance of the proposed IDRS.

1.3 Thesis outline

The remainder of the thesis is organized as follows:

Chapter 2 briefly introduces MANETs, their main vulnerabilities, and highlights some of the proposed solutions to cope with their security-related issues.

Chapter 3 presents some important concepts related to our work. The vulnerabilities of the mobile ad hoc networks and the proliferation of intrusions and thereby the need for survivability have been widely studied in the literature. For instance, there has been considerable research work in the fields of network monitoring, intrusion detection and response, fault-tolerant systems, self-healing and survivable networks. A thorough discussion of some interesting works in these areas is included to highlight the marking achievements and the remaining open issues.

Chapter 4 is devoted to the description of the proposed agent-based IDS, MASID: its general architecture, constituent agents and their roles and interactions, and intrusion detection algorithms.

Chapter 5 introduces the proposed intrusion response framework. It introduces new notions like the cumulative damage and the cumulative severity degree to reflect intrusions' progress over time; $E^j_{min} and\ E^j_{max}$, and the severity index to accurately assess intrusions' severity. Severe and simple response algorithms are also presented.

Chapter 6 encompasses two main parts. The first part illustrates how to improve the reliability of MASID to attain a highly available IDRS. In this respect, a novel dynamic replication framework is presented.
The second part of chapter 6 extends the proposed IDRS with self-healing features. Therefore, a recovery-oriented approach is proposed to improve the reliability and consistency of the network, so as to enable it to heal itself of faults and to better survive malicious attacks.

In both chapter 5 and chapter 6, the performance of the proposed approaches is evaluated through simulations with different network and attack scenarios.

chapter 7 concludes the thesis with a summary and discussion of the presented work. Some concluding remarks and future work directions are also presented in this chapter.

Part I

State of the Art

Chapter 2

Mobile Ad hoc NETworks

2.1 Introduction

As social beings, humans used the different available means to convey the message to their correspondent, and thereby to communicate. Through the years, they were always looking for more adequate methods of communication (hand gestures, smoke signals, written documents, telegraph, fax, etc.) and a new era of human communication started with the recent technological advancements and the appearance of computer networks.

A computer network is a communication system intended to connect various equipments including computers, printers, and other hardware devices, thus making it possible to share computer resources. Computer networks evolved relentlessly to span from central processing systems (where some passive terminals were connected to a central computer (Mainframe) performing all the tasks) to a world of untethered connectivity.

One interesting form of these networks is Mobile Ad hoc NETworks (MANETs), in which we are particularly interested in this thesis. This introductory chapter provides an overview of these networks, their main features, and their security-related issues.

2.2 Overview

A mobile ad hoc network is a self-configuring, self-organizing network that consists of a collection of mobile nodes that communicate with each other via wireless links without the help of any pre-existing infrastructure. Each node can function both as a router and as a host. In other words, a node can communicate directly with another node if they are within the transmission range of each other, otherwise, intermediate nodes will be involved to relay the messages. Figure 2.1 illustrates the general architecture of a MANET.

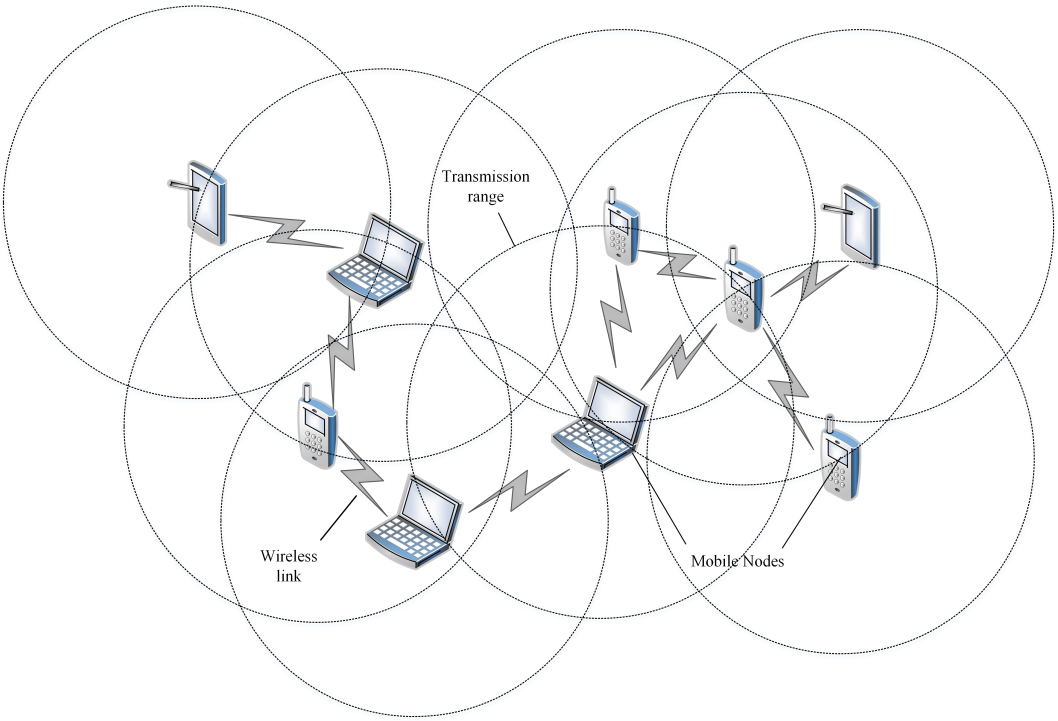


Figure 2.1: General architecture of a MANET

A mobile node can be among others: a laptop, smartphone, Personal Digital Assistant (PDA), or wearable devices. Such devices are getting smaller in size (which is favourable to mobility) and greater in their capabilities like storage, battery life, and processing ability (more complex applications can be run on these devices). With these phenomenal advancements in the fields of communication devices and mobile computing, MANETs are being easily deployed in different domains. Example application domains are: tactical networks, search and rescue operations, disaster relief operations [55], Personal Area Networks, entertainment and educational applications [104], environment monitoring [3], and commercial applications.

MANETs exhibit several characteristics among which we cite:

- Easy, fast, and cost-effective deployment.
- Resource limitations: CPU, storage capacity, battery life, and bandwidth.
- Due to nodes' mobility, the network's topology experiences frequent and unpredictable changes.
- Nodes are free to join or leave the network at any moment.

- Decentralization, Self-organization, and self-management: there is no fixed infrastructure or centralized control system.

2.3 Routing in MANET

In MANET, nodes can either communicate directly if they are within the radio range of each other or communicate through other nodes otherwise. In this latter case, one or more nodes are chosen based on specific criteria to relay the message between its source and the intended destination. This process is referred to as routing.

Several routing protocols were proposed for MANET. Based on routing information source, routing protocols for MANET fall into three main categories: proactive, reactive, and hybrid protocols.

2.3.1 Proactive Routing Protocols

Also called table-driven routing protocols, refer to the class of routing protocols where every node maintains a routing table containing routing information about every other node as long as this latter belongs to the network. Nodes mobility leads to frequent changes in the network's topology and eventually to the change of existing routes. This change manifests in different forms: the breakage of an existing route, an update in an existing route, and the establishment of a new route. With such frequent changes in routes, routing tables must be updated to maintain consistent routing information. OLSR [19] and DSDV [74] are examples of proactive routing protocols. Following is a brief description of the DSDV routing protocol.

DSDV (Destination Sequenced Distance Vector)
DSDV [74] is an adaptation of RIP (Routing Information Protocol [44]) to Ad hoc networks. In a DSDV-based network and similarly to other proactive routing protocols, every node maintains a routing table. In order to fix looping problems, DSDV extends the routing tables with the notion of sequence numbers to characterize the freshness of routes in the routing table. In order to maintain consistency of the routing tables, routing updates are periodically propagated throughout the network. DSDV employs two types of updates: full dump and incremental. For a full dump update, a node sends its entire routing table to its neighbours. In an incremental update, however, only routing table entries that have changed since the last full dump update are transmitted.
Along with the routing table, the source sequence number is included in the update message. Update and thereby, forwarding decisions are made based on the values of these

sequence numbers. Routes with the greatest sequence numbers are always chosen for update. In case of receiving multiple update packets where sequence numbers are equal, the one with the shortest route is retained.
A small change in the network's topology like a single link break entails the broadcast of updates to all network nodes leading to a considerable overhead witch calls the protocol's scalability into question.

2.3.2 Reactive Routing Protocols

In a reactive (also called on-demand) routing protocol, routes are created only when needed. For instance, it suffices for a node that needs to communicate with another node in the network to broadcast a request to establish a new route. On reception of the request, the destination node responds with a message that goes back to the originator of the request. Example reactive routing protocols are DSR (Dynamic Source Routing [52]) and AODV (Ad hoc On-Demand Distance Vector [76]). In this thesis, we are particularly interested in AODV. Following is a detailed description of this protocol.

Ad hoc On-Demand Distance Vector (AODV)
AODV [76][75] is a reactive routing protocol that enables multi-hop, self-starting and dynamic routing in MANET. In networks with large number of mobile nodes, AODV is very efficient as it relies on dynamically establishing route table entries at intermediate nodes. Also, AODV provides loop-free routes thanks to the concept of destination sequence number borrowed from DSDV. Sequence numbers serve as time stamps and allow nodes to compare how fresh the information they have for other nodes in the network.
In AODV, routes between nodes are created only when they are requested by source nodes. AODV supports both unicast and multicast routing. AODV nodes use four types of messages for their communications, namely, HELLO messages, Route REQuest (RREQ), Route REPly (RREP), and Route ERRor (RERR) messages. RREQ and RREP messages are used for route discovery, while RERR and HELLO messages are used for route maintenance.
Whenever a source node needs to send a packet to a destination node, it, first, checks its routing table to determine if it already has a route to the destination node. If it is the case, then the packet is forwarded to the next hop node. Otherwise, it initiates a route discovery process. Figure 2.2 illustrates a sample scenario of the route discovery process in an AODV-based MANET.

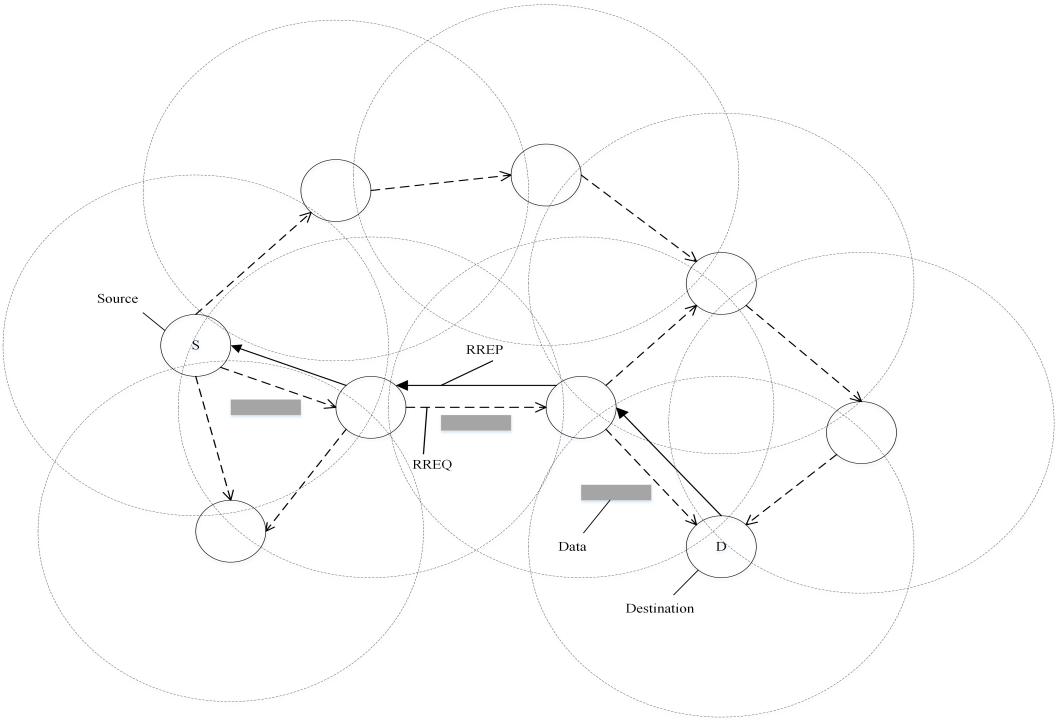


Figure 2.2: Routing in an AODV-based MANET

Route discovery process starts by a node that wants to communicate with another node for which there is no routing information in its routing table. For that, it broadcasts a RREQ packet containing the following information to its neighbour nodes:

- RREQ ID,
- Destination IP Address,
- Source IP Address,
- Destination sequence number: represents the latest sequence number received in the past by the source for any route towards the destination,
- Source Sequence Number: represents the latest sequence number to be used in the route entry pointing towards the source of RREQ,
- Hop Count: representing the distance in hops from the source to destination.

Upon receiving a RREQ, the recipient node checks if it has already received a RREQ with the same information within the Path Discovery Time. If it is the case, it discards the newly received RREQ. Else, it either responds by sending a RREP packet back to the source node or rebroadcasts the RREQ to its own neighbours after incrementing the

hop count field by one. This process will continue until the packet is received by the destination node or an intermediate node that has a fresh route entry for the destination. If a node receives more than one RREP, it updates its routing information and propagates the RREP only if the RREP contains either a greater destination sequence number than the one in the previous RREP, or the same destination sequence number with a smaller hop count.

AODV presents several security vulnerabilities that can be exploited by malicious nodes to launch their attacks. Examples of possible exploits are: modification of sequence numbers and/or hop counts, spoofing, and tunneling.

2.3.3 Hybrid routing protocols

Hybrid routing represents a combination of both proactive and reactive routing. Proactive routing is used within a small perimeter area around the source node while reactive routing is reserved to route data beyond this perimeter. An interesting example of hybrid routing protocols is ZRP (Zone Routing Protocol).
ZRP [40] is especially suitable for large networks and those with diverse mobility patterns. The protocol uses the notion of routing zone to refer to the local area surrounding a node. The perimeter of a zone is referred to as the Zone radius. It represents the maximum distance in hops between a node and its zone nodes. Based on routing zones, ZRP defines two different routing protocols: (i) IARP (InterA-zone Routing Protocol [39]) to proactively maintain routes within a zone and (ii) IERP (IntEr-zone Routing Protocol [38]) to be used for reactive routing between zones.

A more comprehensive listing of MANET routing protocols is presented in figure 2.3.

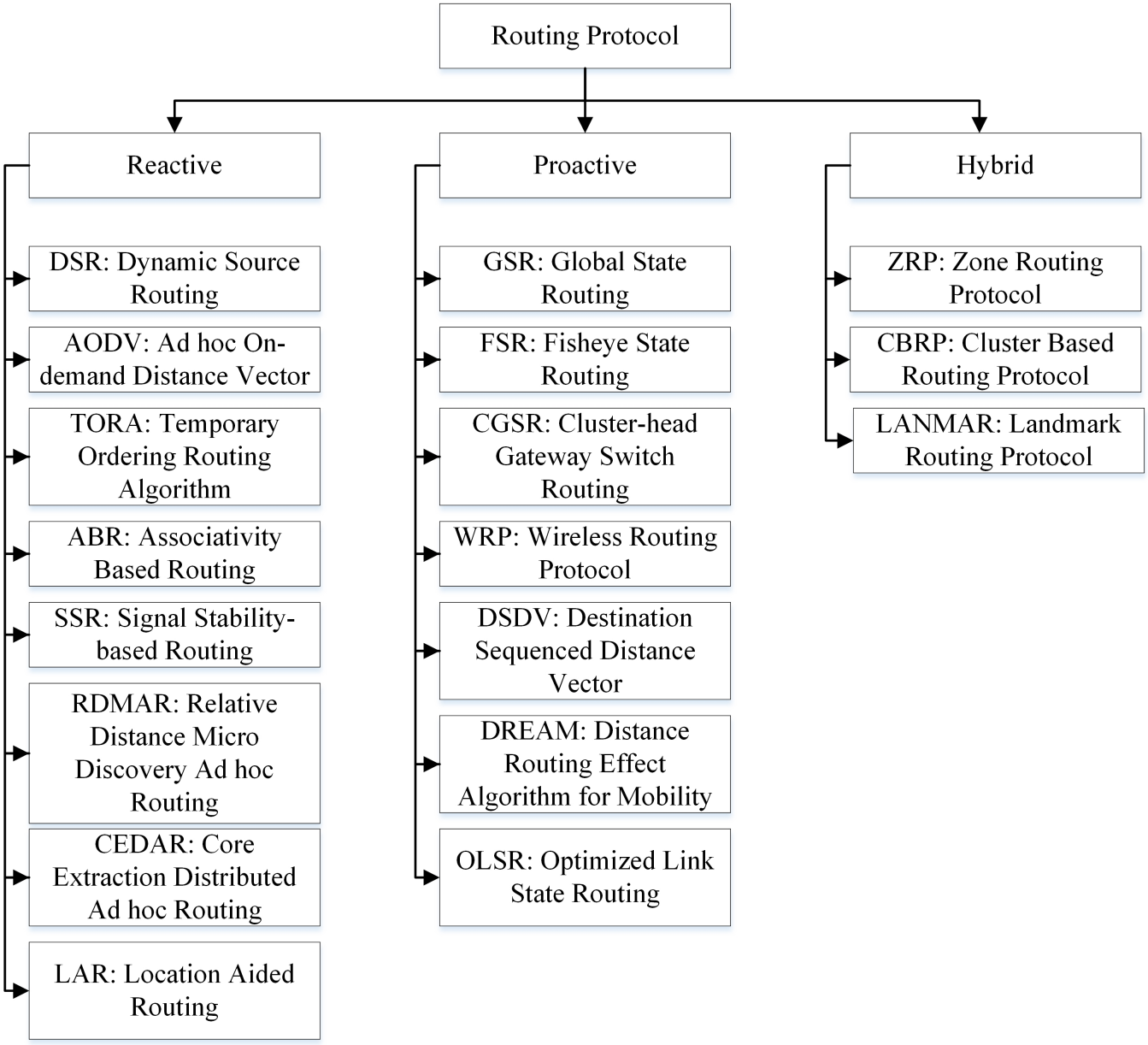


Figure 2.3: MANET routing protocols

2.4 Security in MANET

Regardless of the nature of the adopted routing protocol, communication in MANET is based on nodes' cooperation. This means that nodes' non-cooperation can obstruct the routing process and thereby, leading to the failure of the network's mission. A node's non-cooperation can be the result of its failure or, more seriously, the result of its intentional misbehaviour. The following subsections discuss both security problems and solutions in MANET.

2.4.1 Security Problems in MANET

After presenting the quintessence of MANETs in the previous section, this section highlights the deficiencies and the issues that these networks may raise in terms of security. This section starts with a description of MANETs' main sources of vulnerability and ends up with the description of how these vulnerabilities can be exploited by malicious entities to achieve their goals.

2.4.1.1. Preliminaries

Security threats: are tools, techniques, or methods that can cause unwanted incidents, and potentially result in damaging the network.

Vulnerability: a hardware or software weakness of the network that can be exploited by one or more threats.

Security risk: a risk is the effect of uncertainty on objectives [1]. In terms of network security, a risk represents the effect of uncertainty on network security goals (cf. Section 2.4.2). This uncertainty comes from the potential that threats will exploit network vulnerabilities to cause damage. An attempt to do so represents an attack.

Attack: an attack is an attempt to destroy, expose, alter, disable, steal or gain unauthorized access to or make unauthorized use [1] of network resources. A successful attack is referred to as an intrusion (cf. Section 3.2)

2.4.1.2. Vulnerabilities of MANETs

The inherited characteristics of MANETs presented these networks with several advantages but at the same time, most of these features create a source of vulnerability for these networks.

Absence of a fixed infrastructure: the absence of a fixed infrastructure and centralized control impairs the use of security mechanisms that involve third party entities like multi-party non-repudiation protocols [57] and Certification Authority [2].

Dynamic topology: due to the lack of a fixed infrastructure, network topology is determined by nodes' positions. In MANET, nodes are frequently moving which entails frequent changes in the topology and thereby a change in the established routes. From one side, this creates an opportunity for malicious nodes to choose perfect positions to

launch their attacks or to introduce themselves in newly established routes. From the other side, roaming nodes risk to be isolated in separate partitions and become easily captured by malicious nodes.

Absence of clear boundaries: The openness of the wireless medium permits a flexible extension of the network's size. This, however, creates a serious security gap for MANETs compared to their wired counterparts. While MANET nodes lack external protection that firewalls and network gateways can provide, malicious nodes can easily join the network and interfere with its functions.

Cooperative communication: Nodes reliance on other nodes to route their packets helps malicious nodes to launch attacks either against the routing process by fabricating routes or against routed data by altering or dropping packets once a member of an established route.

Network and node resource constraints: The limitations of resources like nodes' energy and storage capacity as well as the bandwidth limitation facilitate the compromise of their availability. The limitation of nodes processing power impairs the deployment of complicated security solutions. Also, considering bandwidth limitation, these solutions should not involve heavy data exchange between cooperating network nodes.

Unreliability of wireless links: communication through radio links is prone to interception and eavesdropping. It suffices for a malicious node to be within the transmission range of a node to be able to eavesdrop or intercept its communications. Also, DoS attacks can be easily injected into the network.

2.4.1.3. Taxonomy of MANET Attacks

The vulnerabilities entailed by the inherited nature of MANETs together with the lack of a clear line of defence expose these networks to a wide range of security threats. Several classifications and taxonomies of attacks targeting MANETs were proposed in the literature [85, 41, 108, 107]. The commonest taxonomy classifies them as either passive or active attacks as shown in Table 2.1.

A passive attack is usually launched by listening to the channel with the bad intention of retrieving critical information about the nodes and network traffic (e.g., IP addresses, location of nodes, etc.). Examples of passive attacks include eavesdropping, traffic analysis and monitoring attacks that compromise the privacy of node's communications. Location disclosure attacks are also passive attacks in which the privacy of the node itself is

Network layer	MANET attacks	
	Active	Passive
Application	Data corruption, repudiation	
Transport	Session hijacking, side-along jacking	
Network	Fabrication, modification, wormhole, blackhole, selective forwarding, resource consumption, Sybil, Routing table poinoning	Traffic analysis and monitoring, location disclosure attacks
Data link	WEP weakness exploitation (message privacy and integrity attacks, probabilistic cipher key recovery attacks)	
Physical	Jamming (pulse, random noise), interception	Eavesdropping
Multi-layer	Denial of service, impersonation, replay, man in the middle attacks	

Table 2.1: Taxonomy of MANET Attacks

compromised by disclosing its current location in the network. Such attacks are usually conducted with the broader objective of revealing the network's structure. Passive attacks do not directly affect the network's operation which makes their detection a truly difficult task. These attacks can be the initiating phase for some active attacks.

Contrary to passive attacks which are mainly characterized by their non-disruptive nature, an active attack is, generally, performed by a malicious node with the deliberate intention of interrupting the functionality of one or more specific or random nodes or the network itself. Active attacks can have several forms: modification, fabrication, and impersonation attacks.

A detailed description of some of these attacks is presented in the following section.

2.4.1.4. Example attacks against MANET

This section provides a brief description of some common attacks against MANETs. A malicious node can act alone or in conjunction with other malicious nodes. Techniques used by these nodes fall into one of following categories or are combinations of one or more categories.

A. Modification attacks: A modification attack is typically launched by a malicious node with the deliberate intention of redirecting network traffic, by altering some fields of the routed packets. A subtle example of modification attacks is the blackhole attack where some packet fields like the sequence number and hop count are modified.

B. Fabrication attacks: Instead of modifying or interrupting the existing routing packets in the network, malicious nodes can fabricate their own packets to cause chaos in network operations. They can launch message fabrication attacks by injecting fake routing messages such as routing updates and route error messages into the network, thereby, resulting in some attacks such as falsifying route error message, route cache poisoning, routing table overflow, and sleep deprivation attacks (also known as resource consumption attacks). The main purpose of fabrication attacks is to drain off limited resources of the other MANET nodes, such as battery power and network bandwidth.

C. Impersonation attacks: Impersonation or spoofing [99, 7] refers to the case where a malicious node, intentionally, misrepresents its identity in the network. Thus, an impersonation attack occurs when a malicious node uses for example the IP or MAC address of another node in outgoing packets, thereby, disrupting the normal functionality of the network by either receiving packets meant for other nodes or worse yet, completely isolating some of the network nodes. A well-known example of impersonation attacks is the packet misrouting attack.

D. Deletion attacks: A deletion attack consists of the intentional deletion of routed packets either by dropping them or by not relaying them with the bad intention of disrupting data packets being sent to the destination node or to simply obstruct the route discovery process. A well-known example of deletion attacks is the blackhole attack.

Blackhole [69, 84] is one of the active attacks against MANET. In this attack, a malicious node falsely replies to route requests without having an active route to the destination. It exploits routing protocols such as AODV to advertise itself as having a valid and good path to the destination node. The blackhole node first tries to gain a position in active routes. Then, it can choose either to drop all the packets to perform a denial of service attack or to selectively drop packets as a manifestation of the grayhole attack [72, 102]. Figure 2.4 illustrates a scenario where the blackhole node is dropping the received data packets whereas Figure 2.5 illustrates a grayhole node selectively dropping the received packets.

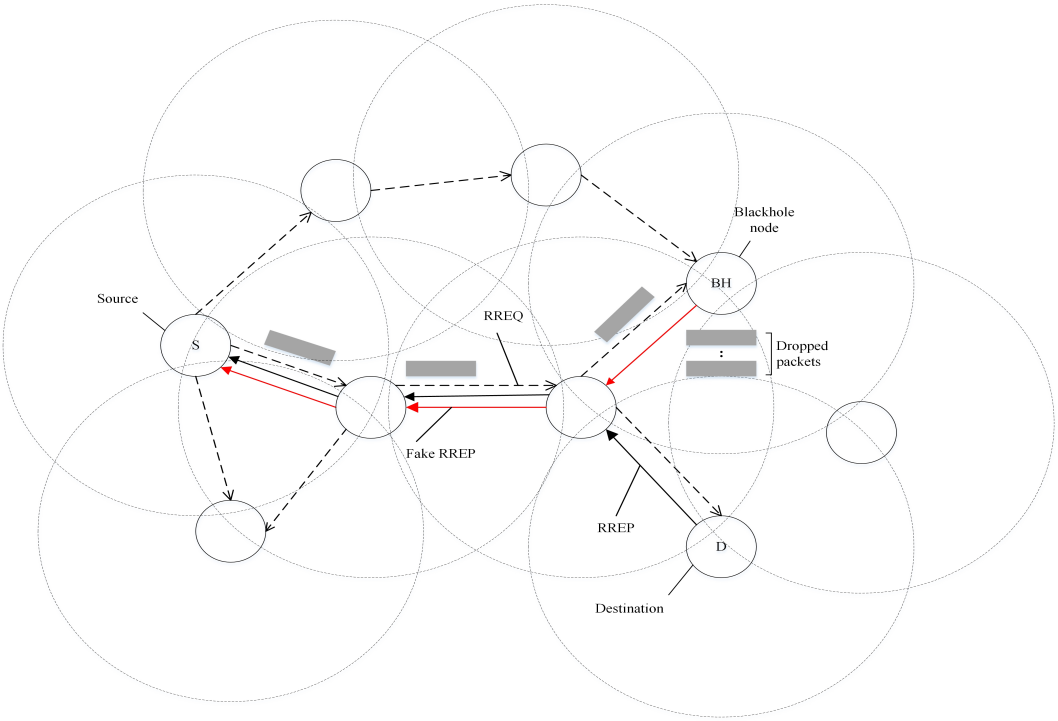


Figure 2.4: Blackhole attack in an AODV-based MANET

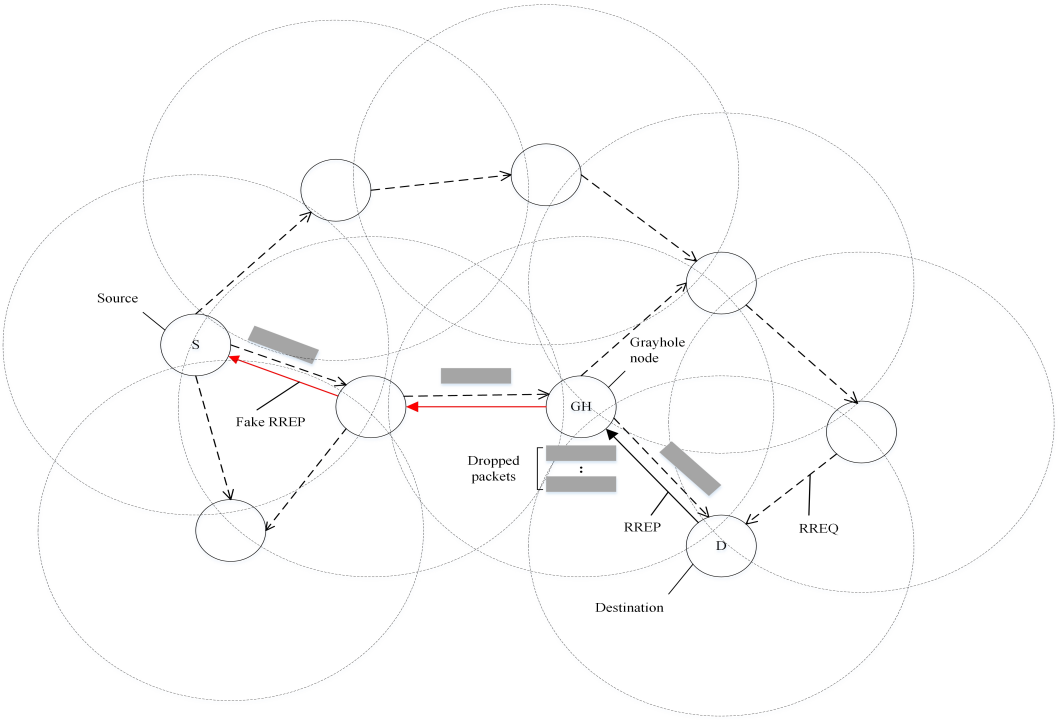


Figure 2.5: Grayhole attack in an AODV-based MANET

E. Selfish behaviour attacks: The selfish behaviour of a node [82, 93] can be carried out by refraining from forwarding data or control packets. Thus, a selfish behaviour attack refers to situations where a selfish node does not perform the expected network functions. More specifically, the selfish node, intentionally, does not cooperate in the routing process in the hope of saving its resources such as battery power. Such an attack, although not necessarily intended to cause any damage, can lead to serious disruptions in network communications such as high route discovery delays and, sometimes, the isolation of of one or more nodes if the only connection to the rest of the network is through the selfish node. Example attacks are packet mistreatment and energy consumption attacks. Figure 2.6 illustrates a scenario where the selfish node causes the draining off of other nodes' resources while preserving its own.

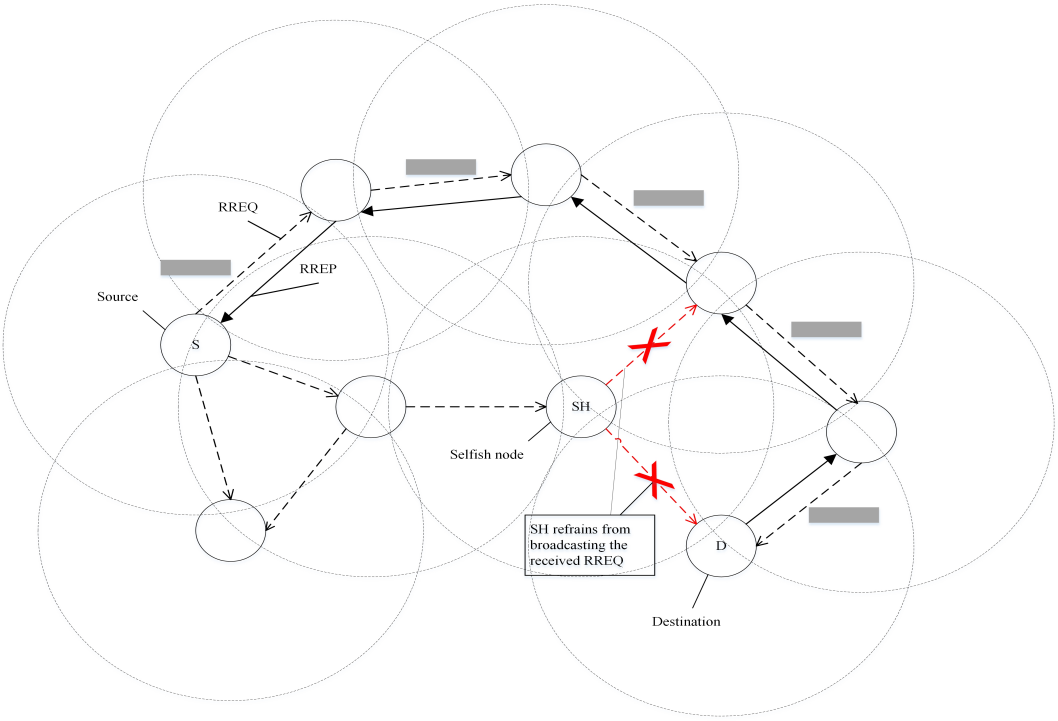


Figure 2.6: Selfish behaviour attack in an AODV-based MANET

2.4.2 Security Solutions for MANET

The use of MANETs in critical applications like those pertaining to medical and military domains renders them a target for attackers and boosted the need to secure them. This section discusses the different security requirements as well as some of the proposed security solutions for MANET.

2.4.2.1 Security Goals and Requirements

The International Organization for Standardization (ISO) specifies the following requirements as the properties to be preserved to guarantee information security:

Confidentiality: it is the property that information is not made available or disclosed to unauthorized individuals, entities, or processes [1].

Integrity: it is the property of accuracy and completeness [1].

Availability: the property of being accessible and usable on demand by an authorized entity [1].

Authenticity: authenticity is the property that an entity is what it claims to be [1].

Non-repudiation: it represents the ability to prove the occurrence of a claimed event or action and its originating entities [1].

Other properties that can also be involved are: accountability and reliability [1].

Table 2.2 redefines these requirements from a computer networks' security perspective. It also provides an overview of their main goals, key methods and techniques to ensure them, example attacks that can violate them, and highlights some incurred risks in case of their non-fulfilment.

Let A and B be two Communicating Entities (CE), and let M be a message communicated from A to B.

Security goals	Description	Scope	Example of techniques	Example of attacks	Incurred risk
Confidentiality	Ensuring confidentiality means that M is only accessible to A and B. This property assures data privacy.	Data	Cryptographic techniques [23], Access control	eavesdropping [61]	Data disclosure to non-authorized entities

Integrity	Ensuring integrity means that B is able to detect any unauthorized alterations to M. This property ensures that data is accurate and trustworthy.	Data	Message Authentication Codes [9], cryptographic checksums [20]	Man-in-the-middle attacks (Packet injection, alteration, replay) [25]	Unauthorized manipulation of data
Availability	Ensuring availability means that services and data are available (accessible and usable) to authorized entities in a timely manner.	Data - Service	Fault-tolerance, Redundancy [24], Scheduling [13]	Jamming attacks [103], Flooding attacks [84], Sleep deprivation	Network performance degradation (Unavailability of services or resources)
Authenticity	- Authentication of Communicating Entities means that A can verify the identity of B and vice versa. - Authentication of data means that B can verify that M was truly generated by A.	CE - Data	Digital signatures [53]	impersonation attacks [7]	Unauthorized access to resources and sensitive information
Non-repudiation	Ensuring non-repudiation means that A cannot deny having sent the message M. This property helps in detecting compromised nodes.	CE	Digital signatures	Repudiation attacks	Inability to prove authorship of malicious activities.

Table 2.2: Security Requirements

2.4.2.2. Existing Security Solutions

Providing security in MANET is a prime concern and an impeding issue that should be addressed. Since conventional security measures such as authentication and firewalls are not sufficient or non-applicable to these networks, plenty of other solutions were proposed in the literature to solve security issues in MANET. In the following we briefly review some interesting works in this area.

A. Prevention techniques

Prevention techniques are mainly useful to reduce the possibility of attacks happening but due to the inherent MANET constraints (eg. limited resources, absence of centralized management) prevention techniques used for wired networks cannot be directly applied to MANETs.

Conventional prevention mechanisms like authentication and encryption are based on cryptographic concepts. Because asymmetric cryptography-based approaches [5, 100] are highly resource consuming, researchers tend to use symmetric cryptography to develop preventive techniques for MANET [80]. In both cases private and/or public keys are needed. However, because of the absence of any infrastructure or central authority that can handle key management in MANET, key management should be handled by the network nodes in other ways. Different key management approaches have been proposed in the literature. For instance, key management can be performed in a distributed manner [65], based on clustering [37, 27, 29], based on identity [35, 113, 111], or based on certificate chaining [51, 21], etc.

B. Secure routing

Secure routing protocols are routing protocols that have security as one of their goals. Such protocols are usually built by extending existing routing protocols with security features. SAODV (Secure AODV [112]), SEAD (Secure Efficient Adhoc Distance vector [48]), and SRP (Secure Routing Protocol [70]) are examples of secure extensions of AODV, DSDV, and DSR, respectively.

For instance, SAODV relies on the assumption that every node has certified public-keys of all network nodes while SEAD-based nodes use authentication of the routing update packets to prevent modification attacks. This is achieved through the use of public-key signed hash chains.

Other routing protocols like ARAN (Authenticated Routing for Ad hoc Networks [88]) were originally developed with security features incorporated. ARAN uses asymmetric

cryptography to guarantee end-to-end authentication, non-repudiation, and message integrity, and thereby to thwart attacks by third parties and peers. ARAN-based nodes use certificates generated by a trusted certificate server to authenticate themselves to other nodes. The authenticated route discovery and the authenticated route setup phases are then used to securely establish a route between the source and the intended destination.

Instead of using a trusted certificate server which is not practical for MANET environments, SRP assumes the existence of a security association between the source and the destination nodes. Thus, every source and destination pair of nodes shares a secret key and uses message authentication code to check the integrity of the packet and to authenticate its sender.

Trust-based routing is also used as a means to achieve reliable and secure routing in MANET. Trust-based routing consists of the measuring of nodes' trust. Nodes' trust measurement can have one of two different forms: centralized and distributed. In centralized trust models [106, 71], nodes rely on a Trust Agent to evaluate the trust of other nodes. Distributed trust models [78, 96, 90, 101], however, rely on trust evaluations and recommendations from other nodes.

C. Intrusion detection and response

Most of the proposed solutions for attack prevention and to build secure routing protocols are attack-oriented. This means that these solutions are designed to deal with some specific attacks. Such solutions can perform efficiently against these specific attacks but not if faced with insider attacks or unknown threats. In addition, the use of cryptography (mainly asymmetric cryptography) is resource consuming and secure routing protocols create extra overhead to achieve their goals. To cope with these problems intrusion detection and response systems are used as a second line of defence. More details about the deployment of IDRSs in MANET are presented in the following chapter.

2.5 Conclusion

This chapter introduced MANETs: their main characteristics, applications, and routing protocols. The chapter also highlighted MANET's main vulnerabilities and described some potential security threats. To conclude, the chapter surveyed some of the proposed solutions to cope with these issues. Among these solutions, IDRSs emerged as a second line of defence to cope with preventive solutions' vulnerability to insider attacks and secure routing techniques' ineffectiveness against unknown attacks. The following chapter focuses on intrusion detection and response in MANET.

Chapter 3

Intrusion detection, intrusion response, and survivability in MANET

3.1 Introduction

We have seen in the previous chapter that the use of MANETs is limited by the several security-related issues induced by they raise due to their special nature and to the numerous constraints they present. Although many research works have been devoted to develop security mechanisms for MANETs, but still the optimal and efficient security solution not found. Some researchers developed preventive approaches to guarantee security while many others prefer the use of secure routing protocols. Also, there has been, recently, a great tendency to develop intrusion detection systems (IDS) specifically designed to fit MANET requirements in terms of both security and constraints.

This chapter presents some basic notions and a brief survey about the recent advancements in the areas of intrusion detection, intrusion response, fault-tolerant systems, self-healing and survivable networks.

3.2 Intrusion detection in MANET

3.2.1 Preliminaries

This section introduces some basic concepts and terminology related to the field of intrusion detection.

a. Intrusion: an intrusion is defined as any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource [49]. For instance, Events like

trying to break into a system from the Internet using software exploits or trying to gain higher privileges on a system are representative events that will be recognized as an intrusion. Several intrusion classifications were proposed in the literature [85, 107]. Arguably, intrusions can be grouped in two broad classes as follows:

- Known intrusions: these intrusions are well known attacks that exploit known vulnerabilities of a target system (host or network).
- Unknown intrusions (Anomalies): these intrusions represent deviations from the normal behaviour of a target system.

b. Intruder: entities that cause or initiate intrusive activities are called intruders. Intruders can be either internal having an authorized access to the target system or external without any authorized access, but generally exploiting compromised systems or nodes to get through.

c. Intrusion detection: intrusion detection is the process of monitoring, tracing, and analysing events of computer systems or networks and the subsequent generation of alarms upon detection of intrusive activities. Its main goal is to uncover any set of actions that attempt to compromise the integrity, confidentiality or availability of a resource [49]. Computer intrusion detection started in 1972 with a paper by J. Anderson [6] identifying the need for what would evolve into today's intrusion detection systems.

d. Intrusion Detection Systems (IDSs): IDSs are important components of the security mechanisms in computer network systems [16]. The goal of an IDS is not to prevent an attack, but to detect it as quickly as possible. Deployed as a second line of defence, IDSs automate the process of monitoring and analysing events of computer systems or networks in the search for security problems. Typically, an IDS comprises three main components: (i) a sensor, through which the IDS monitors and collects data from a target system (host or network), (ii) an analysis or detection unit, responsible for processing and correlating the gathered information, and (iii) a response unit that processes alerts generated by the detection unit and initiates responses if necessary. Figure 3.1 illustrates the major interactions among these components.
IDSs can be categorized as network-based or host-based IDSs depending on the target environment for detection. Network-based IDSs (NIDSs) collect input data by monitoring network traffic. Whereas, Host-based IDSs (HIDSs) rely on events collected by the hosts they monitor.

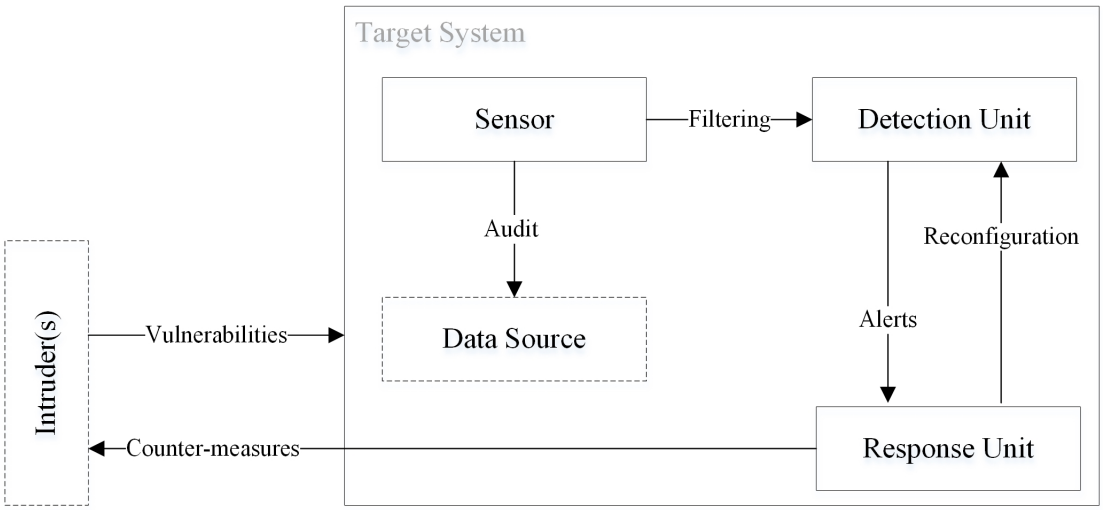


Figure 3.1: General architecture of an IDS

3.2.2 IDS Evaluation

The commonly used metrics for IDS evaluation are summarized in Table 3.1.

Metric	**Formula**	**Description**
Accuracy	$(TP+TN)/(TP+TN+FP+FN)$	The probability that the IDS can correctly predict normal profiles and attacks
Precision	$TP/(TP+FP)$	the proportion of predicted attack cases that are real attacks
Specificity	$TN/(TN+FP)$	The proportion of normal profiles that are successfully identified as normal profiles
Detection rate (Recall)	$TP/(TP+FN)$	The proportion of attacks that are successfully identified as attack cases
False positive alarm rate	$FP/(FP+TN)$	The proportion of learned normal profiles that are considered as attacks

False negative alarm rate	$FN/(FN+TP)$	The proportion of attacks that are not successfully detected

Table 3.1: IDS evaluation metrics

Where values of TP, TN, FP and FN are counted with respect to the relation between the predicted and actual classes of the audited profiles as illustrates Table 3.2.

		Predicted class	
		Normal profile	Attack
Actual class	Normal profile	True negative (TN)	False positive (FP)
	Attack	False negative (FN)	True positive (TP)

Table 3.2: IDS confusion matrix

3.2.3 Intrusion Detection Methods

Detection methods specify the mechanism adopted by the IDS to detect intrusions i.e., data analysis methods used by the IDS while looking for traces of intrusions among collected data. There are two main intrusion detection methods, namely anomaly detection and misuse detection (also referred to as signature-based).

3.2.3.1 Anomaly detection

Anomaly detection techniques model normal behaviour and compare it to observed data to uncover anomalous patterns of behaviour. Anomaly detection is a two-phase process: the training phase and the monitoring phase. The first phase can be performed either offline or online, and either automatically or manually [34]. It consists of the extraction of the main features that characterize the target system's normal behaviour to build a model of the normal profiles dataset. Monitoring is an online phase that follows the training phase to detect deviations in audited data from the normal profiles. In the literature, several techniques have been used to develop anomaly-based detection techniques such as Data mining techniques, hard and soft computing-based techniques, and statistical techniques.

In [26], an association algorithm (Fast Apriori Algorithm) is used to extract necessary traffic features and to collect data streams from various network layers (physical, MAC, and network layers). Subsequently, the local detection module uses the fixed width clustering algorithm to analyse collected data for signs of anomalies. If any detection rule deviates beyond the anomaly threshold, the alert management agent will be initiated.

An artificial immune system based IDS for MANET was proposed in [58]. Here, each node was equipped with two agents: A mobile agent and a master agent. The mobile agent is in charge of gathering information related to bandwidth, packet delivery rate and delay from neighbouring nodes. Collected information is reported to the master agent residing on the mobile agent's home node. This latter uses it to run the artificial immune system to generate and/or update the normal profiles patterns. Upon receiving new packets, a node calculates parameters like packet delivery rate and delay. If the calculated parameters match with the patterns generated by the master agent, then the connection is considered as valid. Otherwise, an alert is generated.

3.2.3.2 Misuse detection

Misuse detection techniques deal with attack behaviour i.e., they compare audited data to known attack patterns. A misuse-based IDS is always equipped with a database containing known intrusions' signatures. A match between audited data and a pattern in the signatures database triggers an intrusion alarm. A variety of techniques can be used to implement misuse detection techniques such as expert systems, pattern matching, and evolutionary computation.

In [94], the authors proposed a misuse-based approach for detecting blackhole attacks using a set of collaborating bayesian watchdogs. They defined α and β as the numeric representation of a node's reputation. Every node runs a watchdog and collects the reputation information for its neighbours to obtain the values of α and β for every neighbour. Periodically, every watchdog shares these data with its neighbours for use as second hand information. Once received, the detection module uses this information along with the locally collected information to estimate the relationship between α and β. If it exceeds a predefined tolerance level, the corresponding node is declared as a misbehaving node.

In [28], an analytical computational framework based on danger theory is implemented. The detection process is preceded by a training phase, during which normal and dangerous signatures are specified. A danger signal is then activated upon any match with the dangerous signatures.

3.2.3.3 Hybrid detection

A hybrid intrusion detection that combines both anomaly and misuse detection can be considered as a third technique of detection. For instance, Farhan et al. [30] presented an IDS for ad hoc networks in which both anomaly detection and misuse-based detection methods were used. In particular, they exploited two anomaly detection techniques: Conformal Predictor K-Nearest Neighbour (CP-KNN) and Distance-based Outlier Detection (DOD). For the implementation of the misuse-based method they focused on three types of attacks which are resource consumption attack, dropping routing traffic attack and blackhole attack. The misuse based detection unit applies string matching to detect these attacks. It raises an alarm to the response module if any activities match an intrusion pattern. In case of an unknown intrusion, the signature generation unit extracts signatures of the detected anomalies and stores them in the attack signature database.
Similarly, Nadeem and Howarth [66] proposed a hybrid detection technique in which the chi-square test is first used for anomaly detection. Then, a rule-based approach is used to identify the attack. To finish, a manager node applies intruder identification rules that are specific to every known attack.

3.2.3.4 Discussion

Each of these techniques has some advantages over the other one, but at the same time, they present some serious drawbacks. For instance, misuse detection is effective in detecting known attacks but it is generally not able of detecting attacks that have not been previously defined. Even variations on known intrusions can be missed if the detection algorithm is not flexible enough. Unfortunately, this inability to detect unknown attacks leads to a high false negative alarm rate. On the other hand, anomaly detection allows the detection of new and unknown attacks since any deviation from what is considered normal is flagged as intrusive.
However, because not all the deviations from the normal behaviour are necessarily intrusive activities, a significant number of false positive alarms may be generated. In both cases, detection accuracy is largely dependent on the accuracy and completeness of the created attacks' signatures and normal profiles. However, since it is often difficult to perfectly create a model that covers all possible variations of the system's normal behaviour, the updating of normal profiles is needed for better accuracy. Signatures database also need to be frequently updated to include new attacks' signatures.

3.2.4 IDS Architectures in MANET

An IDS architecture specifies the mode in which the IDS operates i.e., the structure and organization of the different IDS agents. In MANET, IDSs are, generally, classified into four main classes (architectures), namely, stand-alone IDSs, distributed and cooperative IDSs, hierarchical IDSs, and agent-based IDSs.

3.2.4.1 Stand-alone IDSs

In this category of IDSs, the detection process is performed on each node, and there is no cooperation or data exchange between network nodes. A typical example of stand-alone IDSs is that of [18], termed CAIDS (Context Adaptive Intrusion Detection System). CAIDS is able to dynamically adapt to contextual factors at a given node such as the residual energy, potential security threats and traffic loading to accommodate and inspect new arriving packets. Through the use of an intelligent IDS controller, CAIDS selects optimal values to execute the intrusion detection plan for MANET under energy constraints. Here the main disadvantage is that the authors' main focus was to adapt the IDS to the different contextual factors of the network nodes neglecting the fact that the nature of MANET implies the cooperation of the different nodes of the network in order to get a global vision of what is happening in the network. The absence of such global vision might be the main source of network vulnerability to distributed attacks.

3.2.4.2 Distributed and Cooperative IDSs

MANETs are distributed by nature and require nodes' cooperation. In a distributed and cooperative IDS architecture, every node in the network must participate cooperatively in intrusion detection and response. In [30] the architecture and operation of a distributed and cooperative IDS were described. The proposed intrusion detection model consists of two major components: Gateway Intrusion Detection (GID) and Local Intrusion Detection (LID). GID comprises three components: Global Detection Module (GDM), Global Response Module (GRM) and Cooperation Module (CM). A gateway node can optimize energy use by scheduling only a subset of region members who will activate their monitoring sensor agents at one time. Other region members can minimize their energy consumption at the same time. LID is mainly divided into: Data Collection module (DCM), Pre-process Module (PM), Local Detection Module (LDM), and Local Response Module (LRM). The DCM collects audit data from various network sources and then passes it to the PM. PM selects informative features from all features set, and then pass these features to the LDM. The LDM analyses the collected local data using CP-KNN

and DOD classification algorithms, and identifies malicious nodes in the ad hoc network. The main advantage of this approach is the detection accuracy. However, it may cause the degradation of the network performance with the traffic exchanged between the different LID and GID.

3.2.4.3 Hierarchical IDSs

This architecture proposes using multi-layered network infrastructures where the network is divided into clusters. A cluster consists of a group of interconnected nodes whereby a node playing the role of the cluster head (CH) manages the other nodes referred to as cluster members (CM). The main idea behind this architecture is to create a hierarchy depending on nodes capabilities. A hierarchy can be either simple or multi-levelled. A simple hierarchy consists of only cluster members and their immediate cluster heads. A multi-levelled hierarchy, however, has a bottom level consisting of a simple hierarchy and higher levels with cluster heads of each level $(i + 1)$ playing the role of cluster members for cluster members of level i.

Some hierarchical IDSs involve some or all cluster members in the detection process while leaving the final decision to the CH. In [63], the authors introduced two intrusion detection algorithms, termed ADCLI (Algorithm for Detection in a CLIque) and ADCLU (Algorithm for Detection in a CLUster). Both algorithms are based on the collaboration of a group of nodes that are either directly connected (clique) or within a one-hop-route of each other (cluster). A voting mechanism is used to determine malicious nodes. Messages are passed between the nodes and depending on the messages received; these nodes determine the suspected nodes. These suspected nodes (votes) are eventually sent to the monitor node (the initiator of the detection algorithm). At the monitor node, suspected nodes that receive at least a minimum number of votes are detected as malicious nodes. Hence, the algorithms work in such a way that a group of nodes together make the decision, about the maliciousness of a node, which minimizes the false positive rate. This may, however, create latency in terms of the IDS response as single nodes are not given the authority to decide about the maliciousness of another node even if they have enough evidence.

Another way of modelling the hierarchical architecture was explored in [31]. Here, a zone-based framework is used to divide the whole ad hoc network into non overlapping zones. Nodes in a zone are either gateway nodes (inter-zone nodes), if a connection to a node in a neighbouring zone exists, or intra-zone nodes, otherwise.

In the proposed IDS framework, every intra-zone node runs a LIDS (Local IDS) locally to perform local data collection, anomaly detection and to initiate local response using mobile agents while gateway nodes run GIDS (Gateway IDS). GIDS are organized in multiple layers and are in charge of initiating global and zone intrusion detection and response.

In some other hierarchical IDSs, the CH is solely responsible for the detection of intruders within its cluster. Thus, instead of performing host-based intrusion detection at each node, a dedicated CH is selected to collect security-related information from nodes in a cluster.

For instance, the intrusion detection model proposed in [62] forms a cluster head-centred backbone network. This is achieved through a decision mode of joint detection used among CHs and vote by ballot in partial CHs. More specifically, the proposed model adopts a clustering algorithm for the building of clusters, which form the platform for the agent-based intrusion detection. detection agents are activated on elected CHs at the same time of cluster formation. These agents use a parameter based intrusion detection method that allows them to detect any abnormal activities within a cluster and to generate local response in case of intrusion detection. In case of uncertainty, however, the cluster head node will trigger the joint detection among the CHs that will use a partial voting to determine malicious nodes.
According to its authors, the proposed model has advantages of short computing time, low consumption of both bandwidth and power and high detection rates. Nevertheless, mobility might present a serious problem to the proposed model. Actually, nodes mobility leads to cluster reformation which, by the way, implies the regeneration of detection agents thereby, resetting the detection process. This might result in: delaying the detection and response to intrusions, network overhead and nodes' resource consumption especially if a node is always chosen as a cluster head. Also, no mechanism for preventing a compromised node from being a cluster head was proposed.

Darra et al. [22] presented a hierarchical cluster-based IDS architecture for MANET. The proposed IDS architecture is organized into autonomous and distributed multi-levelled hierarchies. Each level consists of several clusters in which specific nodes act as CHs gathering local audit data from their CMs, analysing them and extracting security-related information. In order to improve detection accuracy and reduce energy consumption, this architecture adopts and enhances the mobility and energy aware clustering algorithm (MEACA). The improved algorithm maximizes the clusters' stability by: (i) forming clusters of nodes with similar direction and speed and (ii) assigning cluster head functions to nodes with relatively low mobility and high energy levels. Thus, mobile nodes

of the same cluster appear more static to each other thereby avoiding cluster reformation. Moreover, this IDS balances the energy consumption in a fair and efficient manner. For instance, nodes with adequate energy undertake more detection responsibilities than nodes with low energy levels.

In a hierarchical architecture, it is also possible to start the detection process at the bottom level of the hierarchy then move up in the hierarchy seeking more detection accuracy.

In [92], the IDS architecture is designed as a dynamic hierarchy in which intrusion data is acquired by network nodes and is incrementally aggregated, reduced in volume, and analysed as it flows upwards to the CH.
This IDS consists of two broad modules: the Cluster-Head Module (CHM) running only on CHs and the Cluster-Member Module (CMM) running on all network nodes i.e., both CHs and CM nodes. Every CMM maintains a database denoted intrusion interpreter base in which attacks' signatures and related thresholds are stored. It detects intrusions locally and may request the CH to initiate a cooperative intrusion detection and response action if additional information or a global response is required. In case of cooperative intrusion detection, the CH dispatches mobile agents to gather information from other members in the same cluster and other clusters, and then processes the gathered information to detect any intrusion in a global scale. If an intrusion is detected by a CMM, it initiates a local response and, if need be, it communicates its response to its CH. This latter, via its CHM, logs the event and informs the nodes within its cluster and the adjacent CHs (which in turn inform their CMs) to isolate the offending node from the network.

A major problem, not tackled in these works, is the fact that, in a hierarchical architecture, there should be a mechanism for preventing a compromised node from being elected as a CH. Nevertheless, this architecture is still the best choice in cases where not all the nodes are capable of performing IDS tasks either because of their limited resources or due to their weak computational capabilities.

3.2.4.4 Agent-based IDSs

This architecture is based on the distribution of the intrusion detection tasks amongst a number of software agents.

A. Software Agents:
An agent [10, 11] can be defined as a computer system that is able to execute autonomous

actions in its environment, in a flexible and intelligent manner, in order to achieve a predefined goal. Therefore, a multi-agent system is a system that consists of a collection of autonomous agents that can interact together to learn or to exchange experiences. Agent-based systems usually encompass three main types of agent architectures, namely: reactive, deliberative and the hybrid architecture where aspects of both reactive and deliberative agents are combined.

Reactive agents do not have representations of their own environment and act using a stimulus/response type of behaviour; they respond to the present state of the environment in which they are situated. They neither take history into account nor plan for the future. Reactive agents make decisions based on local information. Thus, they cannot take into consideration non-local information or predict the effect of their decisions on the global behaviour of the multi-agent system. Moreover, they lack adaptability as they cannot generate an appropriate plan if faced with a state that was not considered a priori. Despite these limitations, reactive agents still have the advantage of being quick which necessarily makes them desired in rapidly changing environments.

The key component of a deliberative agent is a central reasoning system that constitutes the intelligence of the agent. Thus, unlike reactive agents, deliberative agents maintain a model of the internal state and they are able of predicting the effects of their committed actions. More importantly, these agents are mainly characterized by their ability to generate plans that successfully lead to the achievement of their goals even in unforeseen situations. Unfortunately, a major problem with deliberative agents is that the sophisticated reasoning can slow them which may cause latency in the reaction time which is undesirable especially in case of real-time applications.

Regardless of their architecture, agents present several common features, among which we cite:

- Autonomy: agents operate without the direct intervention of humans, and have some kind of control over their actions and internal state. In other words, it takes actions based on its built-in knowledge and its past experiences;
- social ability: agents interact with other agents via some kind of agent-communication language;
- reactivity: agents perceive their environment and respond in a timely fashion to changes that occur in it;
- pro-activeness: agents do not simply act in response to their environment, but they are able to exhibit goal-directed behaviour by taking initiative;

- negotiation: the ability to conduct organized conversations to achieve a degree of cooperation with other agents;
- adaptation: the ability to improve its performance over time when interacting with the environment in which it is embedded.

With these interesting features in mind, many researchers sought to investigate this technology in developing optimal, adaptive and comprehensive intrusion detection systems to fit MANET security requirements. Agents exploited in intrusion detection can be either stationary agents, used mainly for monitoring purposes and for local intrusion detection, or mobile agents best suited for distributed operations such as: gathering network-related information, broadcasting detection results and performing global responses.

B. Stationary Agent Based IDSs

FORK [83] is a two-pronged strategy to an agent-based IDS for ad hoc networks, in which only those nodes that are capable of participating in the intrusion detection process, in terms of their available resources and their reputation level (which increases when the node successfully assists in intrusion detection tasks and decreases in case of failure) are allowed to compete for and get the IDS agent tasks. The authors base the task allocation process on principles of auctioning. Whenever one or more nodes detect certain changes in the network, they initiate an auction process by submitting auction requests to the rest of the network nodes. The interested nodes submit their bids to the initiating node(s) that, then, choose them based on several metrics including a battery power metric. Finally, the chosen nodes perform the intrusion detection tasks using a variation of the Ant Colony Optimization (ACO) algorithm. For instance, each network node contains all the modules (lightweight agents) required to perform the anomaly detection tasks such as: host and network monitoring (data collection), the decision making given a set of audit data, and the activation of defensive actions if malicious behaviours are detected.
Experiments show that the proposed detection algorithm is effective in terms of the accuracy of rules formed and the simplicity in their content. It was also shown that detection rates were improved compared to other IDSs. Nevertheless, node mobility, which highly affects the detection accuracy, was not considered in this evaluation. On the other hand, the distribution of detection tasks among a set of carefully selected nodes helps conserving local resources, mainly battery power. However, this IDS seems to be insecure as no suggestions about securing the mobile agents were given. Also, the cooperative nature of the proposed detection scheme offers the opportunity to malicious nodes to cause resource-consumption-like attacks by initiating fake detection tasks.

The biological immune system was a source of inspiration for several agent-based IDS designers, who tried to take benefit of the analogy that exists between the two fields to approach the distinguished ability of the biological immune system to distinguish self from non-self and to protect the human body from this latter.

One example of such IDS architecture is presented in [15]. Here, the authors designed an immunological intrusion detection system based on the agent concept for securing MANET. This IDS consists of a set of autonomous agents, denoted detectors, distributed among the different network nodes. Each detector implements an anomaly intrusion detection approach based on the negative selection algorithm and monitors the communication of its neighbouring nodes. For that, every node maintains both a set of self-patterns (characterizing normal behaviour) and a set of non-self-patterns (characterizing potential anomalous behaviour). Upon the observation of any kind of disturbance in the behaviour of a node, the concerned detectors communicate with neighbouring detectors in order to consult their observations. Then, a collective decision is undertaken based on the reliability weight of contributing detectors. This weight is applied by the super-detectors that represent the second level of detectors.
Although it seems simple and effective in detecting intrusions, this approach might have a negative effect on the nodes' performance mainly in networks with high mobility, where detectors and super-detectors have to regenerate neighbours' self-patterns and non-self-patterns as well as neighbouring detectors' reliability weights each time the network topology changes. So far, the approach ensures a high level of reliability because even if the detectors cannot maintain contact among themselves, they still may react to the behaviour they sense.

Some generic stationary agent-based IDSs that can be adopted for MANET were also proposed in the literature. For instance, Servin and Kudenko [95] proposed a hierarchical architecture of distributed IDSs integrated by remote sensor agent diversity and reinforcement learning (RL) to detect and categorize DDoS Attacks.
This architecture is built from m cells with each cell composed of one central agent (RL-IDS) and n sensor agents. In RL, agents or programs sense their environment in discrete time steps and they map those inputs to local state information. Under this consideration, distributed sensor agents were configured so that to process the local state information and pass on short signals up a hierarchy of RL-IDS agents. That is, a sensor agent learns to interpret local state observations, and communicates them to a central agent higher up in the agent hierarchy. Central agents, in turn, learn to send signals up the hierarchy, based on the signals that they receive.
Then, via the signals from the lower-level RL-IDS agents, the agent on top of the hierarchy

learns whether or not to trigger an intrusion alarm. If the signal is in accordance with the real state of the monitored network, all the agents receive a positive reward. If the action is inaccurate, all the agents receive a negative reward. Thus, after a certain number of iterations of the algorithm, every agent would know for each state the action that they need to execute to obtain positive rewards. Also, the Q-learning technique and a simple exploration/exploitation strategy are used to enable the agents to learn an accurate signal policy and to maximize the obtained reward over the time.
The proposed approach was evaluated in an abstract network domain with different architectures varying the number of agents, the number of states per sensor agent, the exploration/exploitation strategy, the distribution of attacks as input information, and the agent architecture.
Clearly, a clustered MANET would be a good ground for such an IDS architecture with clusters mapping the cells, cluster-heads running RL-IDS agents, and cluster-member nodes running sensor agents.

C. Mobile Agent Based IDSs

Mobile agents [12, 59] are special software agents that have the ability to roam through networks. Mobile agents offer several potential advantages over stationary agents when used to design MANET applications with respect to load reduction, dynamic and static adoption, and bandwidth conservation. In this overview, Roy and Chaki [87] introduced a totally mobile agent based IDS to detect blackhole attacks in MANET. This IDS, referred to as MABHIDS, define two types of agents: a mobile agent and a specialized agent. First, the source node generates a mobile agent and forwards it to the next hop node in the route to the intended destination. The mobile agent has to collect the raw data from the host machine then it computes the packet delivery ratio R_i or the i^{th} host. The specialized agent then compares the R_i value with a threshold ThR, predefined by the source node, and gives responses to the source node accordingly.
Although this approach was proven to be efficient in detecting the blackhole attack, it is still too limited and needs to be extended to detect more attacks especially as the number of newly discovered attacks is always increasing. In addition, MABHIDS is based on merely mobile agents and their ability to roam in the network, but no security mechanism was integrated to protect them from attacks though they are well known for their security vulnerabilities.

Detection of unknown attacks together with the ability to detect attacks at different network layers is indispensable for a comprehensive IDS. Realizing that, Devi and Bhuvaneswaran [26] proposed an efficient cross layer intrusion detection architecture. If

the node that detects an intrusion has a high accuracy rate, it can independently determine that the network is under attack and thus, it initiates the alert management agent. However, if the support and confidence level is low or intrusion evidence is weak and inconclusive, then it can make collaborative decision by gathering intelligence from its surrounding nodes via protected communication channels. Upon receiving alerts (either from local detection or cooperative detection agents), the alert management agent collects them in the alert cache for t seconds. If there are more abnormal predictions than the normal predictions then it is regarded as abnormal and with adequate information an alarm is generated to inform that an intrusive activity is in the system.
Evaluation of the proposed approach revealed that it has some advantages. For instance, the way in which generated alerts are treated reduces both false positive and false negative alarms. Also, the use of the fixed width algorithm helps in detecting attacks at different layers while the fast apriori algorithm increased the speed of detecting them significantly. Nevertheless, this approach implicates that the nodes should have considerable computational capabilities to run such algorithms. Furthermore, the initialization of a cooperative detection depends on the level of intrusion evidence within the local detection module but there is no specification about when intrusion evidence is deemed weak or strong.

In [92], agents are used along with a hierarchical intrusion detection architecture. CMs detect intrusions locally and may request the CH to initiate a cooperative intrusion detection and response if required. In case of cooperative intrusion detection, the cluster-head dispatches mobile agents to gather information from other members in the same cluster and other clusters, and then processes the gathered information to detect any intrusion in a global scale. Intrusion related message communication is handled by mobile agents. CHs can create, dispatch, and process the results returned by the mobile agents. A database is maintained for the mobile agents that are created and dispatched. They are created only at the time of cooperative intrusion detection and are destroyed immediately after accomplishing the designated tasks successfully or if the associated timer expires.

Pattanayak and Rath [73] proposed that a CH is to be elected, at the initiation of each application, based on a battery power metric. A dedicated mobile agent, consisting of a registration module (RM), a service agreement (SA), a detection module (DM), and a prevention module (PM), is incorporated in each cluster. During the initiation of a new application, all the nodes in the cluster need to register with the mobile agent and to accept a service agreement specific to the initiated application. The mobile agent, on its own, maintains a list of registered nodes in its registration module and uses the detection module to monitor routed packets.

This approach is time and resource consuming for all the packets are routed and monitored by the CH (mainly in case of several concurrent applications). Hence, a battery power metric is not sufficient to choose a reliable CH able of handling all the cluster communications in addition to performing intrusion detection tasks.

In [47, 46], two novel multi-agent-based dynamic lifetime intrusion detection and response schemes are proposed to protect AODV-based MANETs from blackhole and DoS attacks. In both schemes, agents are designed so as to dynamically adapt their creation, execution and expiration to the routing process status and are related to one RREQ–RREP stream. In [47], each agent is responsible for the monitoring of nodes within a three-hop zone. When the RREQ or RREP messages are out of this zone, a new agent is generated to execute the detection algorithm so as to avoid the delay in listening the routed packets. Once created, the current agent executes the intrusion detection algorithm based on the related link list and MAC-IP control table. In [46], however, only link list data is used by the IDS agent. If the agent finds the node itself has malicious behaviour, it can migrate to another high trustworthy node. Finally, if there is no RREQ–RREP stream in the network for some time, the related agent expires and the detection information is saved by the agent node for future detection.
While they efficiently improve trustworthiness, decrease computing complexity and save energy consumption, both approaches badly affect the network performance especially when many nodes initiate routing operations simultaneously. More specifically, the association of a new agent to every RREQ-RREP stream might overload the nodes (mainly those that are involved in many routes) with heavy extra processing loads entailed by the different detection agents.

D. Hybrid-Agent based IDSs

While the previously discussed IDSs were comprised of collections of merely stationary or mobile agents, other works like [105] were looking forward to enhancing the IDSs' fault tolerance and scalability through the combination of both stationary and mobile agents.

Because traditional security-centric mechanisms consume a large amount of network resources and thereby degrading its performance, Wang et al. [105] designed a network performance-centric anomaly detection scheme for resource constrained MANETs. This scheme employs a fully distributed multi-agent framework. More specifically, the system uses a platform of mobile agents to design the energy-aware and self-adaptive anomaly detection. In this concern, four kinds of agents, residing on every node, were defined, namely: the network tomography agent (NTA), the anomaly detection agent (ADA), the communication service agent (CSA), and the state detection agent (SDA).

Anomaly detection proceeds in two phases. The first phase aims at detecting link delay anomalies while the second phase tries to quickly detect and accurately localize malicious nodes on links. For instance, the detection is started by executing an energy-aware root election mechanism that selects the most cost-efficient node as the root that will sponsor system services. By the way, the NTA on that node will be considered as the root NTA while other NTAs remain inactive to save resources. Each ADA independently undertakes to set up the delay distribution profile of the link on which it is located. Once the profile of a link delay characteristics is obtained, it can be compared to the inferred delay of the link delivered by the NTA. If the inferred results go beyond a threshold value, the link is considered as an anomalous link and an alarm is raised. Since each ADA performs local detection using local audit data, the ADAs around an anomalous link can cooperate locally to confirm the maliciousness of a node. CSA agents are used for communication services among the different nodes. For the sake of security, SDA agents are used to check the validity of CSAs and NTAs in the cooperative mobile nodes using MANET security encryption mechanisms. This approach is too limited as it detects only link delay related attacks.

E. Discussion

Building on the studied approaches and their analysis, it is clearly seen that agent-based MANET IDSs have some common features:

- Distribution of the detection tasks among a group of collaborating agents distributed over the network;
- use of mobile agents for both communication (collaboration) and data collection on remote hosts;
- approaching real-time detection and response;
- besides, almost all these IDSs are lightweight, flexible and present an exceptional ease in maintenance (modifications and extensions can be made without halting the whole system).

Table 3.3 [Mechtri2016] summarizes some of the several advantages obtained when using agent technology for the building of MANET IDSs in particular and MANET applications in general, with respect to MANET requirements.

Agent features	MANET shortcomings	Description
Scalability	Constrained processing and energy power	Agents reduce the computational load and consumed energy by dividing the (detection) tasks over different hosts.
Mobility	Limited bandwidth and storage capacity	Instead of transferring huge amounts of data (audited data), the processing unit (detection agent) is moved to data.
Portability	Heterogeneity of devices	Agents run on agent platforms, thereby guaranteeing independence from the platform of the host.
Autonomous execution	Dynamic topology	If the network is segmented or some agents cease to function (under the threat of an attack), the rest of the agents can still continue to function (guaranteeing a proportional level of security).
Fault tolerance	Vulnerability to attacks	An attacker can disable a small finite number of backups but not all of them (agent-based applications use techniques like redundancy to protect their components).

Table 3.3: Advantages of using agents for MANET intrusion detection

3.3 Intrusion response in MANET

Intrusion response is a vital part of MANET defence systems. It can be implemented as either a part of the IDS or as an independent system that works together with the IDS. It represents the way in which the system will react after an intrusion is detected. Intrusion responses are usually reported to auto response systems or security staff for automatic or manual appropriate response actions.

Intrusion response systems usually start by assessing the damage caused by the intruder(s) along with the identification of the potentially exploited vulnerabilities. Then, it proceeds to execute the actual response actions. These latter may range from the generation of simple notifications to actively responding to the source of intrusion. Figure 3.2 illustrates the general processes involved by automated response systems.

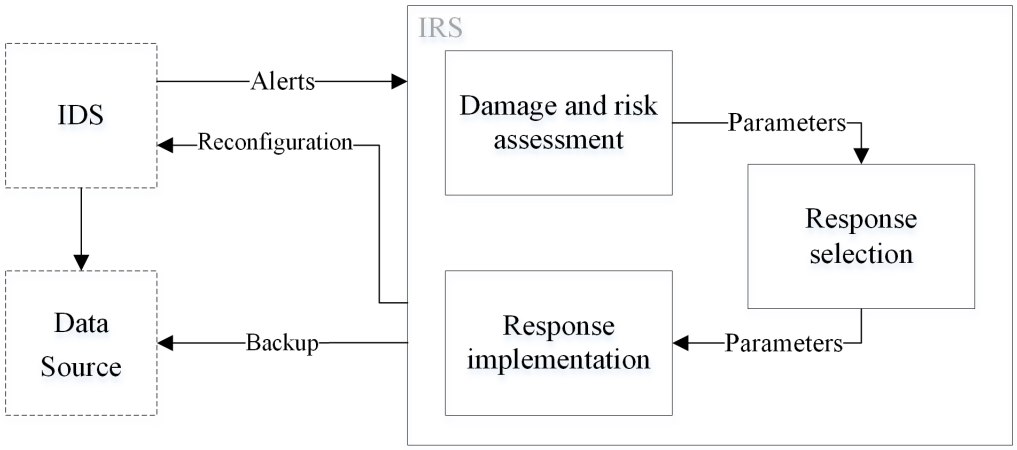


Figure 3.2: General Architecture of an Automated Response System

In the literature, generated responses are typically classified as either passive or active.

3.3.1 Passive responses

Passive response systems, also called notification systems, consist of the generation of alarms and reports [26] and/or the recording of intrusion related information in specific log files or databases for future reference [31]. An IDS generates alarms to inform the node's user or other nodes about the detected intrusion(s). Alarms have several forms: email, SMS, notification message on the screen, or notification sounds.

In [31], the network is divided into zones with each zone having a gateway node connecting it to other zones. If a node detects an intrusion locally, it will initiate a local alarm, by sending an alarm message to the GIDS (Gateway IDS) on the nearest gateway node. The GIDS in turn will trigger either cooperative agents or local and global response agents depending on the strength of evidence in the intrusion. Then, the GIDS, through its manager agent stores the alarm in the long term memory (LTM) if the intrusion is detected with strong evidence or in the short term memory (STM) in case of weak or inconclusive evidence, for future reference.

3.3.2 Active responses

Passive responses are not suited for MANET environments where every node should react on its own since no form of centralized administration exists and nodes cannot rely on other nodes (these latter can move away or leave the network at any moment). In this respect, generated responses evolved to present more effective solutions to hinder the attackers and stop damage spread. Active responses consist of a predetermined set of

actions (countermeasures) executed whenever an intrusion is detected to stop its spread and to locate and deter the accused node(s). Active responses may include: (i) temporal punishment like discarding the intruder from active routes [50], (ii) permanent punishment like interrupting all communications with the intruder (i.e. discarding the intruder from the network) [109, 77, 98], and/or (iii) executing some corrective actions [62].
The following are some interesting active response approaches for ad hoc networks.

3.3.2.1. Static response systems

Ping et al. [77] proposed an intrusion detection and response system for MANET based on mobile agents. It is composed of a monitor agent residing on every node, a decision agent, and a collection of block agents. Each monitor agent collects information of its neighbour nodes' behaviour, filters it from unnecessary information, and sends this information after coding to the decision agent upon receiving a query message from this latter. The decision agent, then, detects intrusions by analysing the received data. Due to resource constraints in MANET, decision agents are distributed over only some nodes. However network dynamics may cause a decision agent to move with its node thereby leaving the zone without any supervision. To tackle this problem, the authors suggested that if monitor agents in a zone have not received the query packet for a long period, a new node will be selected to run the decision agent.
If an intrusion is detected, the decision agent will produce block agents that will be sent to the neighbour nodes of the intruder to form the mobile firewall and isolate the intruder. To finish, a process of local repair will be executed to find new routes to replace all paths that include the intruder.
Though it succeeded in automating the response process, the proposed approach adopted no mechanism to prevent malicious or compromised nodes from initiating blackmail attacks through the generation of fake query messages.

In [86], a two-tier hybrid IDS for MANET is proposed and evaluated. A local-level IDS is located in the first tier and will be triggered first to investigate any suspicious activity before being passed to the global detection mechanism, which is located at the second tier. Since global detection mechanisms rely on information provided by other nodes, this latter must be filtered so as to protect the network from attacks against the IDS itself. However, because voting mechanisms were judged not to be efficient in defending against multiple blackmail attackers, the concept of friendship has been introduced to global detection and response mechanisms. For instance, only votes from friends can be counted to judge any intrusive activity. As for the response mechanism, the authors deployed only a basic re-

sponse strategy that consists of: (i) a local response aiming at excluding malicious nodes from any future network activity and (ii) a global response consisting of broadcasting intrusion alarms thereby, allowing neighbouring nodes to take reactive actions on their own.

Some researchers opt to include some corrective measures rather than just react against the intruder. An example of such solution is presented in [62]. Upon detection of an intrusive activity, a global network response in the form of blacklist broadcasting will be initiated. However, if the intruder is the cluster-head node itself, neighbouring cluster-head nodes, in addition to screening the intruder, will split and merge its cluster, or assign a new cluster-head using the adopted clustering algorithm.

3.3.2.2. Adaptive response systems

It would be inadequate to apply responses to all types of intrusions in a fixed manner mainly if coupled with a high false positive alarm rate. For instance, this would result in discarding some innocent nodes and may lead to the disruption of some network functionalities like losing connectivity, congestion, and an increased network latency. By deploying such responses, the response system might incur more damage than the one caused by the intrusion itself.

Considering these issues, some researchers tried to optimize and adapt the generated responses to the state of the target system and to the detected intrusion(s). This adaptation can be achieved using static mapping, dynamic mapping, or cost-sensitive mapping [97]. The following are some approaches that explore variants of these models.

In [73] , a detection module monitors each packet routed through the CH. A mismatch in the packet's header signals an intrusion and a response is immediately undertaken. If the mismatch occurs in source and destination addresses, a mobile agent will inform the CH to drop the packet and to block the respective node. If the mismatch occurs in the application ID or the packet length exceeds the threshold, then only the packet will be dropped by the CH.

Nadeem and Howarth [66] presented an intrusion detection and adaptive response mechanism (IDAR) that: (a) employs a hybrid detection technique and (b) is based on a hierarchical architecture in which nodes operate as manager nodes, CHs, or CMs. IDAR implements three different response actions: (1) Isolation which aims at treating the intruder as non-existent. Isolation is undertaken only if the confidence in a detected attack is high, and the attack is severe, and the network performance has degraded considerably

since the attack was launched. (2) Route around attacker consisting of the elimination of the intruder from any further route discovery process, while allowing it to forward data packets for other nodes over existing routes. This response action is mainly deployed when the confidence in the detected attack is reasonably high and the network performance degradation is noticeable and (3) no punishment if the attack's confidence is low and its effects on network performance can be tolerated. They also used a decision table to represent the intrusion response action selection criteria. For instance, whenever an intrusion is detected, the manager node calculates its confidence level and evaluates the network performance degradation since it was launched. Then, a response action is selected accordingly and the necessary actions required to implement it are taken. Since the approach is based on a hierarchical architecture, the isolation of intruders (which is likely to cause network partition) might generate additional overhead for the recreation of affected clusters whenever need be. In addition, manager nodes and cluster heads constitute a single point of failure.

In [33] the authors proposed a multi-attribute genetic algorithm model to develop an IRS capable of selecting the most cost-effective response. The model is based on a multi-criteria decision-making technique that considers attributes like the financial cost, the reputation loss, and the processing resource. The IRS assesses the cost of each response alternative based on a cost-benefit model and selects the one that has the least negative effects on the system.

3.4 Survivability in Distributed Systems

The use of centralized IDSs raises several issues like their being a single point of failure, having a limited view and low efficiency against distributed intrusions, latency in both detection and response, and the overloading of one or more nodes with data collection, correlation, and intrusion detection tasks. Such issues boosted the development of more open and decentralized architectures in favour of broader coverage, lower resource consumption, and faster intervention.

However, distributed systems are by nature fault-prone. The situation gets more complex in the presence of intrusions that continue to grow in both number and severity, especially in open environments like MANETs. Thus, to better and safely benefit from the distributed architecture, a distributed IDS should be able to protect itself from being compromised and to not introduce extra overhead so that not to add to the networks' vulnerabilities.

The vulnerabilities of the mobile ad hoc networks and the proliferation of intrusions and thereby the need for survivability have been widely studied in the literature. For instance, there has been considerable research in the fields of self-healing, fault-tolerant systems, and survivable networks. In this section, we review some interesting works in these areas.

3.4.1 Preliminaries

Fault, error, and failure: Failures are events that occur when there are deviations of one or more of the external state of the system from the correct service state. The causes of such deviations, called errors, are known as faults. Figure 3.3 illustrates the sequence leading from faults to failures. The activation of a fault causes an error to happen. If no recovery mechanism is adopted or the recovery cannot be done in time, an error can lead to more errors and eventually, a failure will be observed.

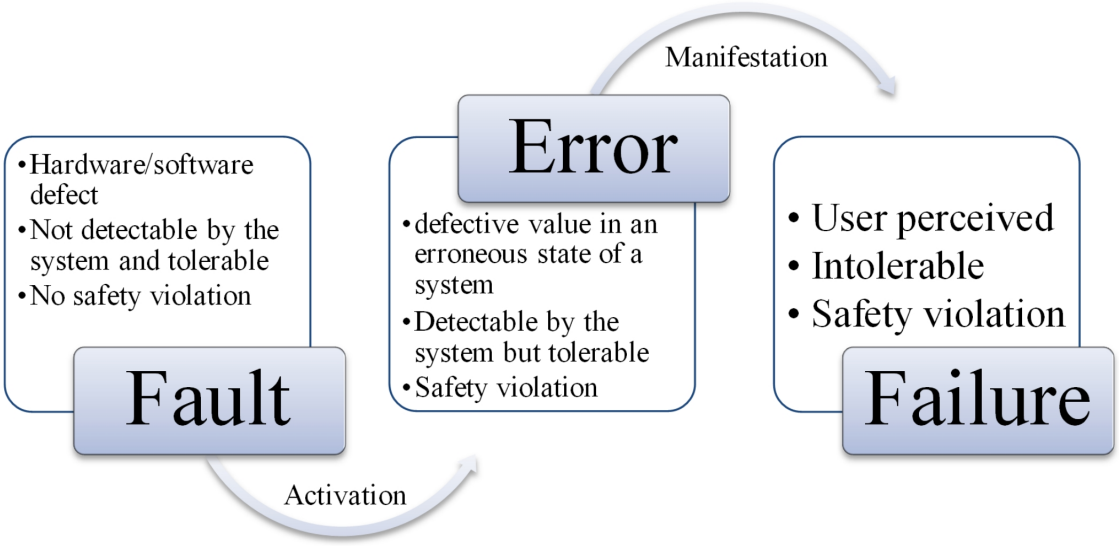


Figure 3.3: Fault-Error-Failure transitions

Fault tolerance: FT is the property of a system to correctly provide its services even in the presence of faults. FT process usually involves error (or fault) detection, diagnosis, containment, and the recovery [81]. Several techniques were proposed in the literature to provide fault tolerance [42, 110].

Self-healing: Self-healing [36] can be defined as the property that enables a system to perceive that it is not operating correctly and, without human intervention, make the necessary adjustments to restore itself to normalcy. Typically, a self-healing process involves operations like monitoring, diagnosis, damage repair, self-testing, and deployment.

3.4.2 Fault-tolerant IDSs

Kaur et al. [54] analysed and discussed one of the most challenging issues in the field of intrusion detection, which is the IDS' fault tolerance. For instance, they evaluated some of the widely used fault tolerance mechanisms, namely replication of software agents, employment of redundancy in processing elements, integrity checking for self-healing, use of reconfigurable hardware and restructuring architectures, and fault detection using heartbeat messages in multi-agent systems.
The results of this study show that an IDS must be fault tolerant and that replication techniques, which we will use in this paper, provide the IDS with both high availability and reliability. Example IDS that support fault tolerance are [17, 89].

In [17], the authors focused on the detection of intrusions at the application layer. Similarly to many other agent-based IDSs, they used a local IDS, consisting of a monitoring and detection agent, a response agent, and a communication agent to detect intrusions at every network node. Their main contribution is the use of mobile agents to augment each node's intrusion-detection capability. Specifically, they equipped the network with a mobile agent server capable of creating and dispatching three types of mobile agents: update, analysis, and verification agents.
If a local IDS fails to identify a suspicious behaviour, its response agent will request the mobile agent server to send analysis agents for further investigation. The analysis agent is capable of a more detailed analysis and diagnosis compared to the local IDS as it can launch multi-point network-based anomaly detection. Once the investigation completed, the analysis agent will report the results to the mobile agent server. Hence, if a new attack type is detected or the suspicious activity is judged as a change in the node's behaviour, an update agent will be created to update local IDSs' databases with the new attack signature or normal profile.
Further, the mobile agent server periodically checks the status of local IDSs using verification agents. If any vulnerability is detected, it will patch and install programs on the concerned mobile nodes via its update agents.
Clearly, mobile agents can overcome network latency and reduce the network load related to intrusion detection. Also, this approach was a step forward in enhancing agent-based IDSs' fault tolerance, but it might lead to further problems. For instance, the mobile

agent server might exhaust the node's resources (mainly processing and storage) in addition to being a single point of failure.

Sasikumar et al. [89] developed a dynamic distributed intrusion detection system (DDIDS) based on mobile agents. In the proposed architecture, each DDIDS system has a connection with other DDIDSs for information sharing as well as problem solving. Each of these systems is composed of three layers: layers consisting of host agents and net agents, mobile agents, and decision making and replication agents. Also, the system has a DDIDS console that has control over every DDIDS agent in the network and that supports for report preparation.
The use of decision-making and replication agents helped greatly in increasing DDIDS fault tolerance. Moreover, the use of agents improved the IDS' performance mainly in terms of real-time intrusion detection.

3.4.3 IDS-based self-healing networks

Most IDSs try to mitigate the detected intrusions but never deal with damaged data. Such issues can be dealt with using self-healing mechanisms. There is a considerable number of works addressing self-healing issues in MANET [36, 8] but only a few of them combine healing solutions with intrusion detection tools. The following are some examples of IDSs that offer healing options to the network.

Lee et al. [60] proposed a decentralized self-healing mechanism that detects and recovers from wormhole attacks in wireless multi-hop sensor networks using connectivity information. This mechanism, denoted SWAT, identifies the locations of malicious nodes, isolates them, and finally recovers the routing structure distorted by them. For that, each sensor node maintains a neighbour list containing the connectivity information about one hop and two hops neighbour nodes. Using this list, a node monitors the connectivity with its neighbours. Anomaly detection within these connections results in the production of a danger signal in the form of a control packet. This latter triggers the recovery phase in which recovery packets are used to isolate the wormhole nodes and to heal the caused damages within the wormhole sphere based on a pre-established routing tree structure.

In [28], a bio-inspired intrusion prevention system (IPS) is proposed. This approach implements an analytical computational framework based on the danger theory. Using agents (Sense, Analysis, and Adaptive agents) of multi-layers, the proposed IPS analyses the behaviour of system processes and network traffic to detect harmful events. Upon detection of a potential intrusion, it will be prevented by disconnecting or blocking the

suspected connection. Then the adopted self-healing mechanism will be triggered so that to regenerate the damaged components. For that, the self-healing agent is provided with a knowledge base containing all candidate system components, in addition to the healing function. For instance, whenever a healing message is received from the Analysis Agent, a healing component is immediately identified, deployed and tested to keep the system in function. The designed IPS is autonomous and the network's fault repair ability was considerably enhanced through the adopted self-healing mechanism.

Kong et al. [56] proposed a new intrusion protection mechanism based on the notion of self-healing communities. These communities consist of groups of neighbouring nodes among which a network service is distributed so as to mitigate the adverse actions of selfish and malicious nodes. For each end-to-end connection, a chain of self-healing communities along the shortest path are established based on localized simple schemes. The idea, here, is that a self-healing community is perceived as a big virtual node that replaces the conventional single forwarding node. Thus, data delivery is considered as a combination of conventional node-based data forwarding and community-based healing.
At each intermediate community in a route, the most recent control packet forwarder is supposed to be the current data forwarder. If this node fails to forward a packet due to maliciousness, selfishness or network dynamics, members in the same self-healing community will make up. This way, routes can be healed locally with minimal latency. Yet, because such self-healing communities might lose shape due to mobility and network dynamics, their reconfiguration is deemed crucial for the survivability of the proposed solution. For that, the authors used end-to-end probing with a probing interval adapted with respect to network dynamics.

3.5 Discussion

The study of exsiting IDRSs revealed the limitations of the stand-alone and hierarchical architectures. It, however, showed the importance of the decentralized architecture and the cooperation among LIDSs as well as the strong relevance to use agents in designing intrusion detection tools for MANET. In fact, many of the agent features show an exceptional match with MANET's inherent characteristics and agents are best suited for applications that are decentralized, changeable, ill-structured and complex like MANET intrusion detection.

In addition, there are not much works that fully address the resource constraints issue. Network and node capabilities should be given an appropriate weight when de-

signing MANET IDSs. For example, nodes should be assigned detection tasks based on their resources and communication between LIDSs should be adapted to the wireless links bandwidth. Also, the great majority of works done in the field neither address the IDRS security issues nor consider enhancing its fault-tolerance. IDRS' security and fault-tolerance are crucial since an IDRS should be highly available and not add to the network's vulnerabilities.

Besides, most existing IDRSs can detect intrusions with high accuracy but fail to eliminate their source. The best they can do is to generate passive responses in terms of alarms and blacklists. Development of adaptive and more corrective responses seems more consistent and can help enhancing the network's survivability and healing ability.

Table 3.4 [Mechtri2016] summarizes the main features of some of the discussed IDRSs, their main contributions, and the issues they do not address.

IDRS	**Technique**			**Data source**			**Response**		**Advantages**	**Limitations**
	Anomaly detection	Misuse detection	Specification based	Host	Neighbourhood	Network	Passive	Active		
CAIDS [18]	✓			✓			✓		- Considers resource constraints	- Vulnerable to distributed attacks
Farhan [31]	✓			✓			✓		- Dynamic adaption to environment changes - Scalable and robust	- Single point of failure (GIDS)
Sen [92]		✓		✓				✓	- Detection of distributed attacks	- High bandwidth consumption

Li [62]		✓		✓				✓	- Reduced Energy and bandwidth - Fast detection - FPR reduction	- High architecture maintenance cost under mobility - Single point of failure (CHs)
Ramachandran [83]	✓					✓		✓	- Accuracy and simplicity of rules - Improves detection rates - Energy conservation	- Mobility of nodes not addressed - IDS security issues not addressed
Byrski [15]	✓				✓		✓		- Simple and reliable	- High computational cost under high mobility
Servin [95]				✓		✓	✓		- Accuracy enhanced over time through learning	- High Communication overhead
Roy [87]		✓			✓		✓		- Simple - lightweight	- Detects only blackhole attack - Security of mobile agents not addressed
Devi[26]	✓			✓			✓		- Detection of attacks at different layers - Reduces FPR and FNR - Fast detection	- High computational load

Pattanayak [73]			✓		✓		✓		- High protection level	- Time and resource consuming - The method is not consistent due to unrealistic assumptions on nodes mobility
Hong-song [46]		✓				✓		✓	- Considers security of IDS agents - Low computional complexity - Saves energy	- Routing protocol dependent - Overload nodes if the number of RREQ/RREP increases
Chang [17]	✓	✓		✓			✓		- Enhances IDS fault-tolerance	- Single point of failure (the mobile agent server)
Wang [105]	✓			✓				✓	- Considers security issues of mobile agents - Considers resource constraints	- Detects only link delay related attacks
Ping [77]	✓				✓			✓	- Automated response - Considers node mobility and resource limitations	- Vulnerable to blackmail attacks

Table 3.4: Comparison of existing MANET IDRSs

3.6 Conclusion

In this chapter, key notions related to the field of intrusion detection and response in MANET were presented. A literature review highlighting the recent achievements in this field was also presented. Key points in this review were agent-based IDSs, fault-tolerant IDSs, survivable systems, and adaptive response systems.

This literature study revealed that the IDS' scalability, performance and fault tolerance can be improved through the use of agents to perform intrusion detection tasks in MANET. In addition, agents proved their utility in overcoming some MANET related problems such as the constrained resources and the heterogeneity of platforms. Thus, agent technology can be a good ground for the building of reliable IDSs that fit security requirements while satisfying MANET constraints.

However, relying on intrusion detection alone is not enough. Even with a perfect IDS, intrusions are merely detected and localized if it focuses only on detection. To actually stop the threat, detected intrusions need to be actively and adaptively responded to. In the following chapter, the architecture and components of a new IDRS are described.

Part II

Propositions

Chapter 4

MASID: A Multi-Agent System for Intrusion Detection in MANET [Mechtri2012]

4.1 Introduction

The literature study in the previous chapter revealed that, over the past years, there has been a growing interest in securing the mobile ad hoc networks. Some researchers developed preventive approaches to guarantee security while many others prefer the use of secure routing protocols in favour of more simplicity and accuracy. Also, there has been, recently, a great tendency to develop intrusion detection systems (IDS) specifically designed to fit MANET requirements in terms of both security and constraints.
This chapter presents MASID (Multi-Agent System for Intrusion Detection) [Mechtri2012], a new IDS for MANET in which a collection of agents is in charge of performing a distributed and cooperative intrusion detection. The distribution is achieved through the implementation of a local intrusion detection system on each network node, and cooperation is guaranteed by mobile agents.

4.2 Proposed Intrusion Detection System

MASID is a new MANET IDS in which the intrusion detection process is divided into subtasks handled by a set of software agents.

4.2.1 General architecture

MASID consists of a collection of agent-based LIDS (Local IDSs), distributed among all the network nodes as illustrated in Figure 4.1.

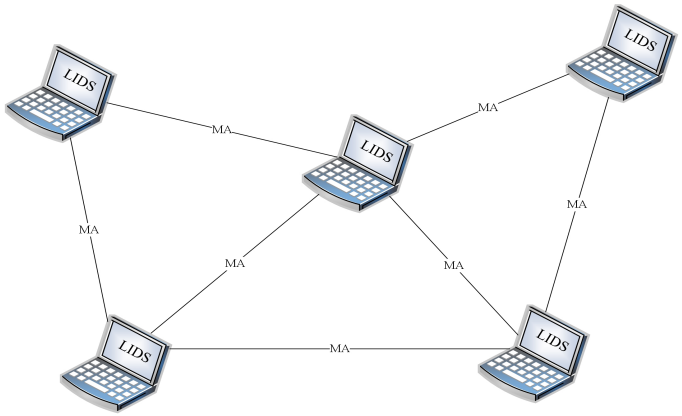


Figure 4.1: Distributed intrusion detection using MASID

Each LIDS runs independently and monitors local activities. It detects intrusions from local traces and initiates local and global response. If an anomaly is detected in the local data, or if there are signs of intrusion and there is not enough evidence, neighbouring local IDSs will cooperatively participate in the detection process, either by participating actively in the response or by, simply, providing some additional information (depending on the results of the local intrusion detection process). In this latter case, data provided by neighbouring nodes can help in taking a definitive decision about the detected suspicious actions. Figure 4.2. illustrates the steps followed by MASID to detect intrusions.

4.2.2 Local IDS

Each LIDS consists of five agents, playing different but complementary roles, and working together as shown in Figure 4.3.

These agents are either stationary or mobile agents, depending on the task they perform. Furthermore, they adopt two different architectures: they are either reactive or deliberative agents. The following subsections describe these agents and the tasks they perform.

4.2.2.1 Collector

The first agent is a data collection agent. It's a reactive agent that captures and gathers audit data from the network. We assume that nodes work in a promiscuous mode which

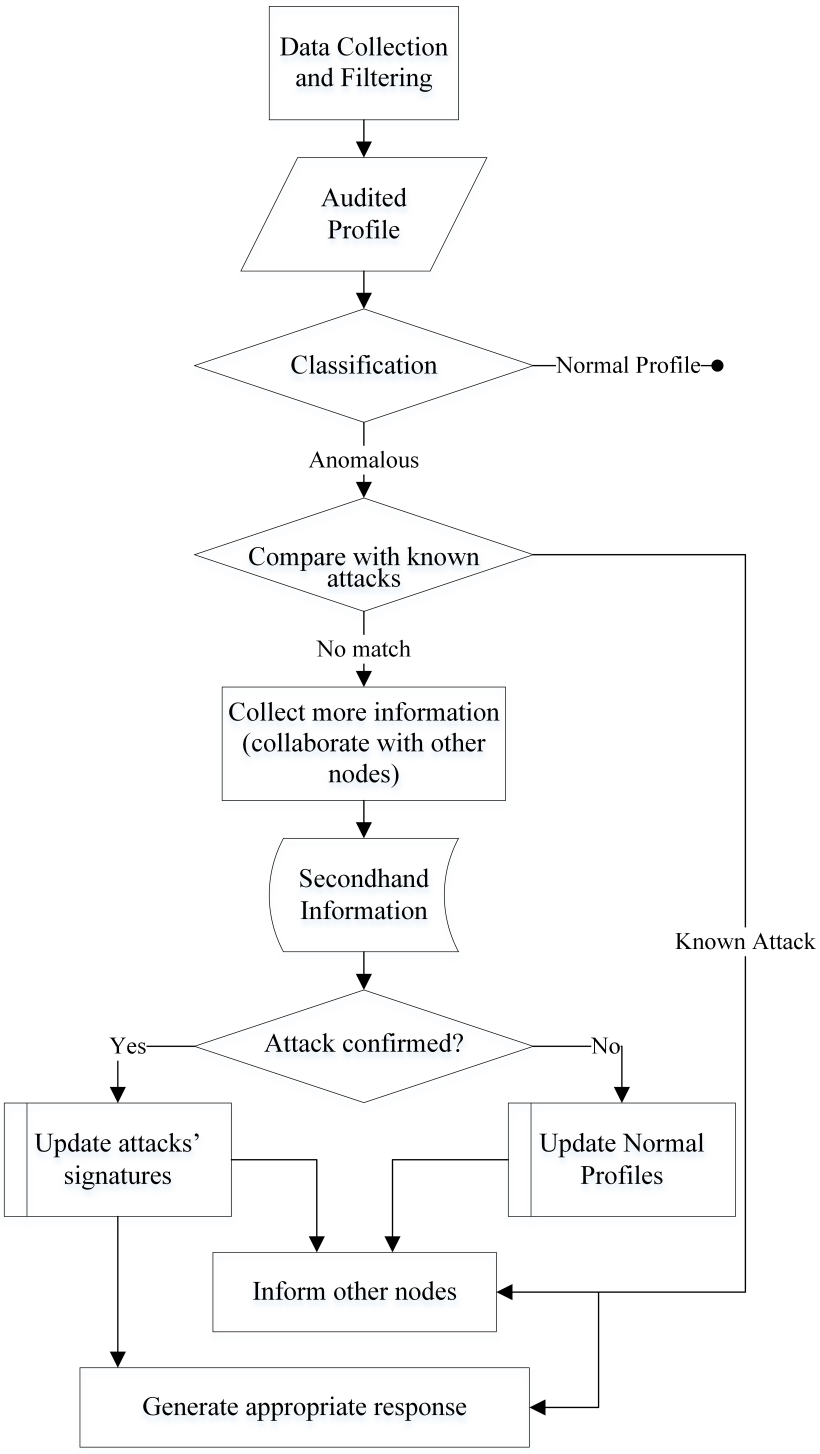


Figure 4.2: Intrusion detection process

means that every node can overhear the traffic within its neighbourhood. This agent is also responsible for filtering the collected data so that it keeps only those features that will be used by the detection agent during the detection process.

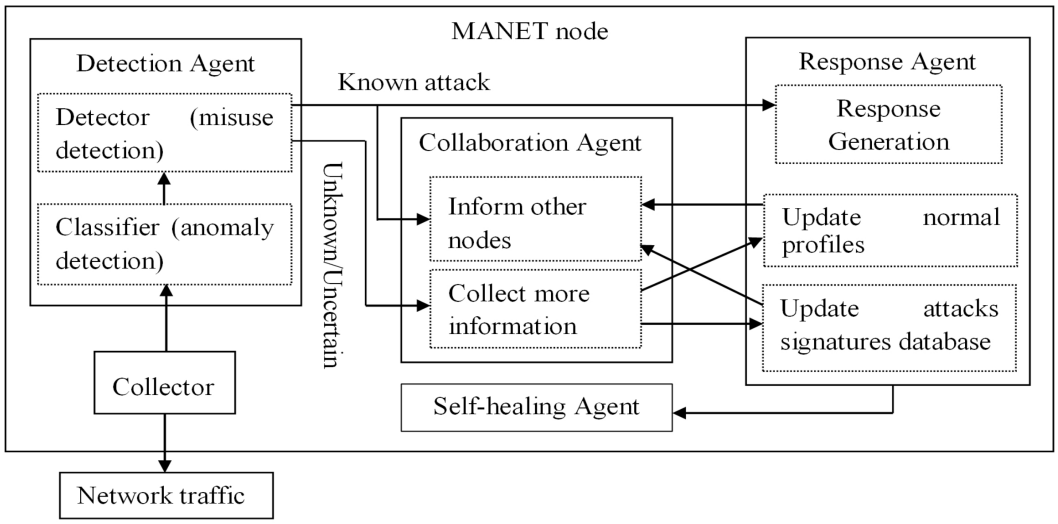


Figure 4.3: Local IDS Architecture

4.2.2.2 Detection Agent

This is a classification and detection agent. It uses data provided by collector to investigate and look up for signs of intrusions. It includes both a misuse detection (detector) and an anomaly detection (classifier) engine: anomaly detection to detect the different anomalies, and misuse detection to determine the exact nature of the detected anomalies. If there were not enough evidence, it will look for more information by cooperating with other LIDSs by means of the collaboration agent. For misuse detection, we will focus on the study of three types of attacks: Blackhole, Grayhole, and Selfish behaviour.

4.2.2.3 Response Agent

The response agent is a deliberative agent. Its main function is to react to the detected intrusions, as quickly as possible, in order to prevent further damage. It is also concerned with the update of both normal profiles and known attacks databases (cf. Chapter 5).

4.2.2.4 Collaboration Agent

The collaboration agent serves as a communication channel between the different LIDSs. Each of these local IDSs will have to communicate with other LIDSs in the network to convey information about the state of the network or to participate in a global intrusion detection and response.

4.2.2.5 Self-healing Agent

The healing agent is a stationary agent. Its main task is to perform the necessary actions for the healing of the network after an intrusion is detected. It has the ability to communicate with the other agents within the same LIDS. It uses its backup data and information collected by the detection agent (e.g., packet drop ratio, delay, victim node(s)' ID(s), intruder(s)' IDs, detection time, and so on) to measure the damage caused by the intruder(s). Then, building on the estimated level of damage, it will create and execute an appropriate list of actions to heal the network (cf. Chapter 5).

4.3 MASID vs. MANET resource constraints

Limited resource constraints such as energy, processing capacity, and memory are important features to, unavoidably, consider when designing an IDS for MANETs. To address this issue, we sought to adjust the behaviour of the agents within a LIDS to the node's state so that we can preserve system and network resources to the maximum possible. This is achieved by creating a kind of active/sleep transition in state (i.e., transition between 'active' and 'sleep' modes) for each of the agents. Every agent has two modes: 'sleep' and 'active'. Only one mode can be activated at a time for each agent. 'Sleep' refers to a state where the concerned agent is not performing any actions. In contrast, active state refers to the agent's state when performing the required funtions.
For example, the response agent is initially set to the 'sleep' mode. Whenever an intrusion is detected by the detection agent, this latter will activate the response agent (switches to 'active' mode). After performing the necessary response actions, the response agent will reset itself to the 'sleep' mode until new intrusion is detected.
Figure 4.4 illustrates which agents an agent can trigger or activate.

4.3.1 Case study

If we consider the example of Figure 4.5, we can see that at a certain moment (Figure 4.5 (a)) all the agents are in a sleep mode with the exception of data collection agent and the self-healing agent. Collector is responsible for continuous supervision of the network to collect relevant data to be used by the detection agent. The self-healing agent continuously generates, stores, and updates backckup data. After filtering collected data, so that to keep only the necessary information for the detection process; collector activates the detection agent to start the detection process (Figure 4.5 (b)). Figures 4.5 (c) and (f) illustrate the two possible scenarios resulting from the activation of the detection agent.

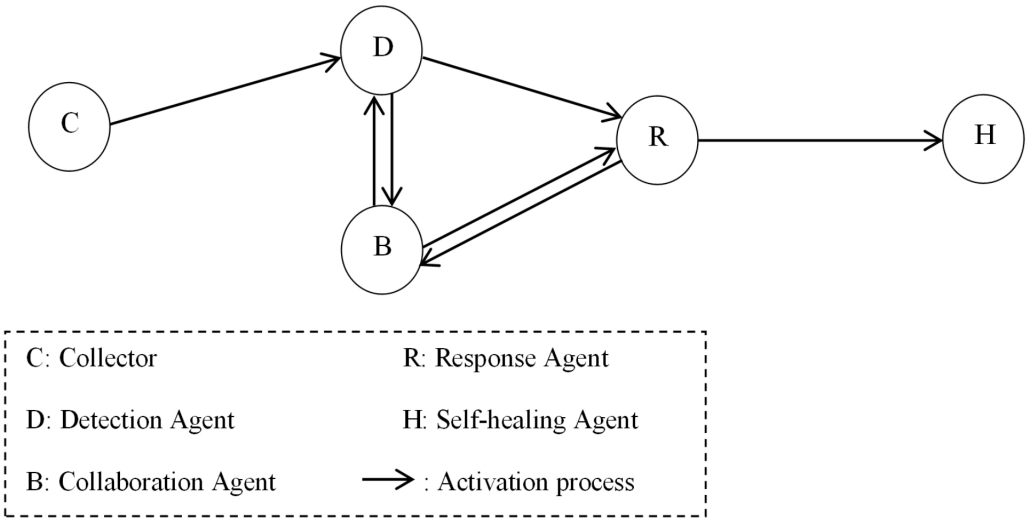


Figure 4.4: Agent activation

The first scenario presented in Figures 4.5 (c), (d), and (e) shows the case where the detection agent detects one of the known attacks. In this case, it will activate the response agent, to deal with the detected attack. This latter will activate the collaboration agent to inform the other network nodes and, if need be, it will trigger the healing agent. After performing the necessary actions, the activated agents will reset themselves to the sleep mode in order to preserve the system's resources.
If the detection agent detects an unforeseen state, only the collaboration agent will be activated to look for more information on neighbouring nodes as illustrates Figure 4.5 (f). The response agent is, latterly, activated (Figure 4.5 (g)) to either generate the appropriate response to the newly detected attack thereby, triggering the self-healing agent if necessary (Figure 4.5 (h)) or to update its databases.
In some cases and if necessary, it might happen that many agents become active at the same time. For example, the detection agent can become active while collaborator and/or the response agent are still active.
After performing the necessary actions, an activated agent resets its state to the sleep mode and maintains that state until new triggering event occurs.

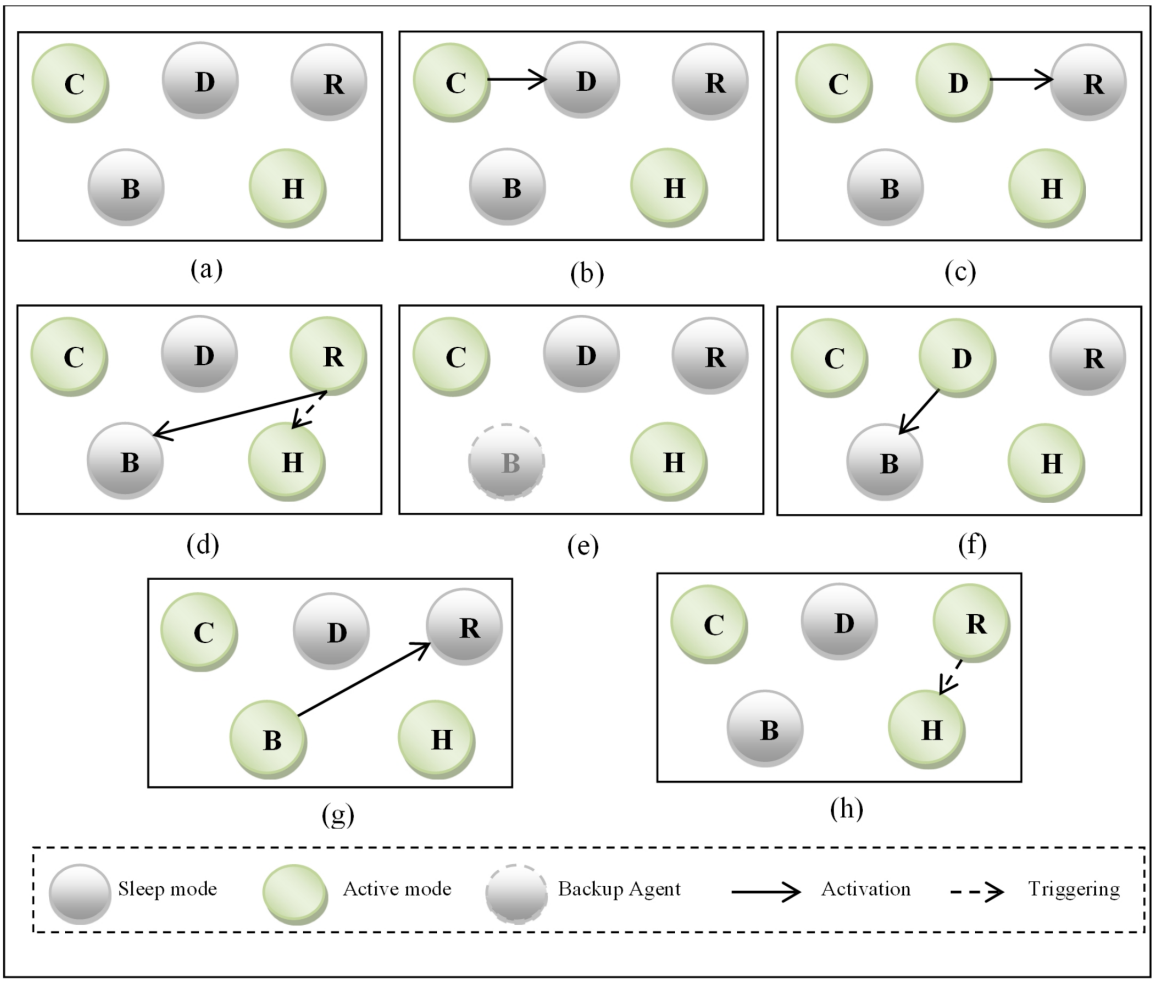


Figure 4.5: Sample Scenario of Intrusion Detection Using MASID

4.4 Discussion

The proposed system has the following properties:

- Automation of the detection and response processes through the distinguished agent properties like autonomy and pro-activeness.
- It presents no single point failure as no central entity is needed for data collection or detection.
- Intrusion detection and response is performed on every node thereby, avoiding problems like having a node or set of nodes overloaded with intrusion detection tasks.

- Decisions are made locally which reduces the detection time and allows for faster reaction.
- Mobile agents are used to handle communication and cooperation between the different LIDSs.
- Local intrusion detection and response is divided into subtasks handled by software agents which enhances the system's fault-tolerance and flexibility. Modifications to the system are made easier through this modular structure which entails more ease of maintenance.
- Having a distributed system reduces the chance of complete system crash. For instance, if the response agent ceases to function, the system can still detect intrusions correctly by means of the other agents.
- The distributed and cooperative nature of the proposed IDRS permits broader coverage and enhances detection rates through redundancy.
- Adaptive response generation reduces the chances of network partitioning and helps in detecting them in case of their occurrence.
- LIDSs rely mostly on their first-hand information for detections which enhances the reliability of the IDS.

4.5 Conclusion

This chapter introduced MASID, a new IDS for ad hoc networks. It performs an agent-based detection process in a distributed and cooperative manner. The main advantages of the proposed intrusion detection scheme can be summarized in the following points. First, no central entity is needed for data correlation or analysis. This increases the fault-tolerance of the system as no single point-of-failure is present. Second, more flexibility and a complete automation of the intrusion detection process were achieved through the use of agents. Finally, low consumption of both node and network resources. Detecting intrusions does not end the threat but, rather, detects the presence of potential intrusions. Therefore, intruders and the damage they caused are yet to be dealt with. This implies that the IDS should be able to provide not only a quick detection but also a rapid response to thwart the intruders and limit the potential damage. The next chapter stresses the notion of intrusion severity and proposes a new intrusion response approach that allows a timely and adaptive response to the detected intrusions.

Chapter 5

An Optimized Intrusion Response System for MANET [Mechtri2017]

5.1 Introduction

Attacks against MANET have been more sophisticated and misleading, which entails a great difficulty in preventing or even detecting them. Executed alone, intrusion detection detects the presence of anomalies and/or some specific node misbehaviour and may permit to identify and localize intruders but does not offer any options to stop the risk. This task is usually handled by dedicated intrusion response systems that can either be integrated within the IDS or independently work along the IDS. Upon detection of an intrusion, an IRS executes a set of actions called response that may range from passive to active. Passive responses are not suited for MANET environments where every node should react on its own since no form of centralized administration exists and nodes cannot rely on each other for their protection due to their mobility. Therefore, the integration of an active IRS is of major importance to successfully achieve the intended goals of intrusion detection in MANET.

In order to avoid negative response scenarios where the IRS might bring more harm than good, an active IRS should have the ability to adapt to the changing state of its environment. Adaptiveness allows the IRS to generate systematic and appropriate responses to mitigate the potential damage and to deter the misbehaving nodes without causing greater damages.

In this chapter, the notions of intrusion's severity-degree, cumulative severity-degree, and the severity index are introduced as new intrusion features based on which an IRS will be able to appropriately and systematically respond to the detected intrusions. The chapter also discusses some issues related to network partitioning and remerging and evaluates

the performance of the proposed IDRS.

5.2 The severity-aware approach

Active response systems are either fixed or adaptive. A static IRS adopts the same response no matter what type of intrusion is detected. Responding to all types of intrusions in a fixed manner would be inadequate mainly if coupled with a high false alarm rate. For instance, adopting a simple response allows more disruptions to take place. Conversely, a severe response like node elimination will result in discarding some innocent nodes and might lead to the disruption of some network functionalities like losing connectivity, congestion, and an increased network latency.

Thus, it is up to the response agent to decide about the way in which it will deal with the detected intrusions so as to mitigate the potential damage without causing more damage. In other words, there should be a certain mechanism to adjust the generated responses to minimize the damages to the maximum possible.

This mechanism is introduced to MASID by extending the response agent's databases with the notion of severity-degrees. The resulting IDRS is called MASID-R-SA, hereinafter. This extension means that the response agent will distinguish between the detected intrusions according to their estimated severity levels and their distribution over time.

Formally, the Severity (S) of an intrusion (A_i) is measured in terms of its potential damage (D_{A_i}) and, in case of several occurences, their distribution over time as specifies equation 5.1:

$$S(A_i) = f(D_{A_i}, t) \tag{5.1}$$

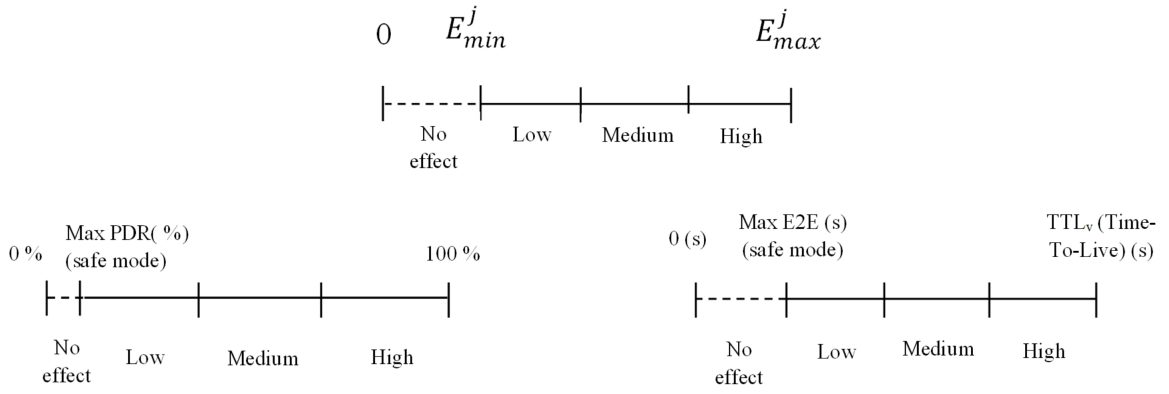
The following subsections detail the process of severity-degrees' assignment, the generation of adaptive responses, and discuss some incurred problems.

5.2.1 Autonomous severity assessment

Definition 5.1 A node's misbehaviour can have no-effect, Low, Medium, or High effect on a specific performance metric. The severity level (SL) of an intrusion (A_i) with respect to a metric M_j represents the degree of damage it causes in terms of that metric. It can be interpreted, mathematically, as (equation 5.2):

$$SL(A_i)/M_j = e_i^j, \quad e_i^j \in [0, E_{max}^j], \tag{5.2}$$

where:
E^j_{max} denotes the maximal value of damage (in terms of the considered metric M_j) that can be caused by an intrusion.
In this concern, we need also to define E^j_{min} ($0 \leq E^j_{min} < E^j_{max}$) that represents the minimal value of damage (in terms of a considered metric M_j) caused by a node to consider it as committing an intrusive act.
E^j_{min} corresponds to the maximal level of tolerated damage.
Figure 5.1 illustrates how severity levels are specified with respect to E^j_{min} and E^j_{max}.



Eg. 1. Packet drop: misbehaviour zone for packet drop is comprised between the maximal value of tolerated packet drop in a safe mode and the potential maximal value of packet drop that can be caused by a misbehaving node.

Eg. 2. E2E delay: misbehaviour zone for the E2E delay is comprised between the maximal value of E2E delay that can be marked by a network in a safe mode and the TTL_v value, which characterizes the upper bound of time that a packet can spend in the network.

Figure 5.1: Severity Levels Assignment

Definition 5.2 Severity-Degrees (SD) are real numbers assigned to intrusions according to their severity levels. Let n be the number of the considered security metrics and m be the number of intrusions (this includes both known intrusions and intrusions detected by the anomaly detection engine).

To assess the severity of an intrusion A_i ($i = 1\ to\ m$), MASID-R-SA calculates its Severity Degree, written $SD(A_i)$ using equation 5.3:

$$SD(A_i) = \sum_{j=1}^{n} W_j \times R_{ij} \tag{5.3}$$

This is a simple weighted sum where values of W_j and R_{ij} ($i = 1\ to\ m; j = 1\ to\ n$) are specified as follows.

Definition 5.3 Several metrics can be used to judge the severity of an intrusion and that depending on the objectives of the adopted security approach. Example metrics are: energy consumption, memory usage, packet loss rates and latency in delivering packets. For instance, if the main goal of the adopted security mechanism is to ensure high packet delivery level i.e. to guarantee that no (or the least possible) packets are to be dropped, then the IDRS will consider the attacks that cause higher packet dropping as the severest and thereby, they will be assigned the highest severity-degrees.

Therefore, a weight of importance $W_j(j = 1\ to\ n)$ is assigned to each performance metric M_j such that:

$$0 \leq W_j \leq 1 \qquad and \qquad \sum_{j=1}^{n} W_j = 1 \tag{5.4}$$

A weight W_j characterizes the importance of a metric M_j in the decision making about the intrusion's severity.

Definition 5.4 Intrusions' effect on network performance can be normalized and expressed in the form of ranks. The ranking matrix R is an $m \times n$ matrix in which a rank R_{ij} indicates the level of performance degradation caused by an intrusion A_i ($i = 1\ to\ m$) with respect to the performance metric M_j ($j = 1\ to\ n$).

The ranking matrix' data can be represented as follows:

$$R = \begin{array}{c} \\ A_1 \\ A_2 \\ \vdots \\ A_i \\ \vdots \\ A_m \end{array} \begin{array}{c} \begin{array}{cccccc} M_1 & M_2 & \cdots & M_j & \cdots & M_n \end{array} \\ \begin{pmatrix} R_{11} & R_{12} & \cdots & R_{1j} & \cdots & R_{1n} \\ R_{21} & R_{22} & \cdots & R_{2j} & \cdots & R_{2n} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ R_{i1} & R_{i2} & \cdots & R_{ij} & \cdots & R_{in} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots \\ R_{m1} & R_{m2} & \cdots & R_{mj} & \cdots & R_{mn} \end{pmatrix} \end{array}$$

Normalization of each rank is obtained using linear scaling. Values of R_{ij} are assigned according to equation 5.5:

$$R_{ij} = \begin{cases} \frac{e_i^j - E_{min}^j}{E_{max}^j - E_{min}^j} & \text{if} \quad e_i^j > E_{min}^j \\ \epsilon & \text{if} \quad 0 < e_i^j \leq E_{min}^j \\ 0, & \text{otherwise} \end{cases} \tag{5.5}$$

In order to maintain consistency among the different LIDSs, network nodes should be informed about any update in the severity-degrees. Therefore, the severity-degree of a newly discovered intrusion will be communicated to all other network nodes while being informed about the intrusion incident by the collaboration agent.

5.2.2 Adaptive response generation

An adaptive system is characterized by its ability to adapt its behaviour according to changes in its environment or in parts of the system itself [43].

MASID-R-SA is an adaptive IDRS that adapts the generated responses to the damage caused by the detected security incidents. Thus, the way in which it responds to intrusions depends on the estimated severity-degrees. Basically, the generated responses are adjusted so as to respond to intrusions that have a low severity-degree with simple responses while severe responses are reserved to severe intrusions (those with a high severity-degree).

Simple responses refer to a temporary cut of the connection to the potential intruder. The punishment period is exponential to the caused damage which entails that response severity will increase with the increase in the intrusions' severity. Severe responses are meant to completely and permanently cut the connection to the potential intruder.

Definition 5.5 The distinction between simple and severe intrusions is done automatically based on a predefined threshold. The value of this threshold, referred to as the Severity-Index (SI), is defined based on the intrusions' potential damage as mentioned in equation 5.6.

$$SI = \frac{\sum_{j=1}^{n} W_j \times R_{max}^j}{2 \times \sum_{j=1}^{n} W_j} \tag{5.6}$$

If the severity-degree of an intrusion exceeds the severity-index, then a severe response should be adopted, else a simple response would be sufficient.

However, if a malicious node is accused of carrying out simple intrusions for a certain number of times then a simple response would no longer be sufficient and a severe response should be undertaken instead. In that case, the cumulative damage (CD) is used to reflect the total damage caused by those intrusions to the network.

Definition 5.6 we define the cumulative damage of a set of simple intrusions initiated by the same intruder as a function of the total damage incurred by those intrusions and their distribution over time. The Cumulative damage of a set $\mathcal{A}_p$ of p ($p \geq 2$) intrusions with respect to a specific metric M_j ($j = 1$ *to* n) is defined as (equation 5.7):

$$CD_p = CD_{p-1} + D_p \times (1 - \tau) \tag{5.7}$$

Where,
CD_{p-1}: the cumulative damage of the first $(p-1)$ intrusions.
D_p: the damage caused by the p^{th} intrusion.
τ: a scaled value that characterizes the distribution of intrusions over time. To estimate the value of τ, linear scaling is used as follows:

$$\tau = \frac{\Delta T - DW}{T_{max} - DW} \tag{5.8}$$

where
ΔT: represents the length of time between the time of detection of the last intrusion and its predecessor.

$$\Delta T = T_p - T_{p-1} \tag{5.9}$$

DW: is the length of the active detection window and T_{max} is the period after which two instances of an intrusion are considered as independent.

Definition 5.7 The cumulative severity degree (CSD) is calculated following the same steps for calculating the severity degree of a single intrusion. Here, the cumulative damage is used instead of the damage caused by a single intrusion.

A major disadvantage of generating severe responses is the possibility of network partitioning. The following subsection discusses this problem with respect to the proposed approach.

5.3 Network partitioning problem

Network partitioning can be defined as the split of a network into several disconnected sub-networks called partitions. This can occur due to node mobility or the failure of some parts of the network.

As a special form of node failure, the elimination of nodes accused of committing intrusive acts can also lead to network partitioning. This is mainly characterized by the presence of intruders serving as gateway nodes in the network.

Definition 5.8 A gateway node is a node (or set of nodes) that serve as the only connection between two or more network partitions. The deletion of these nodes would result in network partitioning.

The generation of adaptive systematic responses reduces the possibility of suffering partitions incurred by typical response systems. For instance, adopting severe responses like node isolation leads to the elimination of all suspected nodes including those of false positive detections. However, the problem persists since the elimination of one intruder (a bridge node) may cause the network to partition (depending on the network's topology at the time of response execution) as illustrated in Figure 5.2.

Network partitioning affects the correct functioning of distributed applications like distributed and cooperative IDRSs that rely heavily on information sharing and the collaboration of different nodes in detecting and responding to intruders. For instance, a split in the network might lead the IDRS to break down or to become inconsistent when the network remerges.

In the case of MASID-R-SA, information about newly detected intruders or an update in the intrusion signature base of a LIDS should be communicated to all other LIDSs in the network. However, a split in the network will prevent updates from reaching LIDSs in other partitions; thereby MASID-R-SA's state might become inconsistent on remerging.

A simple solution to overcome such faults would be to add a Detection History Base (DHB) to every LIDS in MASID-R-SA so that to enable the nodes to keep track of all intrusion-related events in their environment (network or partition). Information contained in the DHB include:

(i) Intruder_ID.

(ii) Maliciousness_factor: the number of times the intruder was accused of committing simple intrusions.

(iii) Intrusion_type: it specifies the type of the detected intrusion (eg., blackhole, grayhole, selfish, or unknown)

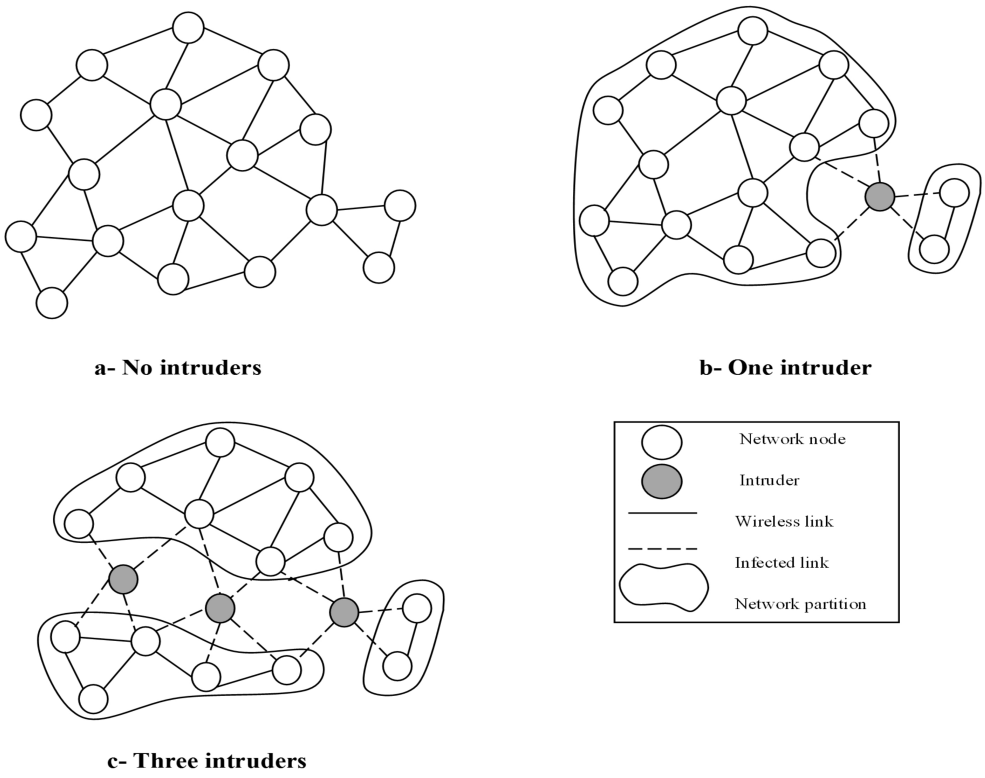


Figure 5.2: Effect of severe responses on network connectivity

(iv) Detection_time: the time at which the intrusion was first detected.

(v) Response: it specifies whether the generated response is severe or simple.

(vi) Punishment_time: it specifies the time at which the punishment period ends. It is set to -1 for severe responses and to T_{end} (equation 5.10) for simple responses where:

$$T_{end} = Detection\_time + punishment\_period \tag{5.10}$$

The DHB is updated whenever an intrusion is detected and it is also used whenever a new node joins the network or after a partitioned network remerges. More specifically, information contained in the DHB is used to provide incoming nodes with an up-to-date summary of the security status of the network (or network partition).

On network remerging, a node, denoted the DHB_node, from every partition is selected by vote or based on a selection algorithm to communicate its DHB to the other DHB_nodes. Based on information they receive, DHB_nodes can decide whether to start a DHB update process within their own partitions or not. Thus, intrusion detection related histories will be merged, generating a general DHB covering intrusion related

events within all parts of the network.

However, a node should have the ability to verify that DHB updates do not introduce any violation of integrity. In other words, there is a need to ensure integrity while allowing legitimate updates to the DHB. Similar issues have been widely discussed in the literature and a variety of solutions have been proposed. For instance, a trust management system can be adopted to ensure integrity of the DHB. The trust management system can assist nodes to decide whether to trust updates proposed by other nodes or partitions.

To evaluate the proposed approach, we have chosen three well-known routing intrusions and studied the effects, on the node and the network, of each of them separately to be able to decide about their severity-degrees and to define the different thresholds needed in both intrusion detection and response. These intrusions are: blackhole, grayhole, and the selfish behaviour attack. The results of this study are presented in the subsequent section.

5.4 Experiments and results

In order to evaluate the proposed approach, we carried out a series of simulation experiments using the network simulator NS-2 [67]. In these experiments, the proposed IDRS is validated against three routing attacks, namely: blackhole, grayhole, and selfish behaviour.

The following sections detail the simulation environment and metrics, the considered threat model, and discuss the obtained results.

5.4.1 Simulation Environment and Parameters

5.4.1.1. Simulation tool and environment

Many simulation tools are available for wireless ad hoc networks [45] such as Network Simulator (NS) and Riverbed Modeler (formerly known as OPNET) [64]. In order to evaluate our approach, we simulated a MANET using NS-2. It is an object oriented discrete event simulator, written in C++, with an OTcl (Object-oriented Tcl) interpreter as a frontend. It can simulate both wired and wireless network systems.

5.4.1.2. Simulation settings

As mentioned before, NS-2 is used to simulate an ad hoc network consisting of 50 nodes and under the threat of several intrusions. Each node in the network is assigned an initial position within a simulation area of $(1000m \times 1000m)$ square meters and joins the network at random. The MAC (Medium Access Control) layer used for the simulations is IEEE

802.11 and the packets are generated using CBR (Constant Bit Rate). Additional parameters needed to build the simulated network are presented in Table 5.1. The performance of the proposed system is measured under different attack scenarios and a variety of traffic loads (ranging from 2% to 50% source nodes) as specified in the following subsections. The simulation takes place for 1200 seconds. All simulation results are averaged over 10 rounds of simulation runs.

Parameter	**Value**
Simulator	*ns*-2 (version 2.34)
Simulation time	1200 s
Number of nodes	50
RP. for legitimate nodes	AODV
RP. for blackhole nodes	blackholeAODV
RP. for grayhole nodes	grayholeAODV
RP. for selfish nodes	selfishAODV
Traffic model	Constant Bit Rate (CBR)
Transport protocol	User Datagram Protocol
Terrain area	$1000m \times 1000m$
Transmission Range	250 m
Maximum bandwidth	2 Mbps
Nb. of source nodes	2%, 10%, 30%, 50%
Nb. of malicious nodes	variable

Table 5.1: Simulation Parameters

5.4.1.3. Threat models

In our threat model, we consider an AODV-based MANET environment, where adversaries are part of the network and can launch attacks simultaneously. Our goal is then to protect the network from the threat of intrusions and to minimize the undesirable effects of the generated responses.

More specifically, we focus on three routing attacks: blackhole, grayhole, and selfish behaviour attacks. In this concern, we assume that the malicious nodes carrying out these attacks are called: blackhole node, grayhole node, and the selfish node, respectively. In this study, the simultaneous attacks' scenarios are characterized by the co-occurrence of different attacks in the same part of the network. However, we assume that these attacks are not colluding i.e., there is no conspiracy among the attacking nodes to launch their attacks simultaneously. Different multi-attack scenarios including combinations like the following are used in the conducted experiments:

- 1 blackhole node, 1 grayhole node, and 1 selfish node.
- 1 grayhole node and 2 selfish nodes.
- 1 blackhole nodes, 2 grayhole nodes, and 1 selfish node.

We simulated the blackhole, grayhole, and the selfish behaviour attacks by modifying the original AODV routing protocol. The following subsections provide a brief description of these attacks with regard to the AODV routing protocol.

A. Blackhole attack

AODV considers RREP messages that have the highest value of the destination sequence number to be the most recent routing information and selects the route contained in that RREP for the current communication session. Realizing that, the blackhole node will always respond to the received RREQ by sending a RREP with the highest possible value of the destination sequence number and the smallest value of hop-count. In our simulations, for instance, we set the value of the sequence number of the RREP packet generated by blackhole nodes to 4294967295, and the hop-count is always set to 1. In this way, the node sending the RREQ considers the path through the attacker as the best path and uses it to route data packets to the intended destination. Eventually, the blackhole node will drop the received data packets instead of relaying them as the protocol requires, as specified by Algorithm 1.

Algorithm 1 BlackholeAODV: Receive Data Packet Function

```
1: begin
2: Upon receiving a new data packet:
3: if Destination_ID == Blackhole_Node_ID  then
4:    Process the received packet
5: else
6:    Drop the received packet
7: end if
8: end
```

B. Grayhole attack

In terms of the AODV routing protocol, the grayhole node replies with a falsified RREP claiming it has the shortest fresh path to the destination each time it receives a RREQ from any other node in the network. In this way, the source considers the path through the grayhole node and uses it for all data flow between it and the destination node. Then, the grayhole node selectively drops some of the traffic passing through it. For instance,

the packet drop exhibited by the grayhole node may be meant to drop some specific packets or to randomly drop some packets. Algorithm 2 illustrates one of the possible implementations of the selective packet drop adopted by the grayhole node.

Algorithm 2 GrayholeAODV: Random Packet Drop

```
1: begin
2: Upon receiving a new data packet:
3: if Destination_ID == Grayhole_Node_ID then
4:    Process the received packet
5: else
6:    if rand()%2 == 0 then
7:       Drop the received packet
8:    else
9:       Forward the packet to its destination
10:   end if
11: end if
12: end
```

C. Selfish behaviour attack

During the path discovery process in an AODV-based MANET, the source node broadcasts a RREQ to look for a route to the intended destination. Upon receiving the RREQ, the neighbours of the source node forward the received RREQ to their neighbours and so forth until reaching the destination node or a node that has a valid route to the destination. Unfortunately, if it were a selfish node that receives the RREQ, it prefers not to participate in this process unless it is concerned with it. That is, it may reject all the received RREQs that are not aimed to it or simply not forward RREPs or worse yet, it may participate correctly in the route discovery process then it refrains from forwarding data packets. Therefore, there are several possible implementations for selfish behaviour. Algorithm 3 presents one of these implementations.

Algorithm 3 SelfishAODV: Receive Data Packet Function

```
1: begin
2: Upon receiving a new data packet:
3: if Destination_ID == Selfish_Node_ID then
4:    Process the received packet
5: else
6:    Discard the received packet
7: end if
8: end
```

The impact of the selfish behaviour attack is studied using three different scenarios

according to the nodes' density within the same area of the selfish node. These scenarios represent cases where the victim node(s) has, other than the selfish node, (1) several, (2) few, or (3) no neighbour nodes.

5.4.1.4. Evaluation Metrics

To validate the efficiency of the proposed approach, we consider the following metrics:

(i) Average attack success rate (ASR)

The attack success rate for both blackhole and grayhole attacks is measured in terms of the ratio of the number of times the attacker is selected to be a multicast forwarding member to the number of times the route discovery process is initiated. Contrary, the success rate of the selfish behaviour attack is the ratio of the number of times the selfish node is not selected to be a multicast forwarding member (although it must be) to the number of times the route discovery process is initiated.

This metric characterizes the completeness and correctness of the simulated attacks.

$$ASR = \frac{Number\ of\ routes\ (not)\ containing\ the\ attacker}{Number\ of\ connections}$$

(ii) Packet delivery ratio (PDR)

Packet delivery ratio designates the ratio between the number of packets originated by the application layer CBR sources and the number of packets received by the final destination. Packet delivery ratio is important as it describes the loss rate that will be seen by the transport protocols.

$$PDR = \frac{\sum received\ packets}{\sum sent\ packets}$$

(iii) Average End-to-End Delay (E2E)

it represents the average time taken by a data packet to arrive to its destination. This includes all delays caused during route acquisition and buffering at intermediate nodes. Only data packets that are successfully delivered to their destinations are counted.

Lower value of end-to-end delay means better performance of the studied protocol.

$$E2E = \frac{\sum(arrive\ time - send\ time)}{number\ of\ connections}$$

(iv) Average Hop Count (AHC)

AHC represents the average number of hops that were traversed by each data packet from source to destination.

$$AHC = \frac{\sum hop\ count}{\sum received\ packets}$$

(v) False Positive Rate (FPR)
FPR represents the ratio of normal profiles that are considered as attacks.

(vi) False Negative Rate (FNR)
FNR represents the ratio of attacks that are not successfully detected.

(vii) True Detection Rate (TDR)
TDR represents the ratio of both learned normal profiles and attacks that are successfully identified as normal profiles or intrusions (TPR), respectively.

(viii) Response Rate
It represents the ratio of intrusions that were successfully responded to. This factor qualifies the adaptiveness of the response system.

$$RR = \frac{\sum successful\_responses}{\sum detected\_intrusions}$$

5.4.2 Experimental Results

In this section, we present and discuss the results of our study. As mentioned earlier, we used ns-2 and simulated an ad hoc network consisting of 50 nodes. We introduce malicious nodes in the network in the form of blackhole, grayhole, and selfish behaviour nodes.

Figures 5.3, 5.4, 5.5, and 5.6 present a comparison of the considered intrusions in terms of: attack success rate, Packet delivery ratio, average end-to-end delay, and path length, respectively (No protection is provided here). This comparison is handled with respect to a variation of traffic loads for four different scenarios. The first scenario represents an AODV-based network in the safe mode. The second, third and fourth scenarios represent three networks having, in addition to their legitimate nodes, one malicious node (blackhole, grayhole, and a selfish node, respectively).

Figure 5.3 shows the success rates of the simulated intrusions with respect to different traffic loads in the absence of IDS protection. Since the main goal of a blackhole or grayhole node is to drop the absorbed network traffic, its success will certainly be affected by the variation in the network traffic load. For instance, it is observed that the success rate of both attacks decreases slightly due to the increase in traffic load. The main goal of a selfish node is, however, to not be implicated in any network operation unless it is concerned with it. Here, it is observed that this could always be achieved regardless traffic loads or node densities.

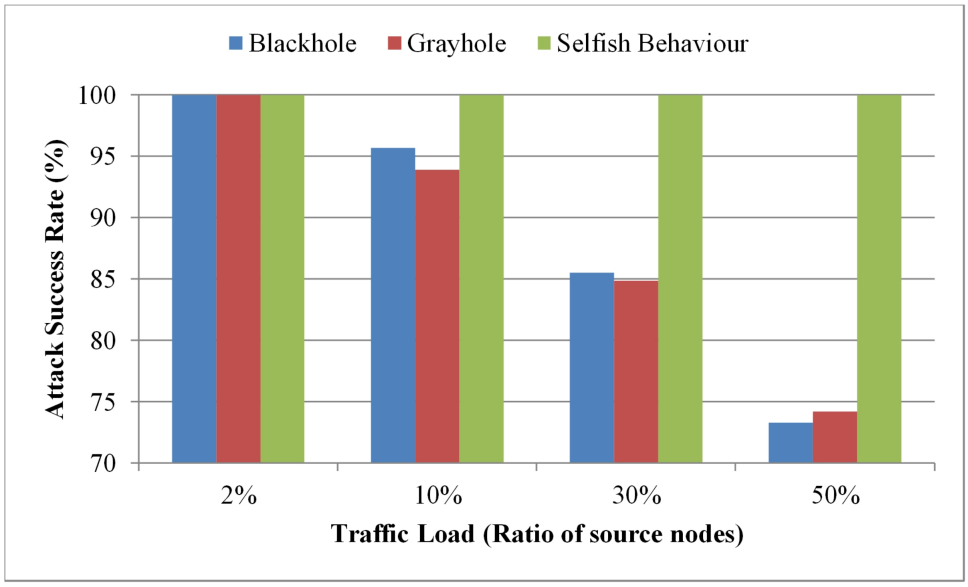


Figure 5.3: Average attack success rate in the absence of MASID-R-SA

Results for packet delivery in an AODV-based network under the threat of a blackhole, grayhole, and the selfish behaviour attack are presented in Figure 5.4. A safe mode scenario is also considered here in order to highlight the negative effect of intrusions on network performance.

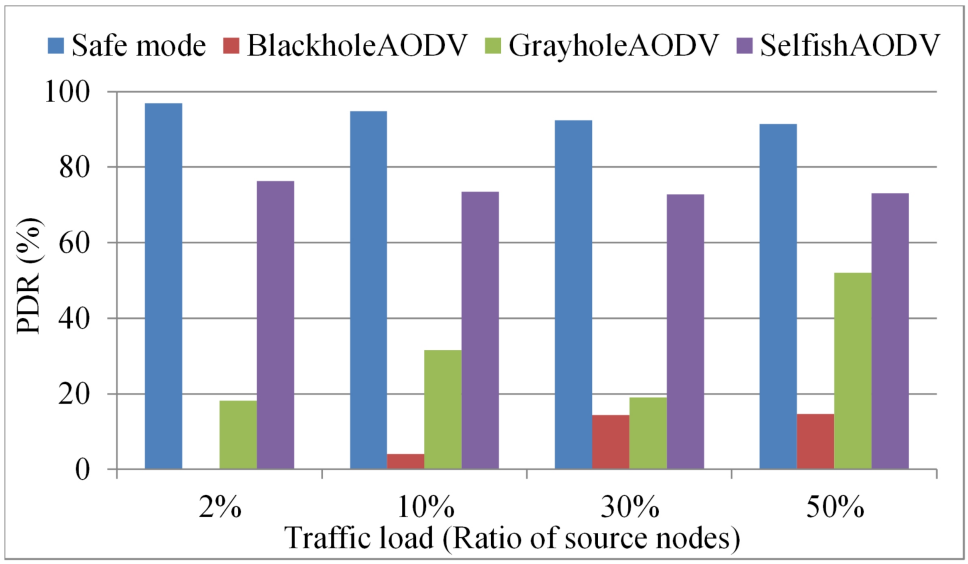


Figure 5.4: Packet delivery ratio in the presence of single attacks

It is clear that PDR degrades considerably in the presence of the blackhole attack

mainly in cases where low traffic loads are generated since it becomes easy for the blackhole node to manage to absorb all the generated traffic. Equally, PDR drastically goes down to range from 16% to 51% when the network is under the grayhole attack. The cause of this change in PDR is that the grayhole node is configured to drop the packets with no fixed probability (i.e., a random packet drop). However, a better delivery ratio was marked when the intruder is the selfish node because PDR is affected only if the only route to the destination is through the selfish node.

Figure 5.5 illustrates the evolution of the end-to-end delay with respect to a variation of traffic loads. It is clear that the end-to-end delay is reasonably low except for the case of a selfish node present in the network. Actually, the non-cooperation of the selfish node causes legitimate nodes within the same area to process more traffic loads, which might lead to more collisions and congestion in the buffers of intermediate nodes. However, the decrease in the end-to-end delay in cases of blackhole and grayhole attacks is due to the fact that a great part of the generated packets were dropped, by intruders, which means that fewer packets are going to traverse the route to the final destination thereby allowing them to reach their destinations faster.

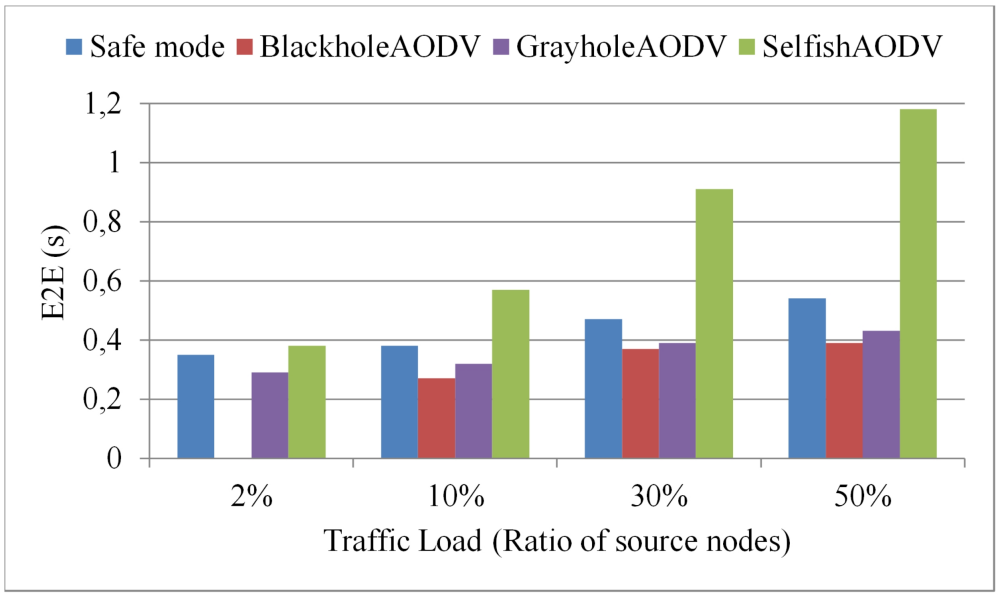


Figure 5.5: Average E-2-E Delay in the presence of single attacks

Similarly to the end-to-end delay, path length increases due to the selfish node's non-cooperation and decreases slightly due to blackhole and grayhole attacks as illustrates Figure 5.6.

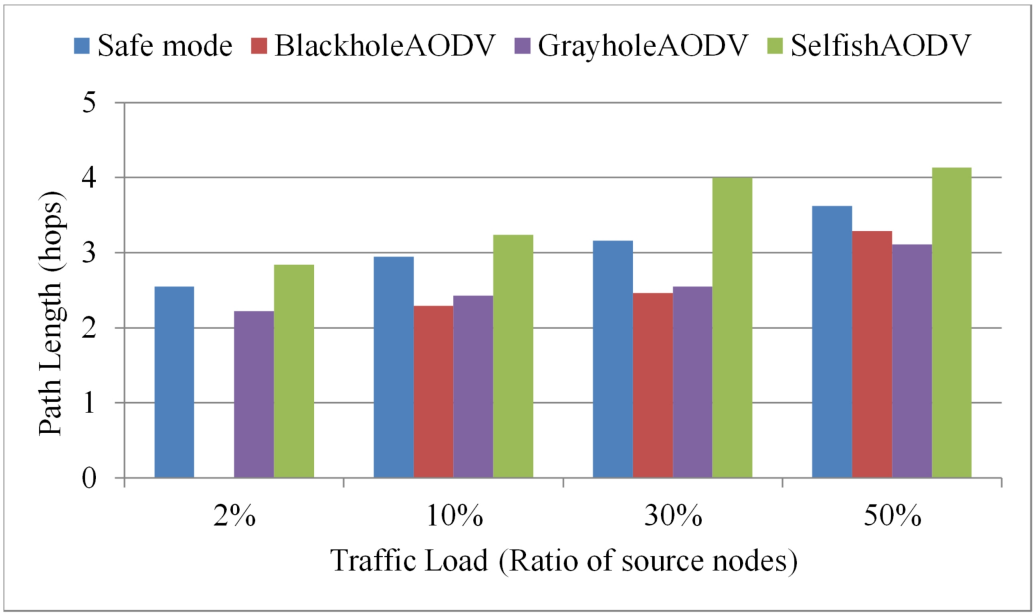


Figure 5.6: Path length in the presence of single attacks

This initial simulation study, with respect to the considered evaluation metrics, revealed that network performance degrades in the presence of all the studied intrusions, but the degree of their impact differs significantly. More specifically:

- When comparing the obtained results for packet delivery under the threat of the different studied attacks, it can clearly be seen that the delivery ratio decreased under the threat of all the studied attacks but the steepest packet dropping rates, regardless special cases, were marked by the blackhole attack.
- The selfish behaviour attack increased significantly the delays of transmitting data packets compared to the blackhole and grayhole attacks where delays were reduced.
- Path length (in terms of the average hop-count) is almost not affected by the blackhole and grayhole attacks, but it increased relatively due to selfish nodes.

Table 5.2 presents the results obtained from the simulation study of MASID-R-SA against different single attack scenarios.

The proposed IDRS has effectively detected the simulated intrusions with very low false positive and false negative rates. However, the high false negative ratio (3.1 %) marked by the IDS while the network is being attacked by a selfish node is due to the fact that in case of a network with a high node density, the selfish behaviour attack has almost no effect on network performance thus, little or no signs of this attack can be found.

Intrusion	**TDR** (%)	**FPR** (%)	**FNR** (%)
Blackhole	99.80	0.0	0.2
Grayhole	99.40	0.4	0.6
Selfish Behaviour	96.90	0.0	3.1

Table 5.2: Detection rates of single intrusions

Therefore, its detection becomes really a hard task. Fortunately, the damage caused in such cases is tolerable and does not significantly affect the network. In contrast to node density, the increase in traffic loads does not prevent the selfish attack but it helps greatly in its detection as shown in Figure 5.7.

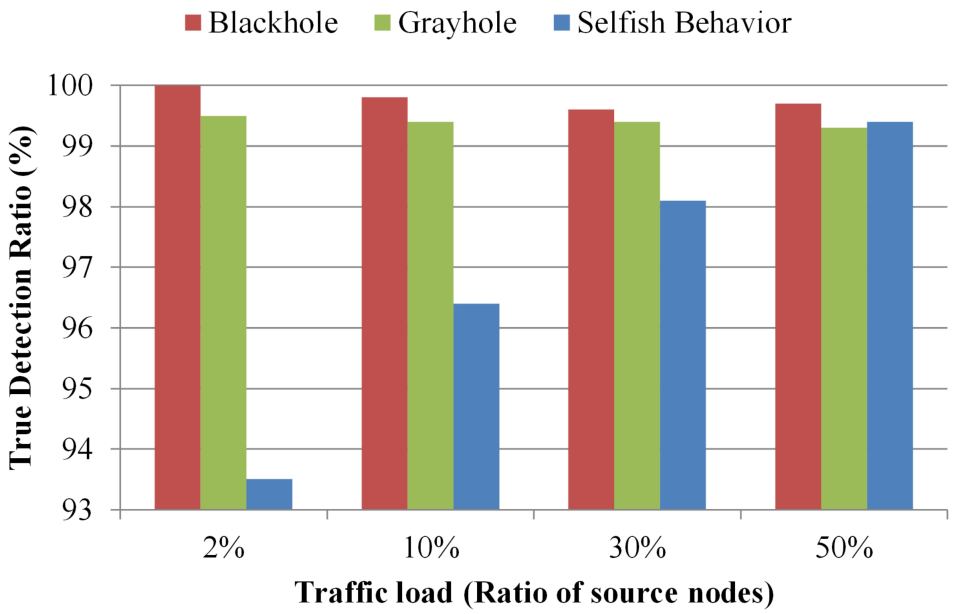


Figure 5.7: True Detection Ratio of single intrusions

Results for the detection of simultaneous intrusions for both MASID (generating one fixed response consisting of the elimination of the intruder) and the improved IDRS (MASID-R-SA, generating responses with respect to the severity of intrusions) are summarized in Table 5.3.

IDS	Detection Rate	Response Rate (%)		
		Blackhole	Grayhole	Selfish
MASID	97.23	93.15	90.45	95.24
MASID-R-SA	97.23	99.00	97.8	94.00

Table 5.3: Detection rates of simultaneous intrusions

The results show that the severity-aware approach helps greatly in increasing our IDS' ability to respond to all the detected intrusions. Detailed results of this evaluation are presented in Figure 5.8. For MASID-R-SA, responses are generated according to the intrusions' severity level and their cumulative severity level in case of several related occurrences. That is, the priority in responding to intrusions is always given to the intrusion that has the highest severity-degree, then to the one with a lower severity-degree and so forth.

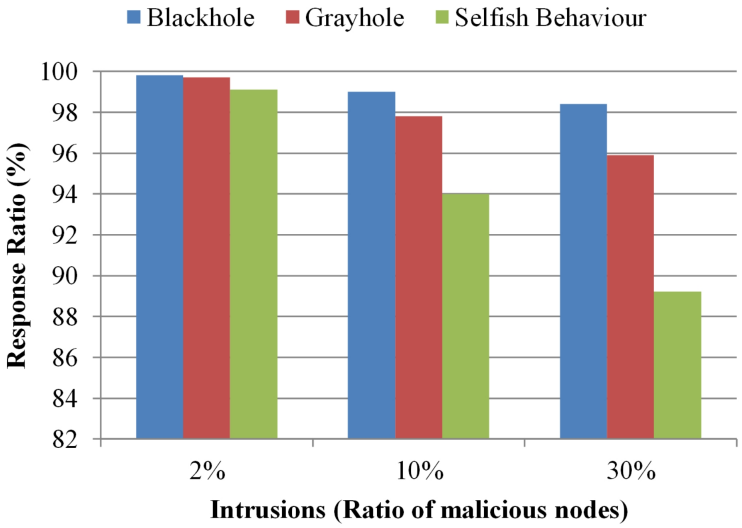


Figure 5.8: Response Ratio under the Threat of Simultaneous Attacks

Moreover, MASID-R-SA generates acceptable false positive and false negative rates as illustrates Figure 5.9.

The main cause of the non-detection of some intrusions (represented in terms of the false negative_special cases ratio) is the complete absence of signs (in terms of the studied parameters) of a network being attacked by a selfish node when being launched in a

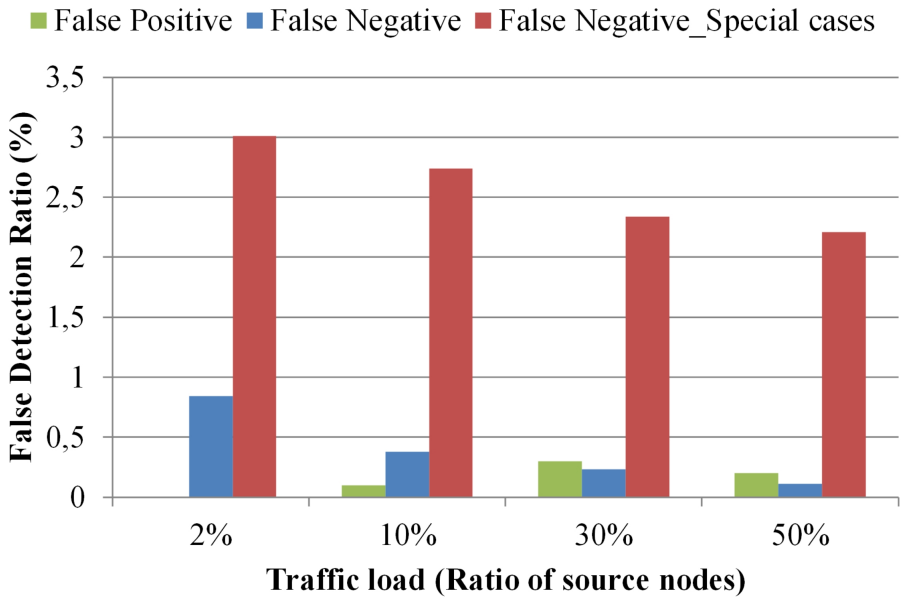


Figure 5.9: False detection ratios for MASID-R-SA under the threat of simultaneous intrusions

network with low traffic load or if the selfish node is present in a heavily populated area of the network. Thus, within the limits of special cases, FNR is too small.
The increase in the number of nodes in the network reduces the possibility of an attack to take place successfully. By the way, redundancy that results from the presence of several IDRSs within the same area increases detection accuracy and minimizes the effects of false detections.

More detailed results showing the impact of traffic load variation on both packet delivery ratio and the end-to-end delay are presented in Figure 5.10 and Figure 5.11, respectively. This impact is measured for four different scenarios. The first one represents an unsecured AODV-based network under the threat of both single and simultaneous intrusions while the second and third scenarios characterize a network imposed to the same threats but this time network nodes use MASID and MASID-R-SA, respectively, to guarantee their protection. The fourth scenario, however, represents an AODV-based network in its ideal situation i.e., a no-attack scenario.

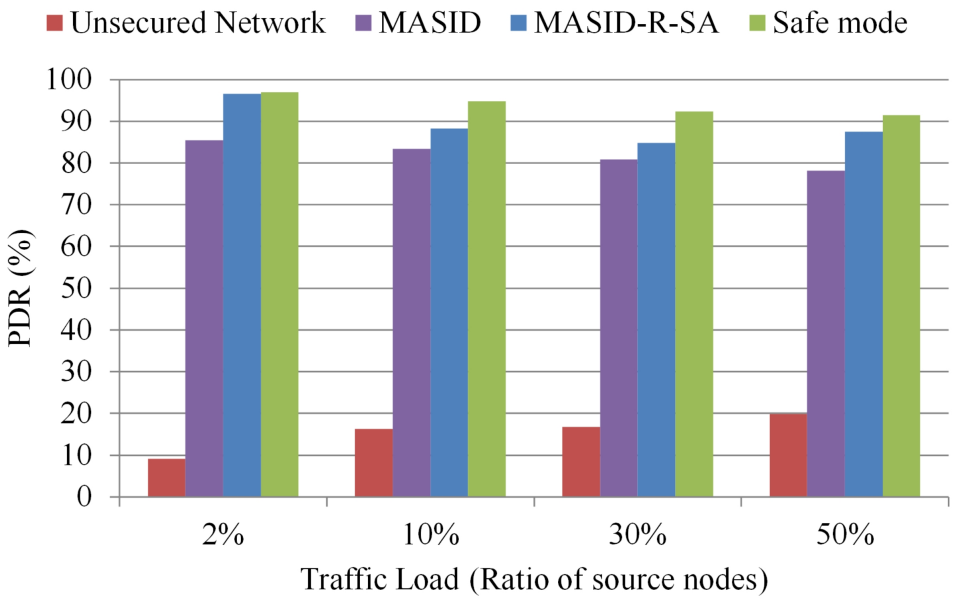


Figure 5.10: PDR in the presence of Simultaneous Attacks

Figure 5.10 shows that thanks to MASID-R-SA, the delivery ratio increased considerably to yield to nearly a safe network's delivery ratio. This increase is nothing but the result of its accurate detection and systematic response to intrusions. The delivery ratio achieved by MASID is slightly lower than the one achieved by MASID-R-SA as it adopts a fixed response to the detected intrusions while MASID-R-SA with its improved response module gives more importance to the severity of both single and correlated intrusions thereby, reducing the potential damage of generated responses to the maximum possible.

Figure 5.11 shows that a significant reduction in the end-to-end delay caused by the different intruders (about 14.53 % in average) is obtained when using MASID-R-SA. Furthermore, the achieved delay is highly appreciated as it approaches the one generated by a network in its ideal state. However, it is also possible to have a delay that is less than the delay of a safe network due to blackhole and grayhole attacks that usually reduce the delivery delays because of their packet dropping nature. That is why MASID often shows slightly shorter delays.

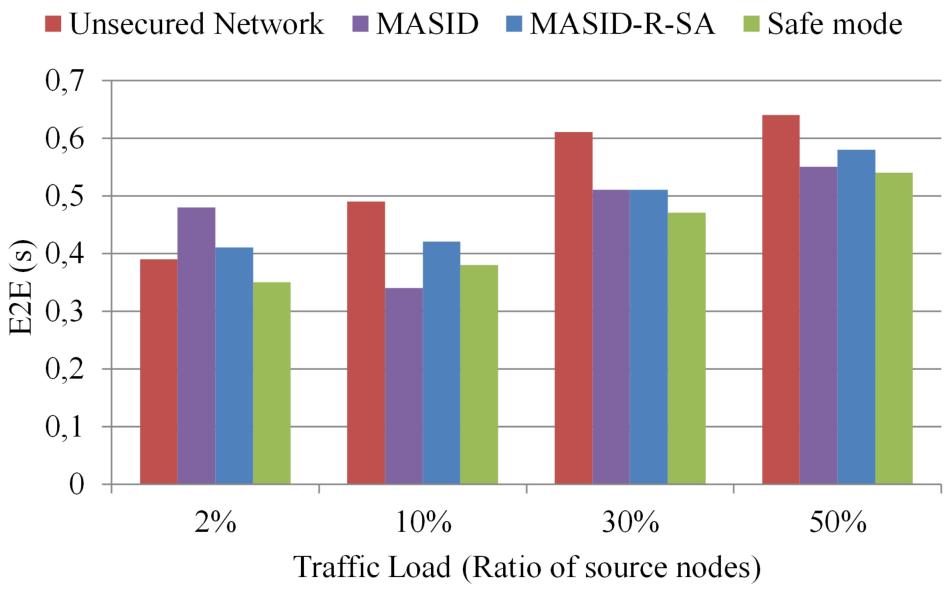


Figure 5.11: E2E Delay in the presence of Simultaneous Attacks

5.5 Conclusion

This chapter presented a new response mechanism as an extension to MASID to allow a timely and adaptive generation of responses to thwart intruders. This mechanism is based on the IDS' ability to assess the severity of any detected intrusion (both known and unknown). It also considers cases of correlated intrusions and adjusts responses accordingly. The severity-aware approach reduces the negative effects of false positive alarms. It also reduces the possibility of network partitioning and the IDS is made resilient enough to maintain its consistency even after network remerging.

Chapter 6

MANET Survivability Reinforcement using self-Healing[Mechtri2017]

6.1 Introduction

Distributed systems are by nature fault-prone. The situation becomes more complex in the presence of intrusions that continue to grow in both number and severity, especially in open environments like MANET. Even in the presence of IDRSs that play a great role in reducing the risks of disruptions and failures entailed by intrusions, the obtained security level does not always guarantee that the network is completely free of faults and malfunctioning. More specifically, some intrusions might succeed in causing considerable damages to the nodes or network services being targeted before being detected and completely removed. For that, systems and networks should be designed so that to survive such situations and to autonomously heal any potential damages. This has led to the emergence of the so-called fault tolerance and self-healing techniques as essential complementary techniques to attain dependable systems.

This chapter presents a twofold self-healing approach to reinforce MANET survivability. The first part of this chapter is devoted to the description of a hybrid replication framework that enables MASID-R-SA to recover from individual and/or multiple agent failures. This is to give it more flexibility, reliability and most importantly high availability which means continuous surveillance of the network.

However, the network is not yet that reliable since data lost due to intrusions is not recovered. For that, it is important to improve the reliability and resilience of the network, so as to enable it to heal itself of faults and to better survive malicious attacks. To this end, a new paradigm for a self-healing MANET is developed in the second part of this chapter. To conclude, the performance of the new IDRS is evaluated.

6.2 Replication for continuous protection

The introduction of a MAS to the distributed and cooperative architecture of the proposed IDRS brought more flexibility and a complete automation of the detection process through the distinguished agent properties like autonomy and pro-activeness. Unfortunately, these features are also the main source of agent faults, thereby multi-agent systems' vulnerability to faults and system failures [79]. Considering this fact, MASID-R-SA is deemed unreliable as it adopts no failure recovery mechanism while being a fault-prone IDRS.

For that, it is necessary to enhance the proposed IDRS' fault tolerance so that to guarantee continuous protection of the network.
Since replication is a key technique to achieve fault tolerance in distributed and dynamic environments (which is the case of MASID-R-SA), we use this technique to avoid malfunctioning resulting from the potential failure of one or more agents within the multi-agent system that makes up MASID-R-SA as described in the following subsections.

6.2.1 Agent Replication

Agent replication [32] is generally defined as the act of creating duplicates of one or more agents in a multi-agent system. Each of these duplicates performs the same task as the original agent. The group of duplicate agents is referred to as a replicate group and the individual agents within the replicate group are referred to as replicas.

There are two basic types of agent replication: heterogeneous and homogeneous. In heterogeneous replication, replicas are functionally equivalent, but they may have been implemented separately i.e., they are not identical but designed to perform the same action. In homogeneous replication, replicas are exact copies of the original agent. In other words, the replicas are not only functionally equivalent but are copies of the same code.

Furthermore, considering the relation between an agent and its replicas, we can distinguish two categories of replication, namely passive and active replication. In passive replication, also called single-copy passive replication or primary-backup replication [14], there exist one active replica (denoted primary) that processes all input messages and periodically updates the other replicas (called backups) in order to maintain coherence and to constitute a recovery point in case of failure. Figure 6.1 (a) illustrates a simple example of a passive replication scenario.

Active replication, also called the state machine approach [91], is characterized by the existence of several replicas that process concurrently all input messages as illustrated in Figure 6.1 (b). Since all of the replicas are active at the same time, this category of

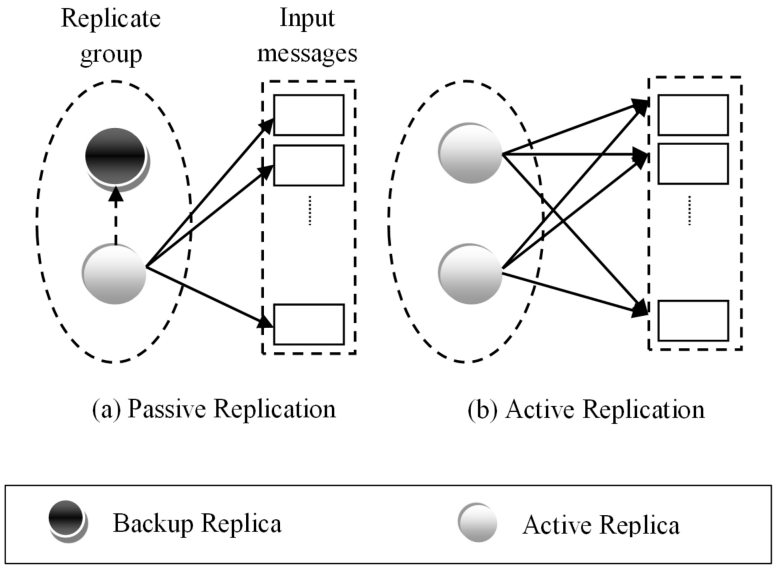


Figure 6.1: Active vs. Passive Replication

replication might lead to the overhead of the CPU but it is still the best choice if fast recovery delays are required. On the contrary, passive replication needs more processing time for recovery but it requires less CPU resources as it activates replicas only in case of failures. It is then obvious that the choice of the most suitable replication technique depends on the application and its environment, such as the failure rate, or the available resources.

6.2.2 Dynamic agent replication

In [4], the authors specified a set of requirements that an IDS for MANET should satisfy. It is, for example, necessary for the IDS to run continuously, minimize resource consumption, and to not degrade the system performances by creating extra overhead.
To satisfy these requirements, the proposed replication framework should:

- Minimize the number of replicas because an increase in the number of replicas entails more resource consumption.
- Not activate several replicas at the same time because having many replicas activated simultaneously implies more processing power.
- Have enough replicas to guarantee that the system will not crash leaving the network without protection.

- Reduce communication cost between active agents and their respective replicas.

From that perspective, a new dynamic replication technique based on passive replication is proposed. The proposed replication framework concerns all of the constituent agents of MASID-R-SA. In summary, the agents concerned with replication are: collector, the detection agent, the collaboration agent and the response agent. This framework works in two separate phases: replication at system initialization and on-demand replication.

6.2.2.1. Replication at System Initialization

An increase in the number of replicas within the IDRS implies an increase in the resources consumed by the IDRS. Moreover, it is, indeed, very difficult to estimate the number of potential failures that an agent would suffer and thus the number of its required replicas. Thus, instead of using several replicas (a replicate group) for each agent, a new type of agents, called the replication manager, is introduced. This latter helps to get rid of the different problems related to replicate groups such as how many replicas to create for each agent, replicas' update and so on, in return of little overhead for the creation of replicas only when failures occur thereby introducing a constant trade-off between consistency and efficiency.

Since the initialization of MASID-R-SA, each of its constituent agents will have a replica ready to take over at any moment. On-demand replication is triggered in case of failure of one or more active agents.

6.2.2.2. On-Demand Replication

On-demand replication means that it is only when failures occur that a new replica is created. Therefore, in case of an agent failure, simply a new replica is created and the already existing replica will take its place as an active agent instead of the failed one. This latter will no longer belong to the IDRS i.e., it will be dropped out from the system. Briefly, MASID-R-SA's state is rolled back to the most recent restoration point and restarted from there whenever needed.

A failed agent might not be able to carry out the replication process by itself. To address this problem, we suggest adding one more agent to MASID-R-SA, which we call the replication manager. This agent has nothing to do with intrusion detection but it is rather responsible for observing and detecting failures within the MAS constituting the IDRS. Additionally, the replication manager dynamically adds or removes replicas, carries out the update of the current replicas and handles failure recovery within each

IDRS. Nevertheless, the replication manager might fail as well. For that, it also needs to be replicated. In that way, we can guarantee that if the replication manager fails, one of its replicas will continue to supervise the system. To handle the replication of the replication managers and to avoid having a single point of failure, each of them will serve as a supervisor for the replication managers on neighbouring nodes and vice versa. Figure 6.2 illustrates the proposed replication framework.

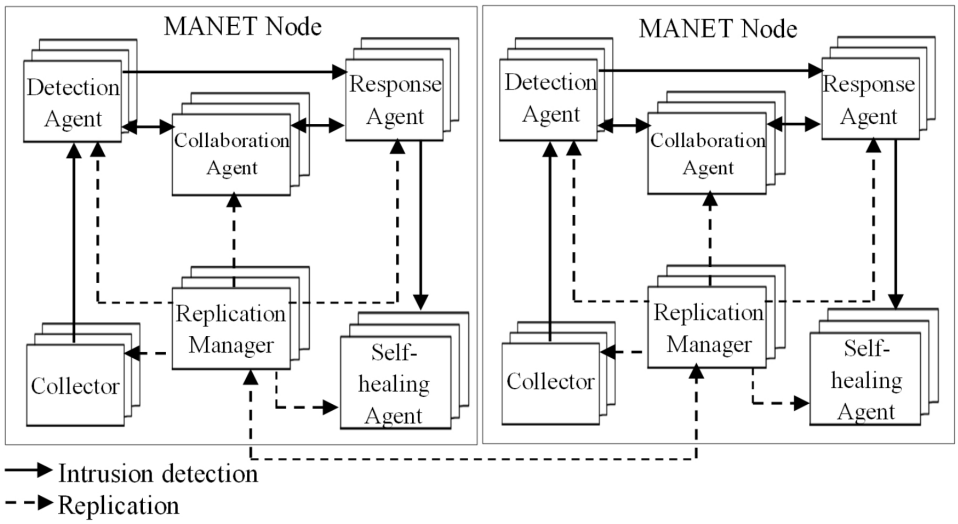


Figure 6.2: Replication Framework for MASID-R-SA

6.2.2.3. Consistency problem

The basic problem with replication techniques is that an update to any given logical object (in our case, active agent) must be propagated to all stored copies (replicas) of that object [68].

In the proposed framework, we distinguish three types of agents. The first type is performing the required intrusion detection tasks. We call them the active agents. The second type of agents represents the replicas (one replica for each agent). The last type of agents is referred to as the replication manager. It is used to supervise and recover the active agents in case of their failure. These agents communicate using a peer-to-peer message-passing mechanism. As explained earlier, the Replication Manager is an agent that continuously observes the active part of the IDRS, builds a state of the IDRS and handles recovery whenever necessary thereby, guaranteeing consistency amongst active agents and their replicas.

Figure 6.3 presents a comparison between the cost, in term of the number of generated messages, of updating replicas using the proposed approach and the one generated using standard replication approaches.
For the sake of simplicity, we assume that, for the standard replication approach, the number R of replicas is the same for all the agents.

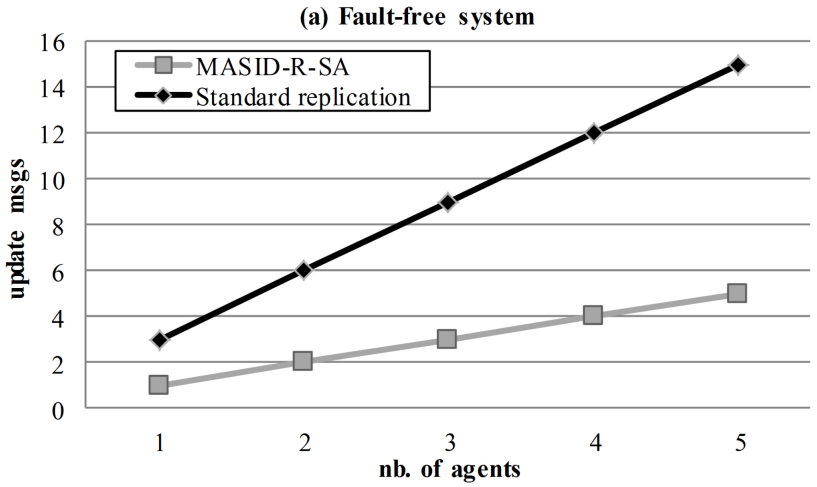


Figure 6.3: Consistency cost - Fault-free System

Compared to standard replication approaches, where the number of replicas' update messages depends not only on the number of updated agents but also on the number of each agent replicas, the number of update messages generated by MASID-R, is significantly reduced especially in case of large-scale systems. Not only the dynamic replication reduces the cost of updating the replicas but it takes the replicas' creation (how many?; where to place?; how to communicate?) burden off the system's designer.

Figure 6.4 shows the impact of failures on the replicas' updating process. Contrary to systems incorporating standard replication approaches, where the system will inevitably crash if the number of failures is greater than the estimated number of replicas, MASID-R-SA continues to work steadily whatever been the number of failures.

The proposed replication framework enabled MASID-R-SA to recover from individual and/or multiple agent failures, thereby guaranteeing permanent protection of the network. However, the network is not yet that reliable since data lost or altered due to intrusions is not recovered. For that, we would like to improve the reliability and consistency of the network, so as to enable it to heal itself of faults and to better survive malicious attacks. In the following section, a recovery-oriented approach for a self-healing MANET based on MASID-R-SA is presented.

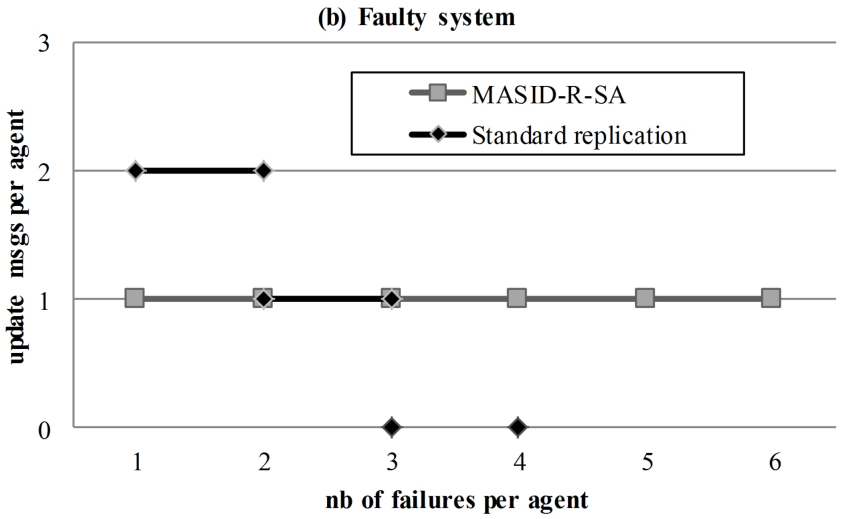


Figure 6.4: Consistency Cost - Faulty System

6.3 IDS-based self-healing

As explained earlier, a self-healing system is one that has the ability to perceive that it is not operating correctly and in that case can make the necessary adjustments to restore itself to normalcy. From that perspective, we divided the proposed self-healing process into two main phases. The first phase is Fault Detection and Damage Spread Stopping while the second will be Self-healing or Fault-repair. The next sub-sections discuss these phases.

6.3.1 Fault detection and damage spread stopping

The proposed healing process is meant to heal faults and damages caused by intruders. Thus, the detection of faults and damages is dependent on the detection of intrusions. For that, we based the healing approach on MASID-R-SA's detection results.

Upon detection of an abnormal behaviour by the detection agent, the response agent will execute the necessary actions to stop the intrusion(s). These may include: dropping the connectivity to the potential intruder either permanently or for a limited period of time, informing other nodes about the detected intrusion and its potential source, and the update of both normal profiles and known attacks databases whenever necessary. To finish this phase, the response agent will trigger the self-healing process by activating the healing agent. The healing agent is a stationary agent with the main function of performing the necessary actions for the healing of the network. It has the ability to communicate with the other agents within the same LIDS.

6.3.2 Self-healing or fault-repair

In the self-healing phase, the healing agent will use information collected by the detection agent about the detected intrusion(s) (e.g., packet drop ratio, delay, victim node(s)' ID(s), intruder(s)' IDs, detection time, and so on) to measure the damage caused by the intruder(s). Then, building on the estimated level of damage, it will create and execute an appropriate list of actions to heal the network.

The healing agent stores information (backup information) about network traffic regularly (during the detection phase). Once an abnormal behaviour is detected by the detection agent, or a notification of a detected intrusion is received by the collaboration agent, this will trigger the healing agent to start the recovery process using both its backup data and data collected during the detection phase as illustrated in the example of Table 6.1.

Detection data	**Healing data**
Node I is the intruder.	Active routes having node I as a member.
x packets were dropped by node I during T (T is the active_detection interval of time).	Source and destination nodes' IDs for each path (it knows the dropped packets were generated by node S and are destined to node D).
Detection time	Copy of the packets sent during T

Table 6.1: Example of IDS and healing data (case of a blackhole or grayhole attack)

The healing agent performs the following tasks:

- The node, on which the healing agent resides, keeps a copy of every sent packet during every active detection interval of time (detection window T).
- Receive messages about anomalous events from the detection agent: if no message is received from the response agent during T. Then the healing agent will purge the recovery base (i.e., it will delete the stored packets' copies from the recovery base at the end of the current detection session). Else, it will start the diagnosis and fault identification by using information contained in the received message (e.g., ID of the intruder, intrusion detection time, drop ratio, and so on).
- Repair the damage caused by the detected intrusive activities. This is a twofold task: the healing agent will first establish a new route, not including the intruder and the suspected nodes (if they exist), to replace the damaged route. Then, it will resend the stored packets to their destination via the newly established route.

Steps for both phases with respect to intrusion detection and response processes are presented in the flowchart of figure 6.5.

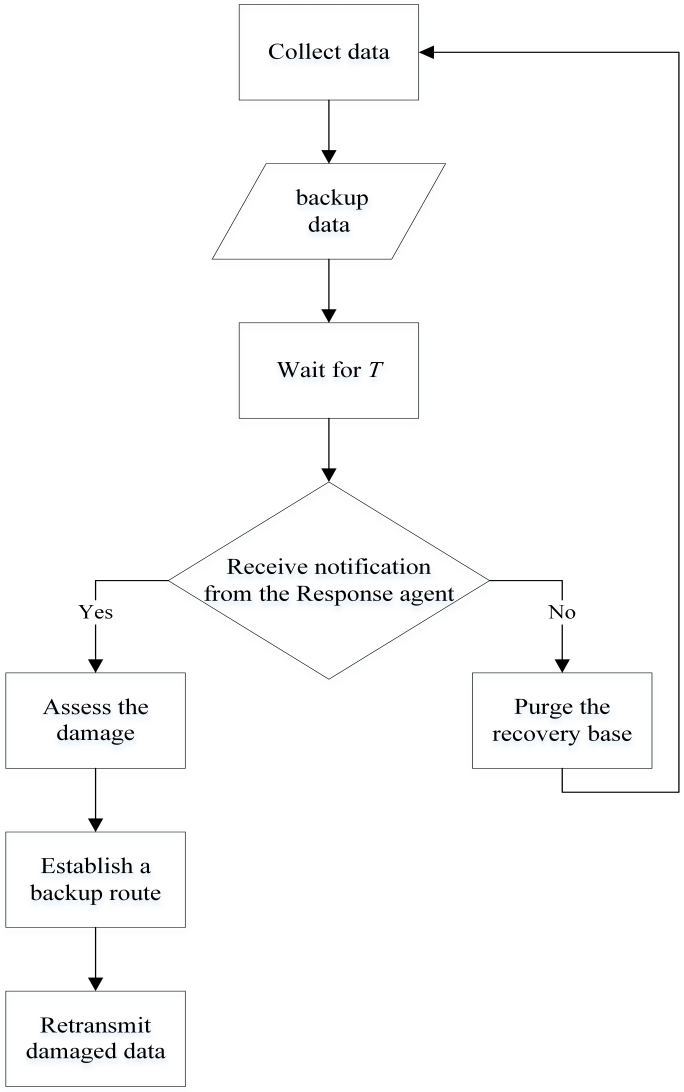


Figure 6.5: Fault Detection and Self-healing Process

With the self-healing agent added to the proposed IDRS, a cumulative of six different but complementary agents was proposed. An example of agent interactions within a LIDS is presented in the AUML (Agent Unified Modelling Language) sequence diagram of Figure 6.6.

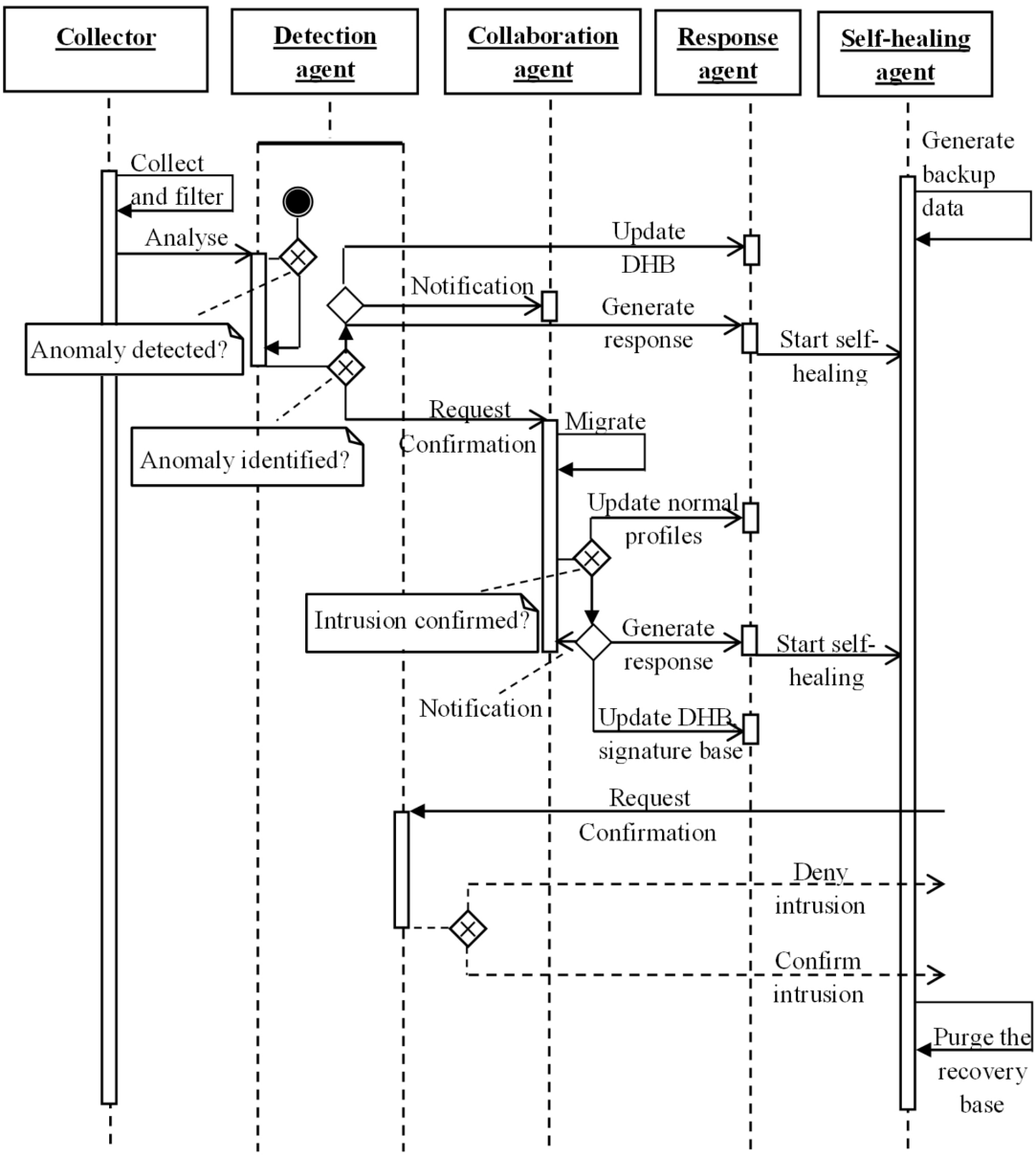


Figure 6.6: Agent Interactions within a LIDS

6.4 Experiments and Results

n order to evaluate the proposed approach, we continue our series of simulation experiments using NS-2. This time, the proposed approach is validated against the two packet dropping attacks, namely: blackhole and grayhole.
The same simulation settings and parameters used in the previous chapter (Table 5.1) are used in these experiments. A failure of MASID and MASID-R-SA is planned at 30 s, in some simulation scenarios, to demonstrate the feasibility of replicating agents within MASID.

Figure 6.7 presents the evolution over time of the packet delivery ratio. In the presence of intrusions, PDR has notoriously increased through the use of MASID-R-SA but a more considerable increase was achieved after the integration of the healing agent since even packets that were timed-out or dropped due to congestion could be restored. Upon failure occurrence, however, MASID will no longer be able to perform correctly leading to the success of intruders in dropping considerable amounts of packets whereas, MASID-R-SA could heal itself, using its replication system thereby, guaranteeing a continuous protection of the network.

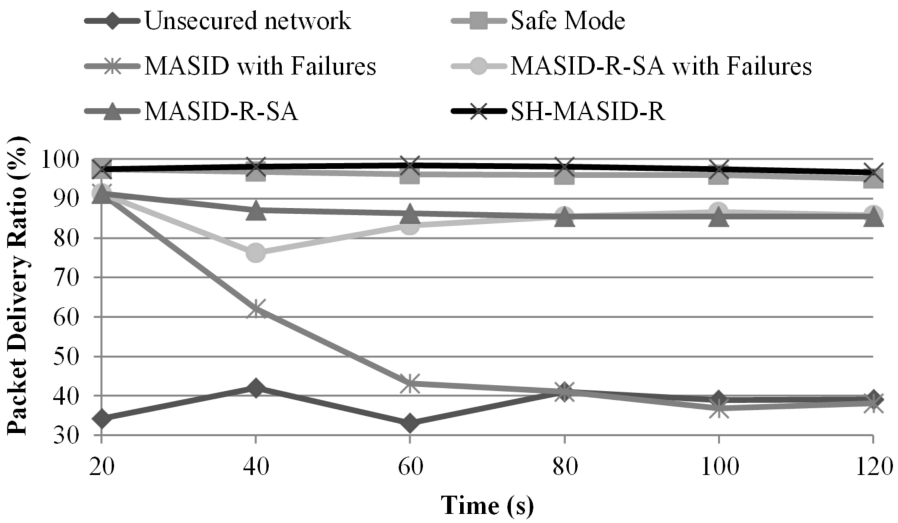


Figure 6.7: Packet delivery ratio vs. time

Unfortunately, to achieve these rates, it was necessary to create a kind of trade-off between guaranteeing the delivery of packets and both the overall communication time and

the generated control overhead. More accurately, the healing approach tends to increase the ratio of correctly delivered packets at the cost of increased latency in the interrupted communication's delay resulted from the resubmission of the damaged packets as shown in Figure 6.8.

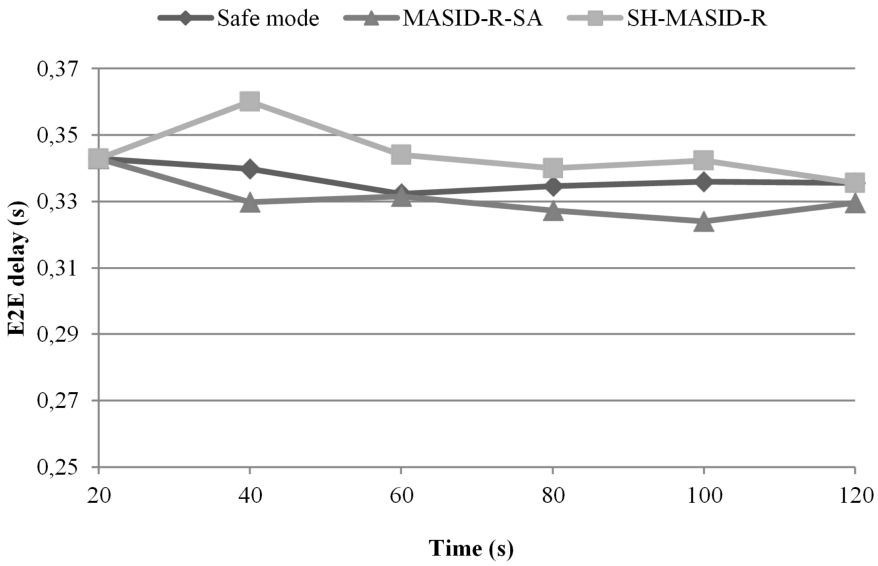


Figure 6.8: End-to-End Delay vs. time

In addition to the potential increase in the communications' delays, some traffic overhead may result due to the new route search as shown in Figure 6.9.

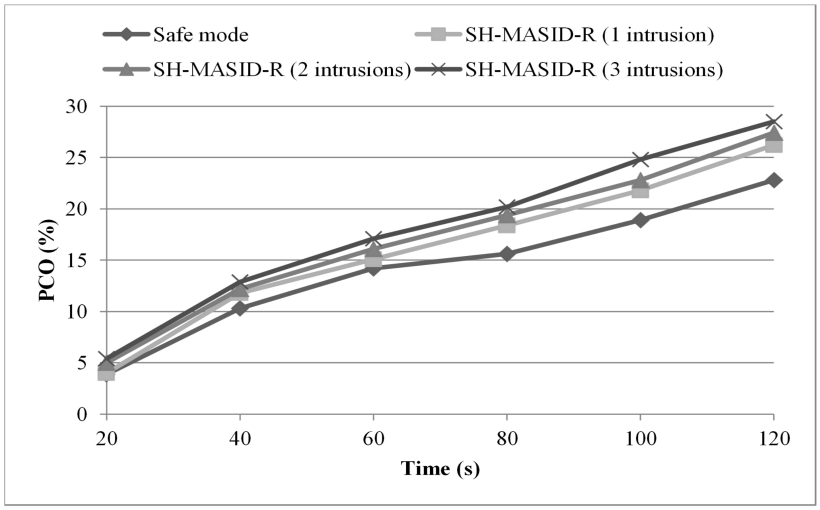


Figure 6.9: Packet Control Overhead vs. time

Fortunately, this overhead is proportional to the number of intrusions and their distribution over time, i.e., it increases with the increase in the number of intrusions and decreases if the risk disappears.

6.5 Conclusion

In this chapter, a twofold self-healing approach for MANET survivability was presented. First, a fault-tolerant IDS is designed by replication of individual agents within MASID-R-SA. This IDS is flexible as it possesses the ability to decide about when and which parts of the IDRS should be replicated. Also, replication helps in overcoming individual agent failures which leads to a significant increase the IDRS' availability.

Then, a recovery-oriented approach is proposed to enable the supervised network to heal itself of those potentially caused faults and damages. It is based on the ability of MASID-R-SA to assess the damage caused by the detected intrusions. Building on this assessment, MASID-R-SA, via its healing agent, initiates and executes the necessary actions to heal the network. The main objective of this approach is to enable the supervised network to heal itself of faults and damages caused by intrusions and to better survive malicious attacks (mainly packet dropping attacks) and malfunctioning, which would considerably improve the resilience and reliability of the network, thus improving its survivability.

The resulting system is fully autonomous: it accurately detects intrusions launched against it, appropriately responds to them, and perfectly heals the caused damages.

The following chapter summarises our contributions, presents some concluding remarks, and initiates for possible future work directions.

Chapter 7

General conclusion and perspectives

7.1 Conclusion

In this thesis, we tackled the problem of intrusion detection and response in MANET. First, an intrusion detection system denoted MASID was proposed. MASID is an agent-based distributed and cooperative IDS where every network node is equipped with a local IDS (LIDS). Each LIDS consists of different but complementary agents: collector, the detection agent, collaborator, replication agents, the response agent, and the self-healing agent. Neighbouring LIDSs can communicate using mobile agents. Each LIDS detects intrusions from local traces and initiates local and global response. If an anomaly is detected or if there are signs of intrusion and there is not enough evidence, neighbouring LIDSs will cooperatively participate in the detection process by providing some additional information. By using agents we achieved not only a complete automation of the detection process but also we took advantage of the interesting characteristics presented by the agent technology like their autonomy, reactivity, fault-tolerance, and mobility. The distributed and cooperative nature of MASID permits broader coverage and enhances detection rates through redundancy. An IDS detects intrusions and may permit to identify and localize their source. Ideally, an IDS should have an embedded response module or can trigger an independent response system to mitigate the detected intrusions. From that perspective, we further proposed a new response mechanism to enable MASID to adaptively respond to the detected intrusions. This mechanism is based on its ability to assess the severity of any detected intrusion (both known and unknown) and the generation of responses accordingly. More specifically, the response agent distinguishes between the detected intrusions according to their estimated severity levels and their distribution over time. Once an intrusion is detected, the response agent calculates its SD and compares it to the SI. The generated response is then adjusted so as to respond to intrusions that have a SD

smaller than the SI with simple responses while severe responses are reserved to severe intrusions (those with a SD greater than the SI). Simple responses refer to a temporary cut of the connection to the potential intruder. The punishment period is exponential to the caused damage which entails that response severity will increase with the increase of the intrusions' severity. Severe responses are meant to completely and permanently cut the connection to the potential intruder. In case of a repeated occurrence of a simple intrusion, the cumulative severity degree is used to reflect the aggregated damage caused by those intrusions.

Finally, considering that MASs are fault-prone, a fault-tolerant IDS is designed by replication of individual agents within MASID-R-SA to ensure continuous supervision of the network. However, since not all intrusions are predictable, some damage might be experienced before these intrusions are detected and completely removed. For that, even if the implications of intrusions could be minimized by MASID-R-SA, still the need for the recovery of altered or deleted data is a vital step to ensure the correct functioning of the network. For that, a recovery-oriented approach for a self-healing MANET was also presented. It is based on the ability of MASID-R-SA to assess the damage caused by the detected intrusions and aimed at enabling the supervised network to heal itself of those faults and damages. In this way, we could achieve better detection rates and the proposed IDRS is now capable of providing appropriate and systematic responses to the detected intrusions. In addition, the severity-aware approach reduces the effects of false positive alarms. It also reduces the possibility of network partitioning and the IDRS is made resilient enough to maintain its consistency even after network remerging.

7.2 Perspectives

- Providing security for both agents and their communications in addition to a trust mechanism that can be adopted to prevent malicious nodes from initiating blackmail attacks through the generation of fake alarms.

- This thesis was limited to the detection of blackhole, grayhole, and the selfish behaviour attacks. Future research will provide a more comprehensive attacks' signature database (for misuse detection); as well as behaviour patterns to enable anomaly detection. We plan also to extend the parameters used for identifying intrusive acts so as to reduce false detection rates. By the way, we tend to expand our experiments to include more complex network scenarios and traffic patterns.

- Software agents are characterized by their ability to adapt their actions through interaction with their environment thereby, improving their performance over time.

This can help in considering the feedback of intrusion detection and response to improve future judgements about detections and the generated responses.

- It is worth mentioning that node mobility makes the problem of detecting intruders harder. Future works will focus on the effects of mobility on both attacks and the intrusion detection and response processes.
- The healing ability of the proposed IDRS can be improved so that to heal the network of all kinds of damages and faults that can be caused by the potential intrusions.

Acronyms

ACO	Ant Colony Optimization
ADA	Anomaly Detection Agent
ADCLI	Algorithm for Detection in a CLIque
ADCLU	Algorithm for Detection in a CLUster
AODV	Ad hoc On-demand Distance Vector
ARAN	Authenticated Routing for Ad-hoc Networks
ASR	Attack success rate
AUML	Agent Unified Modelling Language
CAIDS	Context Adaptive Intrusion Detection System
CBR	Constant Bit Rate
CD	Cumulative Damage
CE	Communicating Entities
CH	Cluster Head
CM	Cluster Member
CP-KNN	Conformal Predictor K-Nearest Neighbour
CSA	Communication Service Agent
CSD	Cumulative Severity Degree
DCM	Data Collection Module
DDIDS	Dynamic Distributed Intrusion Detection System
DDoS	Distributed Denial of Service
DHB	Detection History Base
DM	Detection Module
DOD	Distance-based Outlier Detection

DoS	Denial of Service
DSR	Dynamic Source Routing
DW	Detection Window
E2E	End-to-End Delay
FN	False Negative
FNR	False Negative Rate
FP	False Positive
FPR	False Positive Rate
FT	Fault tolerance
GDM	Global Detection Module
GIDS	Gateway Intrusion Detection System
GRM	Global Response Module
HIDS	Host-based Intrusion Detection System
ID	IDentifier
IDAR	Intrusion Detection and Adaptive Response
IDRS	Intrusion Detection and Response System
IDS	Intrusion Detection System
IP	Internet Protocol
IPS	Intrusion Prevention System
IRS	Intrusion Response System
ISO	International Organization for Standardization
LDM	Local Detection Module
LID	Local Intrusion Detection
LIDS	Local IDS
LRM	Local Response Module
LTM	Long Term Memory
MAC	Medium Access Control
MANET	Mobile Ad-hoc NETwork
MAS	Multi-Agent System
MASID	Multi-Agent System for Intrusion Detection

NIDS	Network-based Intrusion Detection System
NS-2	Network Simulator 2
NTA	Network Tomography Agent
OLSR	Optimized Link State Routing
OTcl	Object-oriented Tcl
PCO	Packet Control Overhead
PDR	Packet Delivery Ratio
PM	Pre-process Module
RERR	Route ERRor
RL	reinforcement learning
RM	Registration Module
RP	Routing Protocol
RR	Response Rate
RREP	Route REPly
RREQ	Route REQuest
SA	Service Agreement
SAODV	Secure AODV
SD	Severity Degree
SDA	State Detection Agent
SEAD	Secure Efficient Adhoc Distance vector
SI	Severity Index
SL	Severity Level
SRP	Secure Routing Protocol
STM	Short Term Memory
TDR	True Detection Rate
TN	True Negative
TP	True Positive
ZRP	Zone Routing Protocol

List of Publications

Articles in Journals and Book chapter:

[Mechtri2017] L. Mechtri, F. T. Djemili, S. Ghanemi. "An Optimized Intrusion Response System for MANET: An Attack-Severity Aware Approach." Peer to Peer Networking and Applications, Vol. 11, No. 3, pp. 602-618, 2017.

[Mechtri2017] L. Mechtri, F. T. Djemili, S. Ghanemi, and D. Magoni. "A Twofold Self-Healing Approach for MANET Survivability reinforcement", International Journal of Intelligent Engineering Informatics (IJIEI), Vol. 5, No. 4, 2017.

[Mechtri2016] L. Mechtri, F. T. Djemili, S. Ghanemi. "Agents for Intrusion Detection in MANET: Survey and Analysis". In W. Awad, E. El-Alfy, and Y. Al-Bastaki (Eds.) Improving Information Security Practices through Computational Intelligence (Pp. 126-147). Hershey, PA. IGI Global, 2016.

[Mechtri2013] L. Mechtri, F. T. Djemili, S. Ghanemi. "On the Design of a New Intrusion Detection System for Securing MANET: An Agent-Based Approach." International Journal of Advanced Computer Science, Vol. 3, No. 6. 2013.

Conferences

[Mechtri2015] L. Mechtri, F. T. Djemili, S. Ghanemi, D. Magoni. "An IDS-based Self-healing Approach for MANET Survival", International Conference on Intelligent Information Processing, Security and Advanced Communication (IPAC'2015), Nov. 23-25, 2015, Batna, Algeria.

[Mechtri2014] L. Mechtri, F. T. Djemili, S. Ghanemi. "Towards High Reliability of a Multi-Agent System Designed for Intrusion Detection in MANET". ICEECA'2014, Nov. 18-20, 2014.

[**Mechtri2013a]** L. Mechtri, F. T. Djemili, S. Ghanemi. "Détection d'Intrusions Fiable dans MANET", Troisièmes Journées Doctorales en Informatique (JDI' 2013), Université 08 Mai 1945 de Guelma, December 4-5, 2013.

[**Mechtri2013b]** L. Mechtri, F. T. Djemili, S. Ghanemi. "A New Agent-based Intrusion Detection System for MANET", 2nd National Conference on Theoretical and Applicative

Aspects of Computer Science (CTAACS'13), Université 20 août 1955 – Skikda, November 25-26, 2013.

[Mechtri2012] L. Mechtri, F. T. Djemili, S. Ghanemi. "MASID: Multi-Agent System for Intrusion Detection in MANET". Ninth International Conference on Information Technology- New Generations (ITNG 2012), Las Vegas, Nevada, USA, 2012.

[Mechtri2011] L. Mechtri, F. T. Djemili, S. Ghanemi. "Nouvel Algorithme de Détection d'Intrusions dans les Réseaux Informatiques". 1ères Journées Doctorales du Laboratoire d'Informatique d'Oran (JDLIO'11), les 31 Mai et 01 Juin 2011.

Bibliography

[1] Information technology – security techniques – information security management systems – overview and vocabulary. ISO/IEC 27000: 2018 (e), 2018.

[2] C. Adams, R. Housley, and S. Turner. Certification authority. In T. van, C. A. Henk, and S. Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 193–195. Springer US, Boston, MA, 2011.

[3] F.T. AL-Dhief, N. Sabri, S. Fouad, N.M. Abdul Latiff, and M. Albader. A review of forest fire surveillance technologies: Mobile ad-hoc network routing protocols perspective. *Journal of King Saud University - Computer and Information Sciences*, 2017.

[4] P. Albers, O. Camp, J. Percher, B. Jouga, and R. Puttini. Security in ad hoc networks: a general intrusion detection architecture enhancing trust based approaches. In *In Proceedings of the First International Workshop on Wireless Information Systems (WIS-2002)*, pages 1–12, Ciudad Real, Spain, 2002.

[5] S. Aluvala, K.R. Sekhar, and D. Vodnala. A novel technique for node authentication in mobile ad hoc networks. *Perspectives in Science*, 8:680–682, 2016.

[6] J.P. Anderson. Computer security technology planning study, vol. 2, 1972.

[7] M. Barbeau, J. Hall, and E. Kranakis. Detecting impersonation attacks in future wireless and mobile networks. In M. Burmester and A. Yasinsac, editors, *Secure Mobile Ad-hoc Networks and Sensors*, pages 80–95, Berlin, Heidelberg, 2006. Springer Berlin Heidelberg.

[8] D. Bein. Self-configuring, self-organizing, and self-healing schemes in mobile ad hoc networks. In Chandra Misra S. Misra S., Woungang I., editor, *Guide to Wireless Ad Hoc Networks*, pages 27–41. Springer-London, 2009.

[9] M. Blanton. Message authentication codes. In Ling Liu and M. Tamer Özsu, editors, *Encyclopedia of Database Systems*, pages 1–1. Springer New York, New York, NY, 2016.

[10] K. Breitman, M.A. Casanova, and W. Truszkowski. *Software Agents. In Semantic Web: Concepts, Technologies and Applications. NASA Monographs in Systems and Software Engineering, pages 219-228.* Springer London, 2007.

[11] W. Brenner, R. Zarnekow, and H. Wittig. *Intelligent Software Agents - Foundations and Applications.* Springer berlin Heidelberg, 1998.

[12] B. Brewington, R. Gray, K. Moizumi, D. Kotz, G. Cybenko, and D. Rus. Mobile agents for distributed information retrieval. In M. Klusch, editor, *Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet*, pages 355–395. Springer Berlin Heidelberg, 1999.

[13] P. Brucker. *Scheduling Algorithms (Fifth Edition).* Springer Berlin Heidelberg New York, 2007.

[14] N. Budhiraja, K. Marzullo, F.B. Schneider, and S. Toueg. The primary-backup approach. In S. Mullender, editor, *Distributed Systems (2nd Ed.)*, pages 199–216. ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1993.

[15] A. Byrski and M. Carvalho. Agent-based immunological intrusion detection system for mobile ad-hoc networks. In *Proceedings of the International Conference on Computational Science*, pages 584–593, Kraków, Poland, 2008.

[16] H. Cavusoglu, B. Mishra, and S. Raghunathan. The value of intrusion detection systems in information technology security architecture. *Information Systems Research*, 16:28–46, 2005.

[17] K. Chang and K.G. Shin. Application-layer intrusion detection in manets. In *43rd Hawaii International Conference on System Sciences*, pages 1–10, Honolulu, HI, 2010.

[18] B. Cheng and R. Tseng. A context adaptive intrusion detection system for manet. *Computer Communications*, 34(3):310–318, March 2011.

[19] T. Clausen and P. Jacquet. Optimized link state routing (OLSR) protocol, RFC 3626, IETF, 2003.

[20] F. Cohen. A cryptographic checksum for integrity protection. *Computers & Security*, 6(6):505–510, 1987.

[21] H. Dahshan and J. Irvine. On Demand Self-Organized Public Key Management for Mobile Ad Hoc Networks. In *IEEE 69th Vehicular Technology Conference, VTC Spring 2009*, pages 1–5, Barcelona, Spain, April 2009.

[22] E. Darra, C. Ntantogian, C. Xenakis, and S. Katsikas. A mobility and energy-aware hierarchical intrusion detection system for mobile ad hoc networks. In *TrustBus 2011, LNCS 6863*, pages 138–149. Springer, 2011.

[23] H. Delfs and H. Knebl. *Introduction to Cryptography: Principles and Applications*. Springer Verlag London, 2015.

[24] M. Dell'Amico, P. Michiardi, L. Toka, and P. Cataldi. Adaptive redundancy management for durable P2P backup. *Computer Networks*, 83:136 – 148, 2015.

[25] Y. Desmedt. Man-in-the-middle attack. In T. Van, C.A. Henk, and S. Jajodia, editors, *Encyclopedia of Cryptography and Security*, pages 759–759. Springer US, Boston, MA, 2011.

[26] V.A. Devi and R.S. Bhuvaneswaran. Agent based cross layer intrusion detection system for MANET. In *Advances in Network Security and Applications*, pages 427–440, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[27] K. Drira, H. Seba, and H. Kheddouci. ECGK: An efficient clustering scheme for group key management in MANETs. *Computer Communications*, 33(9):1094–1107, 2010.

[28] M. Elsadig and A. Abdullah. Biological inspired intrusion prevention and self-healing system for network security based on danger theory. *International Journal of Video & Image Processing and Network Security*, 9:16–28, 2009.

[29] O. Ermiş, Ş. Bahtiyar, E. Anarım, and M.U. Çağlayan. A secure and efficient group key agreement approach for mobile ad hoc networks. *Ad Hoc Networks*, 67:24–39, 2017.

[30] A.F. Farhan, Z.Md. Dahalin, and S. Jusoh. Distributed and cooperative hierarchical intrusion detection on MANETs. *International Journal of Computer Applications*, 12:32–40, 2010.

[31] A.F. Farhan, D. Zulkhairi, and M.T. Hatim. Mobile agent intrusion detection system for mobile ad hoc networks: A non-overlapping zone approach. In *Proceedings of the 4th IEEE/IFIP International Conference on Internet*, pages 1–5, Tashkent, 2008. IEEE.

[32] A. Fedoruk and R. Deters. Improving fault-tolerance by replicating agents. In *Proceedings of the First International Joint Conference on Autonomous Agents and Multiagent Systems: Part 2*, AAMAS '02, pages 737–744, New York, NY, USA, 2002. ACM.

[33] B.A. Fessi, S. Benabdallah, N. Boudriga, and M. Hamdi. A multi-attribute decision model for intrusion response system. *Information Sciences*, 270:237–254, 2014.

[34] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez. Anomaly-based network intrusion detection: Techniques, systems and challenges. *Computers & Security*, 28:18–28, 2009.

[35] M. Gharib, Z. Moradlou, M. A. Doostari, and A. Movaghar. Fully distributed ECC-based key management for mobile ad hoc networks. *Computer Networks*, 113:269–283, 2017.

[36] D. Ghosh, R. Sharman, H.R. Rao, and S. Upadhyaya. Self-healing systems - survey and synthesis. *Decision Support Systems*, 42(4):2164–2185, 2007.

[37] K. Gomathi, B. Parvathavarthini, and C. Saravanakumar. An Efficient Secure Group Communication in MANET Using Fuzzy Trust Based Clustering and Hierarchical Distributed Group Key Management. *Wireless Personal Communications*, 94(4):2149–2162, Jun 2017.

[38] Z.J. Haas, M.R. Pearlman, and P. Samar. The interzone routing protocol (ierp) for ad hoc networks. ietf internet draft, IETF, 2002.

[39] Z.J. Haas, M.R. Pearlman, and P. Samar. The intrazone routing protocol (IARP) for ad hoc networks. ietf internet draft, IETF, 2002.

[40] Z.J. Haas, M.R. Pearlman, and P. Samar. The zone routing protocol (zrp) for ad hoc networks. ietf internet draft, IETF, 2002.

[41] S. Hansman and R. Hunt. A taxonomy of network and computer attacks. *Comput. Secur.*, 24:31–43, 2005.

[42] M. Hasan and S. Goraya. Fault tolerance in cloud computing environment: A systematic survey. *Computers in Industry*, 99:156–172, 2018.

[43] J. He. Complexity in Adaptive System. In Sammut C. and Webb G.I., editors, *Encyclopedia of Machine Learning and Data Mining*, pages 243–247. Springer, US, 2017.

[44] C. Hedrick. Routing information protocol, RFC 1058, IETF, 1988.

[45] L. Hogie, P. Bouvry, and F. Guinand. An overview of MANETs simulation. *Electronic Notes in Theoretical Computer Science*, 150(1):81–101, 2006.

[46] C. Hong-Song, Z. Jianyu, and H. Lee. A novel NP-based security scheme for AODV routing protocol. *Journal of Discrete Mathematical Sciences and Cryptography*, 11(2):131–145, 2008.

[47] C. Hong-Song, J. Zhenzhou, H. Mingzeng, F. Zhongchuan, and J. Ruixiang. Design and performance evaluation of a multi-agent-based dynamic lifetime security scheme for AODV routing protocol. *Journal of Network and Computer Applications*, 30(1):145 – 166, 2007.

[48] Y-C. Hu, D.B. Johnson, and A. Perrig. SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks. *Ad Hoc Networks*, 1(1):175–192, 2003.

[49] L. Hung-Jen, R. L. Chun-Hung, L. Ying-Chih, and T. Kuang-Yuan. Intrusion detection system: A comprehensive review. *Journal of Network and Computer Applications*, 36:16–24, 2013.

[50] G. Indirani and K. Selvakumar. Swarm based intrusion detection and defense technique for malicious attacks in mobile ad hoc networks. *International Journal of Computer Applications*, 50(19):1–7, 2012.

[51] S. P. John and P. Samuel. Self-organized key management with trusted certificate exchange in MANET. *Ain Shams Engineering Journal*, 6(1):161–170, 2015.

[52] D.B. Johnson, D.A. Maltz, and Y-C. Hu. The dynamic source routing protocol (dsr) for mobile ad hoc networks for ipv4. rfc 4728, IETF, 2007.

[53] J. Katz. *Digital signatures*. Springer, boston, MA, 2010.

[54] P. Kaur, D. Rattan, and A.K. Bhardwaj. An analysis of mechanisms for making ids fault tolerant. *International Journal of Computer Applications*, 1(24):22–25, 2010.

[55] V.Y. Kishorbhai and N.N. Vasantbhai. AON: A survey on emergency communication systems during a catastrophic disaster. *Procedia Computer Science*, 115:838–845, 2017. 7th International Conference on Advances in Computing & Communications, ICACC-2017, 22-24 August 2017, Cochin, India.

[56] J. Kong, X. Hong, Y. Yi, J-S. Park, J. Liu, and M. Gerla. A secure ad-hoc routing approach using localized self-healing communities. In *Proceedings of the 6th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '05, pages 254–265, New York, NY, USA, 2005. ACM.

[57] S. Kremer and O. Markowitch. Fair multi-party non-repudiation protocols. *International Journal of Information Security*, 1(4):223–235, Jul 2003.

[58] P. Kumar and K. Reddy. An agent based intrusion detection system for wireless network with Artificial Immune System (AIS) and Negative Clone Selection. In *International Conference on Electronic Systems, Signal Processing and Computing Technologies*, pages 429–433, 2014.

[59] D. B. Lange and M. Oshima. Introduction to mobile agents. *Personal Technologies*, 2(2):49–56, Jun 1998.

[60] C-H. Lee and J. Suzuki. SWAT: a decentralized self-healing mechanism for wormhole attacks in wireless sensor networks. In Y. Xiao, H. Chen, and F. Li, editors, *Handbook on Sensor Networks*, pages 01–21. World Scientific, 2008.

[61] X. Li, J. Xu, H-N. Dai, Q. Zhao, C.F. Cheang, and Q. Wang. On modeling eavesdropping attacks in wireless networks. *Journal of Computational Science*, 11:196–204, 2015.

[62] Y. Li and Z. Qian. Mobile agents-based intrusion detection system for mobile ad hoc networks. In *Proceedings of the International Conference on Innovative Computing and Communication and 2010 Asia-Pacific Conference on Information Technology and Ocean Engineering*, pages 145–148, Macao, China, 2010. IEEE.

[63] N. Marchang and R. Datta. Collaborative techniques for intrusion detection in mobile ad-hoc networks. *Ad Hoc Networks*, 6(4):508 – 523, 2008.

[64] Riverbed Modeler. https://www.riverbed.com/gb/products/steelcentral/steelcentral-riverbed-modeler.html.

[65] A. Mukherjee, A. Gupta, and D. P. Agrawal. Distributed key management for dynamic groups in MANETs. *Pervasive and Mobile Computing*, 4(4):562 – 578, 2008.

[66] A. Nadeem and M.P. Howarth. An intrusion detection and adaptive response mechanism for MANETs. *Ad Hoc Networks*, 13:368–380, 2014.

[67] NS-2. The network simulator. http://isi.edu/nsnam/ns/.

[68] M.M. Oo, T.T. Soe, and A Thida. Fault tolerance by replication of distributed database in p2p system using agent approach. *International Journal of Computers*, 4(1):09 – 18, 2010.

[69] C. Panos, C. Ntantogian, S. Malliaros, and C. Xenakis. Analyzing, quantifying, and detecting the blackhole attack in infrastructure-less networks. *Computer Networks*, 113:94 – 110, 2017.

[70] P. Papadimitratos, Z.J. Haas, and P. Samar. The Secure Routing Protocol (SRP) for Ad Hoc Networks. IETF Internet Draft, IETF, 2002.

[71] S. S. Park, J. H. Lee, and T. M. Chung. Cluster-based trust model against attacks in ad-hoc networks. In *2008 Third International Conference on Convergence and Hybrid Information Technology*, pages 526–532, Busan, South Korea, Nov 2008.

[72] A.D. Patel and K. Chawda. Dual security against grayhole attack in manets. In L.C. Jain, S. Patnaik, and N. Ichalkaranje, editors, *Intelligent Computing, Communication and Devices*, pages 33–37, New Delhi, 2015. Springer India.

[73] B.K. Pattanayak and M. Rath. A mobile agent based intrusion detection system architecture for mobile ad hoc networks. *Journal of Computer Science*, 10(6):970–975, 2014.

[74] C. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings of the Conference on Communications Architectures, Protocols and Applications*, SIGCOMM '94, pages 234–244, New York, NY, USA, 1994. ACM.

[75] C. Perkins and E. Royer. Ad-hoc on-demand distance vector routing. In *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, WMCSA '99, pages 90–100, Washington, DC, USA, 1999. IEEE Computer Society.

[76] C. Perkins, E. Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. RFC 3561, IETF, 2003.

[77] Y. Ping, Z. Futai, J. Xinghao, and L. Jianhua. Multi-agent cooperative intrusion response in mobile adhoc networks. *Journal of Systems Engineering and Electronics*, 18(4):785–794, 2007.

[78] A. A. Pirzada and C. Mcdonald. Trust establishment in pure ad-hoc networks. *Wireless Personal Communications*, 37(1):139–163, Apr 2006.

[79] K. Potiron, A. Seghrouchni, and P. Taillibert. *From Fault Classification to Fault Tolerance for Multi-Agent Systems*. Springer berlin Heidelberg, 2013.

[80] C. Prasenjit, G. Rajasekhar, P.R. Babu, D.M. Babu, and V. Satyanarayana. A New Multi-language Encryption Technique for MANET. In *Proceedings of Information and Communication Technologies*, pages 22–28, Berlin, Heidelberg, 2010.

[81] L. Pullum. *Software Fault Tolerance Techniques and Implementation*. Artech House, Inc., Norwood, MA, USA, 2001.

[82] K. Rama Abirami and M.G. Sumithra. Preventing the impact of selfish behavior under manet using neighbor credit value based aodv routing algorithm. *Sādhanā*, 43(4):60, Apr 2018.

[83] C. Ramachandran, S. Misra, and M.S. Obaidat. A novel two-pronged strategy for an agent-based intrusion detection scheme in ad-hoc networks. *Comput. Commun*, 31(16):3855 – 3869, 2008.

[84] P. Ramya and T. SairamVamsi. Impact analysis of blackhole, flooding, and grayhole attacks and security enhancements in mobile ad hoc networks using sha3 algorithm. In J. Anguera, S.C. Satapathy, V. Bhateja, and K.V.N. Sunitha, editors, *Microelectronics, Electromagnetics and Telecommunications*, pages 639–647, Singapore, 2018. Springer Singapore.

[85] S.A. Razak, S.M. Furnell, and P.J. Brooke. Attacks against mobile ad hoc networks routing protocols. In *5th annual postgraduate symposium on the convergence of telecommunications, networking and broadcasting*, PgNeT, pages 147–152, 2004.

[86] S.A. Razak, S.M. Furnell, N.L. Clarke, and P.J. Brooke. Friend-assisted intrusion detection and response mechanisms for mobile ad hoc networks. *Ad Hoc Networks*, 6(7):1151–1167, 2008.

[87] D.B. Roy and R. Chaki. MABHIDS: A New Mobile Agent Based Black Hole Intrusion Detection System. In N. Chaki and A. Cortesi, editors, *Computer Information Systems – Analysis and Technologies*, pages 85–94, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[88] K. Sanzgiri, D. LaFlamme, B. Dahill, B.N. Levine, C. Shields, and E. Royer. Authenticated routing for ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 23(3):598–610, March 2005.

[89] R. Sasikumar and D. Manjula. Dynamic distributed intrusion detection system based on mobile agents with fault tolerance. *Journal of Computer Science, Science Publications*, 8(7):1092–1098, 2012.

[90] M. Saswati, C. Matangini, C. Samiran, and K. Pragma. EAER-AODV: Enhanced Trust Model Based on Average Encounter Rate for Secure Routing in MANET. In C. Rituparna, C. Agostino, S. Khalid, and C. Nabendu, editors, *Advanced Computing and Systems for Security (6)*, pages 135–151. Springer Singapore, Singapore, 2018.

[91] Fred B. Schneider. Implementing fault-tolerant services using the state machine approach: A tutorial. *ACM Computing Surveys*, 22(4):299–319, December 1990.

[92] J. Sen. A robust and fault-tolerant distributed intrusion detection system. In *Proceeding of 1st International Conference on Parallel, Distributed and Grid Computing (PDGC'10)*, pages 123–128, India, 2010.

[93] J. Sengathir and R. Manoharan. Co-operation enforcing reputation-based detection techniques and frameworks for handling selfish node behaviour in manets: A review. *Wireless Personal Communications*, 97(3):3427–3447, Dec 2017.

[94] M.D. Serrat-Olmos, E. Hernández-Orallo, J.C. Cano, C.T. Calafate, and P. Manzoni. A collaborative bayesian watchdog for detecting black holes in MANETs. In *Intelligent Distributed Computing VI*, page 221–230, 2013.

[95] A. Servin and D. Kudenko. Multi-agent reinforcement learning for intrusion detection. In K. Tuyls et al., editor, *Adaptive Agents and Multi Agent Systems III: Adaptation and Multi Agent Learning*, pages 211–223. Springer-Verlag, Berlin Heidelberg, 2008.

[96] P. Sethuraman and N. Kannan. Refined trust energy-ad hoc on demand distance vector (ReTE-AODV) routing algorithm for secured routing in MANET. *Wireless Networks*, 23(7):2227–2237, Oct 2017.

[97] A. Shameli-Sendi, M. Cheriet, and A. Hamou-Lhadj. Taxonomy of intrusion risk assessment and response system. *Computers & Security*, 45:1–16, 2014.

[98] P. Sharma, N. Sharma, and R. Singh. A secure intrusion detection system against ddos attack in wireless mobile ad-hoc network. *International Journal of Computer Applications*, 41(121):16–21, 2012.

[99] S. Shaw, K. Orea, P. Venkateswaran, and R. Nandi. Simulation and performance analysis of OLSR under identity spoofing attack for mobile Ad-Hoc networks. In V. Das, J. Stephen, and Y. Chaba, editors, *Computer Networks and Information Technologies*, pages 308–310, Berlin, Heidelberg, 2011. Springer Berlin Heidelberg.

[100] N. Sridevi and V. Nagarajan. A curve based cryptography for wireless security in MANET. *Cluster Computing*, Mar 2018. doi=10.1007/s10586-018-2612-2.

[101] S. Tan, X. Li, and Q. Dong. Trust based routing mechanism for securing OSLR-based MANET. *Ad Hoc Networks*, 30:84–98, 2015.

[102] K. Ullah and P. Das. Trust-based routing for mitigating grayhole attack in MANET. In J.K. Mandal, G. Saha, D. Kandar, and A.K. Maji, editors, *Proceedings of the International Conference on Computing and Communication Systems*, pages 713–721, Singapore, 2018. Springer Singapore.

[103] S. Vadlamani, B. Eksioglu, H. Medal, and A. Nandi. Jamming attacks on wireless networks: A taxonomic survey. *International Journal of Production Economics*, 172:76 – 94, 2016.

[104] A. Vasiliou and A.A. Economides. Mobile collaborative learning using multicast MANETs. *Int. J. Mobile Communications*, 5(4):423–444, 2007.

[105] W. Wang, H. Wang, B. Wang, Y. Wang, and J Wang. Energy-aware and self-adaptive anomaly detection scheme based on network tomography in mobile ad hoc networks. *Elsevier Information Sciences*, 220(20):580–602, 2013.

[106] X. Wang, K. Govindan, and P. Mohapatra. Provenance-based information trustworthiness evaluation in multi-hop networks. In *2010 IEEE Global Telecommunications Conference GLOBECOM 2010*, pages 1–5, Dec 2010.

[107] B. Wu, J. Chen, J. Wu, and M. Cardei. A survey of attacks and countermeasures in mobile ad hoc networks. In Y. Xiao, X.S. Shen, and D.Z. Du, editors, *Wireless Network Security, Signals and Communication Technology*, pages 103–135. Springer, 2007.

[108] Z. Wu, Y. Ou, and Y. Liu. A taxonomy of network and computer attacks based on responses. In *Proceedings of the International Conference on Information Technology, Computer Engineering and Management Sciences*, page 26–29, 2011.

[109] H. Yang, J. Shu, X. Meng, and S. Lu. SCAN: self-organized network-layer security in mobile ad hoc networks. *IEEE Journal on Selected Areas in Communications*, 24(2):261–273, Feb 2006.

[110] M. Yang, G. Hua, Y. Feng, and J. Gong. Software fault-tolerance techniques. In M. Yang, G. Hua, Y. Feng, and J. Gong, editors, *Fault-Tolerance Techniques for Spacecraft Control Computers*, pages 151–178. Springer, 2017.

[111] F. R. Yu, H. Tang, P. C. Mason, and F. Wang. A Hierarchical Identity Based Key Management Scheme in Tactical Mobile Ad Hoc Networks. *IEEE Transactions on Network and Service Management*, 7(4):258–267, December 2010.

[112] M. Guerrero Zapata. Secure ad hoc on-demand distance vector (SAODV) routing. IETF Internet Draft, IETF, 2005.

[113] Y. ZHANG and H-F. QIAN. An efficient identity-based secret key management scheme for MANETs. *The Journal of China Universities of Posts and Telecommunications*, 19:127–136, 2012.