Laboratoire des Systèmes Informatiques

جامعة هواري بومدين للعلوم والتكنولوجيا
USTHB

Rapport de Recherche

Djenouri Djamel & Nadjib Badache

LSI-TR0504 ***February 2004***

FACULTE ELECTRONIQUE & INFORMATIQUE
Département informatique
El Alia BP n°32 Bab Ezzouar 16111 Alger.
Tél / Fax : 213 (0) 21 24 79 17 - 24 76 07

A Survey on Security Issues in Mobile Ad hoc Networks

Djamel DJENOURI[§], Nadjib BADACHE[‡]

§: Basic Software Laboratory, CERIST Center of Research, Algiers, Algeria
$E-mail$: ddjenouri@mail.cerist.dz
‡: Computer Systems Laboratory, Computer Science Department, University of Science and technology, Algiers, Algeria
$E-mail$: badache@wissal.dz

abstract *An ad hoc network is a collection of mobile nodes equipped with wireless communication adapters, these nodes dynamically form a temporary network without the need of any existing network infrastructure. The absence of the central infrastructure imposes new challenges, since the services ensured by this central infrastructure must now be ensured by the mobile nodes themselves in this new environment. Moreover, other characteristics such as frequent changes of the topology, nodes' limitations (energy resource, storage device, CPU etc..) and communication channel limitations (bandwidth, reliability) add extra challenges. Earlier studies on ad hoc networks aimed to propose solutions to some fundamental problems, such as routing, coping with the new challenges caused by network's and nodes' features without taking the security issues into account. Hence, all these solutions are vulnerable to threats. More recent studies focused on the security problems in ad hoc networks.*
This paper is a survey on security problems in ad hoc networks and the current proposed solutions.

Keywords: mobile ad hoc networks, wireless networks, security.

1. Introduction

Nowadays, microprocessors and wireless adapters are embedded in many devices, as cell-phones, PDAs, Laptops, digital sensors, and GPS receivers. These well-equipped devices allow the creation of *wireless mobile* networks, which make the vision of nomadic computing with its ubiquitous access more and more attractive.
Some applications of mobile networks can not support the dependence on any fixed infrastructure. As example of such applications we quote; emergency disaster relief after a storm or an earthquake, this operation takes place in a damaged area, where no fixed infrastructure may still be available. Or a set of digital sensors placed to make some measures around a region unreachable to the human. Other examples of such applications are, a military tanks and planes in a battlefield, and students, researchers, or business associates sharing information during a lecture, a conference or a meeting. This infrastrructure independency requirement leads to a new kind of mobile networks; *ad hoc* networks.
A mobile ad hoc network, or MANET, is a temporary infrastructureless network, formed by a set of mobile hosts that dynamically establish their own network *on the fly*, without relying on any central administration.
Mobile hosts used in MANET have to ensure the roles that were ensured by the powerful fixed infrastructure in traditional networks. This is a challenging task, since these devices have limited resources (CPU, storage, energy, etc..). Moreover, the network's environment has some features that add extra complications, such as the frequent topology changes caused by nodes' mobility, and the unreliability and the bandwidth limitation of wireless channels.
Earlier studies on the ad hoc networks field aimed to propose solutions to some fundamental problems, coping with the new challenges caused by network's and nodes' features, these studies end to interesting new solutions. However, the problem with these solutions is that they do not take the security issues

into account, hence, they are vulnerable to threats. Whereas, many emergent applications designed for ad hoc networks necessitate robust security primitives and privacy protection. A robust security is also required to ensure fair and right functioning to the system, and to provide tolerable quality of service in such an open vulnerable environment.
Latest studies focused on the security problems in MANETs. But, and to the best of our knowledge, there is no previous published art-of-the-state wich presents and discuses security issues and the actual appropriate proposed solutions at *different network layers*.
This paper is a survey on the main research areas in this field, where we discuss the different security problems and the proposed solutions. In the rest of this paper, we first present basic concepts in the next section, followed by security issues regarding routing protocols in section 3, and those regarding data forwarding at the same layer (network) in section 4. In section 5, we deal with the nodes misbehavior problem both at the network and the MAC layers, and the solutions related to it. After that, we present the key management problem in section 6, followed by MANETS's Intrusion detection systems IDSs in section 7. Finally, section 8 concludes the paper.

2. Basic concepts

2.1. Security requirements:

The security services of ad hoc networks are not altogether different than those of other network communication paradigms. The goal is to protect the information and the resources from attacks and misbehavior. In dealing with network security, we shall explain the following requirements that an effective security paradigm must ensure:
Availability: ensures that the desired network services are available whenever they are expected, in spite of attacks. Systems that ensure availability seek to combat denial of service and energy starvation attacks that we will present later.
Authenticity: ensures communication from one node to another is genuine. It ensures that a malicious node cannot masquerade as a trusted network node.
Data confidentiality: is a core security primitive for ad hoc networks, It ensures that a given message cannot be understood by anyone else than its (their) desired recipient(s). Data confidentiality is typically enabled by applying cryptography [1].
Integrity: denotes the *authenticity of data* sent from one node to another. That is, it ensures that a message sent from node A to node B was not modified by a malicious node, C, during transmission. If a robust confidentiality mechanism is employed, ensuring data integrity may be as simple as adding oneway hashes [1] to encrypted messages.
Non-repudiation ensures that the origin of the message is legitimate. i.e when one node receives a false message from another, *nonrepudiation* allows the former to *accuse* the later of sending the false message and enables all other nodes to know about it. Digital signature [1] may be used to ensure nonrepudiation

2.2. MANETs features and their impact on security:

The features of MANETs make them more vulnerable to attacks and misbehavior than traditional networks, and imposes the security solution to be different from those used in other networks. These features are
Infrastructureless: Central servers, specialized hardware, and fixed infrastructures are necessarily absent. The lack of infrastructure precludes the deployment of hierarchical host relationships; instead, nodes uphold egalitarian relationships. That is, they assume contributory collaborative roles in the network rather than ones of dependence. i.e any security solution should rely on cooperative scheme instead of centralized one.
Wireless links use: The use of wireless links renders a wireless ad hoc network susceptible to

attacks. Unlike wired networks where an adversary must gain physical access to the network wires or pass through several lines of defense at firewalls and gateways, attacks on a wireless ad-hoc network can come from all directions and target at any node. Hence, a wireless ad hoc network will not have a clear line of defense, and every node must be prepared to threats. Moreover, since the channel is widely accessible, the MAC protocols used in ad hoc networks, such IEEE802.11, rely on *trusted* cooperation in a neighborhood to ensure channel access, which presents a vulnearability.

Multi-hop: Because the lack of central routers and gateways, hosts are themselves routers, then packets follow multi-hop routes and pass through different mobile nodes before arriving to the destination. Because of the possible untrustworthy of such nodes, this feature presents a serious vulnerability.

Nodes movement autonomy: mobile nodes are autonomous units that are capable of *roaming* independently. This means that tracking down a particular mobile node in a large scale ad hoc network cannot be done easily.

Amorphous: Nodes mobility and wireless connectivity allow nodes to enter and leave the network spontaneously. Therefore, the network topology has no form regarding both the size and the shape. Hence, Any security solution must take this feature into account.

Power limitation: Ad hoc enabled mobile nodes are small and lightweight, therefore, they are often supplied with limited power resources, small batteries, to ensure portability. The security solution should take this restraint into account. Furthermore, this limitation causes a vulnerability since a node powering-off can cause its break-down. Thereby, attackers may targets some nodes batteries to disconnect them, even to make network partition. This is called energy starvation attack or sleep deprivation torture attack [2].

Memory and computation power limitation: Ad hoc enabled mobile nodes have limited storage devices and weak computational capabilities. High complexity security solutions employed, as cryptography, should take these constraints into consideration.

2.3. Threats

We divide threats that can affect security in ad hoc networks into two classes,

2.3.1. Attacks

It includes any action that *intentionally aims to cause any damage* to the network, it can be divided according to their origins or their nature. Origin based classification splits attacks up into two categories; external and internal, whereas, nature based classification splits them up into passive attacks and active attacks

External attacks: This category Includes attacks launched by a node that do not belong to the *logical* network, or is not allowed to access to it. Such a node penetrates the network area to launch its attack [3].

Internal attacks: This category includes attacks launched by an internal *compromised* node, It is a more several kind of threat to the network since the proposed defence toward external attacks is ineffective against compromised and internal malicious nodes [3].

Passive attacks: A passive attack is a continuous collection of information, these information would be used later when launching an active attack. That means the attacker eavesdrops packets and analyzes them to pick up required information. The security attribute that must be provided here is information *confidentiality*.

Active attacks: Include almost all the other attacks launched by actively interacting with victims, like *sleep deprivation torture* that aims the batteries charges, *hijacking*, in which the attacker takes control of a communication between two entities and masquerades as one of them, *jamming*, that causes channel unavailability, attacks against routing protocols, etc... most of these attacks result in a denial of service (DoS), that is a degradation or a complete halt in communication between nodes.

2.3.2. Misbehavior

We define misbehavior threats as an unauthorized behavior of an *internal* node that can result *unintentionally* in a damage to other nodes, i.e the aim of the node is not to launch an attack, but it may have other aims as obtaining an unfair advantage compared with the other nodes. For instance, a node may do not adhere to the MAC protocol, with the intent of getting higher bandwidth, or it may accept but do not forward packets on behalf of other nodes to safe its resources.

2.4. Defence classes

We divide security solutions into two categories
proactive solution: It consists of security-aware protocols and applications design, These protocols must take the new environment features into consideration.
reactive solution: Since proactive solution is insufficient, because the system is complex then vulnerable to design and program errors or lucks, reactive solution is as a second security wall, in other words, it consists of attack detection. Intrusion detection systems belong to this class.

Up to now, we have presented some basic notions regarding security in ad hoc networks, in the following sections we will deal with the current research areas on securing ad hoc networks, and will discus the existing problems and the proposed solutions.

3. Routing security issues

A MANETs' routing protocol finds routes between nodes, then allows data packets to be forwarded through other network's nodes towards the final destination. In contrast to traditional network routing protocols, ad hoc network routing protocols must adapt more quickly to cope with MANETs factors presented previously, especially the frequent change of the network topology. This problem of routing in ad hoc networks is an important one and has been extensively studied, particularly in the MANET working group of the Internet Engineering Task Force (IETF). This study has resulted in several mature protocols as [4, 5, 6, 7, 8, 9, 10] that can be divided into two classes; proactive (table driven) and reactive (on-demand), a survey on the two classes of routing protocols is available in [11], we have remarked in [12], that reactive protocols are more adaptable to MANET environment than proactive ones. However, the problem with all these solutions is that they do not take the security factor into account, therefore, these solutions are vulnerable to many attacks.
Since MANETs environment is untrusted, a secure routing protocol is required. Recently, several secure MANET routing protocols have been proposed, [13, 14, 15, 16, 17, 18, 19, 20, 21]. In this section we deal with the security issues of routing protcols, we first present a classification of different attacks that threat earlier MANET routing protocols, then we discus the recent proposed solutions.

3.1. Routing protocols attacks classes

The current proposed routing protocols for MANET are subject to many different type of attacks. Analogous exploits exist in wired networks [22], but are more easily defended against by the infrastructure present in a wired network. In this subsection, we classify *modification*, i*mpersonation*, and *fabrication* exploits against ad hoc routing protocols as in [18], moreover, we add a new kind of attacks, *Rushing attacks*, recently defined by Hu et al [20]. The attacks presented below are described in terms of the AODV and DSR protocols, which are used as representatives of ad hoc on-demand protocols, almost all on-demand protocols have the same vulnerabilities. We think that table driven approach is unsuitable for MANET, so it is excluded from the issue. Table 1 provides a summary of each protocol's vulnerability to the following exploits.

Attack	AODV	DSR
Attacks using modification		
modifying route sequence numbers	yes	no
modifying hop counts	yes	no
modifying source route	no	yes
Tunneling	yes	yes
Attacks using impersonation		
spoofing	yes	yes
Attacks using fabrication		
Falsifying Route Errors	yes	yes
Route cache poisoning	no	yes
Rushing attacks	yes	yes

Table 1: Vulnerabilities of AODV and DSR

3.1.1. Attacks using modification

Malicious nodes can cause redirection of the network traffic and DoS attacks by *altering* control message fields or by forwarding routing messages with *falsified* values. For example, in the network illustrated in Figure 1a, a malicious node M could keep traffic from reaching X by consistently advertising to B a shorter route (or generally more optimal) to X than the route to X that C advertises. Below are detailed several of the attacks that can occur if particular fields of routing messages in specific routing protocols are altered or falsified.

Redirection by modified route sequence numbers: Protocols such as AODV [7] instantiate and maintain routes by assigning monotonically increasing sequence numbers to routes toward specific destinations. In AODV, any node may divert traffic through itself by advertising a route to a node with a destination_sequence_num greater than the authentic value. Figure 1b illustrates an example of an ad hoc network. Suppose a malicious node, M, receives the RREQ that originated from S for destination X after it is re-broadcasted by B during route discovery. M redirects traffic toward itself by unicasting to B an RREP containing a much higher destination_sequenc_num for X than the value last advertised by X. Eventually, the RREQ broadcast by B will reach a node with a valid route to X and a valid RREP will be unicast back toward S. However, at that point B will have already received the false RREP from M. If the destination_sequenc_num for X that M used in the false RREP is higher than the destination_sequence_ num for X in the valid RREP, B will drop the valid RREP, thinking that the valid route is stale. All subsequent traffic destined for X that travels through B will be directed toward M. The situation will not be corrected until either a legitimate RREQ or a legitimate RREP with a destination_sequence_nim for X higher that of M's false RREP enters the network.

Redirection with modified hop counts: A redirection attack is possible by modification of the hop count field in route discovery messages. When routing decisions cannot be made by other metrics, AODV uses the hop count field to determine the shortest path. In AODV, malicious nodes can increase the chances they are included on a newly created route by resetting the hop count field of the RREQ to zero. Similarly, by setting the hop count field of the RREQ to infinity, created routes will tend to not include the malicious node. Such an attack is most threatening when combined with spoofing, as detailed later. Even if the protocol uses other metric than hop count, the redirection attack against it is possible, all what the attacker has to do is to modify the field used to compute the metric instead of hop count.

DoS with modified source route: DSR [4] utilizes source routes strategy, thereby source nodes explicitly states routes in data packets. These routes lack any integrity checks and a simple denial-of-

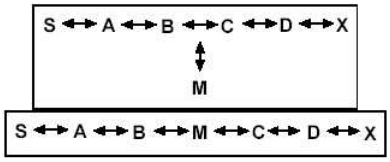

Figure 1: (a) and (b) 2 examples of ad hoc networks [18]

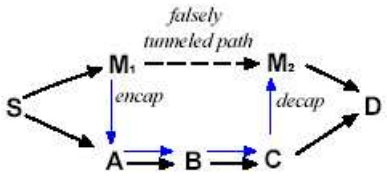

Figure 2: Tunnelling [18]

service attack can be launched in DSR by altering the source routes in packet headers. Assume a path exists from S to X as in Fig.1b. Also assume that C and X cannot hear each other, that nodes B and C cannot hear each other, and that M is a malicious node attempting a denial-of-service attack. Suppose S wishes to communicate with X and that has an unexpired route to X in its route cache. S transmits a data packet toward X with the source route (S,A,B,M,C,D,X) contained in the packet's header. When M receives the packet, it can alter the source route in the packet's header, such as deleting D from the source route. Consequently, when C receives the altered packet, it attempts to forward the packet to X. Since X cannot hear C, the transmission is unsuccessful.

Tunneling: Ad hoc networks have an implicit assumption that any node can be located adjacent to any other node. A tunneling attack is where two or more nodes may collaborate to encapsulate and exchange messages between them along existing data routes. One vulnerability is that two such nodes may collaborate to falsely represent the length of available paths by encapsulating and tunneling between them legitimate routing messages, as route discovery (RREQ) and route reply (RREP) packets , generated by other nodes. resulting in the prevention of intermediate nodes from correctly incrementing the metric used to measure path lengths (as hop count). for example, in figure 2, M_1 and M_2 are malicious nodes, they are not neighbors but they can use the path (M_1, A, B, C, M_2) as a tunnel. When M_1 receives a route request packet (RREQ) from S, it encapsulates it and tunnels it to M_2, this one forwards it. After M_2 gets the RREP from D, it forwards it back to M_1, this later does so to S, the result is the construction of a short wrong route $(M_1, M2)$, which may be selected as the optimal one by S.

3.1.2. Attacks using impersonation (spoofing)

Spoofing occurs when a node *misrepresents its identity* in the network, such as by altering its MAC or IP address in outgoing packets, and is readily combined with modification attacks. These two attacks when combined with each other may result in serious misinformation, as creating route loops[18].

3.1.3. Attacks using fabrication

The generation of false routing messages can be classified as fabrication attacks. Such attacks can be difficult to verify.

Falsifying Route Errors: On-demand routing protocols, among them AODV and DSR, implement path maintenance to recover broken paths when nodes move. If a link of an active route from node S to node D breaks down, the node upstream of the link broadcasts a route error message to all active upstream neighbors. The node also invalidates the route for this destination in its routing table. If S has no other route to D and a route still be needed to this destination, S initiates again a route discovery. The vulnerability is that routing attacks can be launched by disseminating false route error messages. Resulting in packets lose, and/or extra overhead [18].

Route cache poisoning: In DSR, a node can update its routing table (*route cache*) relying on information in headers held by packets they receive to forward. Routes can also be learnt from promiscuously received packets.
The vulnerability is that an attacker could easily exploit this method of learning routes and poison route caches. Suppose a malicious node M wanted to poison routes to node X, if M were to broadcast spoofed packets with source routes to X via itself, neighboring nodes that overhear the packet transmission may add the wrong route to their route caches [18].

3.1.4. Rushing attacks

Recently, Hu et al [20] have defined a new attack, called rushing attack.
In almost all on demand routing protocols, to limit the route discovery overhead, each node forward only one RREQ originated from any route discovery, generally the one that arrives first. This property can be exploited by *rushing* the forwarding of received RREQ.
If the RREQs for a discovery forwarded by the attacker are the first to reach each neighbor of the target, then any route discovered by this route discovery will include a hop through the attacker. That is, when a neighbor of the target receives the rushed REQUEST from the attacker, it forwards that RREQ, and will not forward any further RREQ from this route discovery. When non-attacking RREQs arrive later at these nodes, they will be dropped. As a result, the initiator will be unable to discover any usable routes (i.e., routes that do not include the attacker) containing at least two hops (three nodes). In general terms, an attacker that can forward RREQs more quickly than legitimate nodes can do so, which can increase the probability that routes that include the attacker will be discovered rather than other valid routes. Whereas the discussion above has used the case of nodes that forward only the first RREQ from any route discovery, the rushing attack can also be used against any protocol that predictably forwards any *particular* RREQ for each route discovery, the attacker do the same thing expected that packets it sends must fit the appropriate feature.
How an attacker can perform a rushing attack? to launch a rushing attack, a malicious can use one or more of the following techniques:

- Remove MAC and/or network delays when forwarding packets: MAC and Network layer protocols use delays on packets transmission to avoid collusions, an attacker may deny these delays to rush the request it forwards.

- Transmit RREQs in higher power: An attacker supplied with a powerful physical communication support can use a higher transmission power to forward RREQ, this power allows it to cover a higher power range than other nodes, thereby further nodes will be reached in less hops.

- Employing wormhole technique [23]: two attackers can use a tunnel (presented previously) to pass RREQ packet among them. This can be done when one node is closer to the source and

the other is closer to the destination, and a path with high quality exists between the two nodes (eg, via a wired network).

3.2. Secure routing protocol requirements

A good secure routing protocol aim is to prevent each of the exploits presented in the previous Section. For this purpose, it must satisfy the following requirements

- Routing packets cannot be spoofed
- Fabricated routing messages cannot be injected into the network
- Routing messages cannot be altered in transit
- Routing loops cannot be formed through malicious actions
- Routes cannot be redirected from the shortest path by malicious actions
- Unauthorized nodes should be excluded from route computation and discovery
- The network topology must not be exposed neither to adversaries nor to authorized nodes by the routing messages, since exposure of the network topology may be an advantage for adversaries trying to destroy or capture nodes.

3.3. Solutions

3.3.1. Authentication during all phases

This solution consists of using authentication techniques during all phases, thereby excluding attackers or unauthorized nodes to participate in the routing. Most of the proposed solutions belonging to this class modify existing routing protocols to build authentic ones[18, 15, 13, 21]. They rely on Certificate Authority (CA) presence. For instance, the solution presented in [18] requires the use of a trusted certificate server whose public key is *priory* known to all valid nodes, this renders the solution centralized and less flexible.

3.3.2. Define new metrics

Yi et al [21] define a new metric that governs the routing protocol behavior called trust value. This metric is to be embedded into control packets to mirror the minimum trust value required by the sender, thereby a node that receives any packet can neither process it nor forward it unless it provides the required trust level presented in the packet. For this purpose, authentication is used to design SAR (Security-Aware Routing), a protocol derived from AODV and based on the trust values metric. In SAR, this metric is also used as a criterion to select routes when many routes satisfying the required trust value are available. To define nodes' trust values, authors address the example of military context, when trust level matchs to node's owner rank. But in the general context, where there is no hierarchy in the network, defining the nodes' trust values is problematic.

3.3.3. Secure neighbor detection

It consists of a three round authenticated message exchange between two nodes before each one claims the other as neighbor. If this exchange fails, then the well behaving node ignore the other, and does not handle packets sent by it. This solution beats the illegal use of high power range to launch the rushing attacks. Since the sender using higher powers can not receive the packet from further nodes, it will not be able to perform the neighbor detection process, then their packets will be ignored by these nodes[20].

3.3.4. Randomize message forwarding

This technique is proposed by Yi et al[20] to minimize the chance that a rushing adversary can dominate all returned routes. In traditional RREQ forwarding, the receiving node immediately forwards the first RREQ and discards all subsequent RREQ. Using this scheme, a node first collects a number of RREQs, and selects a RREQ at *random* to forward. There are thus two parameters to randomized forwarding technique: first, the number of REQUEST packets to be collected, and second, the algorithm by which timeouts are chosen. A detailed description is available in [20].
We think the drawback of this solution is that it increases the delay of route discovery, since each node must wait for a timeout or up to receiving a given number of packets before forwarding the RREQ. Moreover, the random selection prevents the discovery of optimal routes, optimality may be in defined as hops number, energy efficiency [24], or according to other metrics, anyway it is not random.

4. Data forwarding security issues

Protecting the network layer in a MANET is an important research topic of wireless security. The core functionalities provided in the network layer are routing and packet forwarding, malicious attacks on either of them will disrupt the normal network operations. Although several recent proposals have addressed the problem of secure ad hoc routing, as shown previously, protection of data forwarding service has received relatively less attention except the works of [25, 26, 27]. In this section we deal with the issue of protecting packet forwarding. As in the previous section, we first present attacks that threat data forwarding, then we discuss current solutions.

4.1. Data forwarding attacks

4.1.1. Eavesdropping

The wireless channels used in MANETs are freely and easily accessible. Moreover, promiscuous mode, which means capturing packets by a node that is not the appropriate destination, is employed by protocols to operate or to ensure more efficiency, eg. a routing protocol may use this mode to learn routes. These features can be employed by malicious to eavesdrop data in transit. The obvious proactive solution against this is to use cryptography, this solution just ensures confidentiality, but does not prevent eavesdropping, and to the best of our knowledge, no detecting solution is available.
Since breaking keys is always possible and using a robust key revocation within MANET is problematic, eavesdropping is a serious attack against data forwarding.

4.1.2. Dropping data packets

Since packets follow multi-hop routes, a malicious can participate in routing and drop all packets it receives to forward. To do this, it first attacks the routing protocol to gain participation in routing, using one or more of the attacks presented previously.

4.1.3. Inject forged data packet

A malicious may fabricate data packets to inject and disperse them with no other interest than overloading the network, this can result in disruption of forwarding legal packets.

4.2. Solutions

4.2.1. End-to-end Security Association (SA)

Security association or trust relationship can be instantiated, for example, by the knowledge of the public key of each other. This SA prevents malicious from Injecting forged data packets.

4.2.2. Information dispersal

It consists of dispersing data over different routes. The source first invokes the underlying route discovery protocol, and then determines an initial set of paths for communication with the specific destination, called APS (Active Path Set). In order to ensure high dispersal, it is preferable for these routes to be node-disjoint.

With a set of routes at hand, the source disperses each outgoing message into a number of pieces. At the source, the dispersal introduces *redundancy* and encodes the outgoing messages. For this purpose, the algorithm proposed in [28] can be used, then each dispersed piece is transmitted across a different route. At the destination, a dispersed message is successfully reconstructed provided that sufficiently a given number of pieces are received, in other words, the message dispersion ensures successful reception *even if a fraction of the message pieces is lost or corrupted*. A detailed description of the use of such a technique is available in [26, 29]

4.2.3. Use of feedback

To prevent data forwarding procedure from drooping attacks, feedback can be used. That is, the destination validates the reception of packets by sending back cryptographical protected feedback or ACK to the source. The obvious problem with this solution is that it increases overhead.

4.2.4. Secure Message Transmission (SMT)

Panagiots et al [26] have proposed a protocol called Secure Message Transmission which aims to safeguard data transmission. They have defined and used all the solutions described above (End-to-end SA, Information dispersal, Use of feedback) to design SMT; a network layer protocol over routing protocol that aims to protect data transmission when operating above any multi route routing protocol. A full description of this solution is available in [26].

4.2.5. According admissions and monitoring in neighborhood

It consists of according admissions to nodes to allow them to interact with others and participate in the network, since there is no centralized administration, the admission decision is a very difficult task. One of the possible solution to this problem is to operate in neighborhood, that means nodes in a neighborhood accords mutually participation admissions. In the following we explain this concept.
Yang et all [27] describe a unified network layer solution to protect both routing and data forwarding in the context of AODV. This is done by exploiting full localized design without assuming prior trust or secret association between nodes.
Each node has a token issued by its local neighbors which allows it to participate in the network operations. These neighbors collaboratively monitor it to detect attacks. The token has a period of expiration whose value depends on how long the node has been behaving well, the node renews the token upon its expiration. The framework of this solution consists of four closely interacted compounds (figure 3):

- Neighbor Verification: which describes how to verify whether each node in the network is legitimate or malicious.

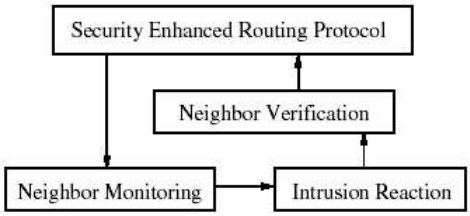

Figure 3: Framework of the solution presented in[27]

- Security Enhanced Routing Protocol, which explicitly incorporates the security information into the ad hoc routing protocol.
- Neighbor Monitoring, which describes how to monitor the behavior of each node in the network and how to detect occasional attacks from malicious nodes.
- Intrusion Reaction, which describes how to alert the network and isolate the attackers.

The full detailed description of the four compounds is presented in [27]. In the following we describe the first two components which are related to our class of solutions (According admissions and monitoring in neighborhood).

Neighbor verification: There is a key pair SK/PK (Secret Key and Public Key), each token carried by a node is signed with a global secret key SK, and is broadcast periodically in the hello message to ask a new validation. A token is valid if and only if it fulfills the following conditions: (i) It is held (sent) by the node with the same ID as the one stated in the token, (ii) It has not expired, (iii) It has been signed by SK. There are 3 critical questions regarding how to issue the token for the node:
Who is responsible for issuing the token? The answer is that each node will participate by a polynomial order K-1, i.e it has a different partial key which is as a part of the SK, and provide a *partial signature of order K*. K different partial signatures are sufficient to provide the right signature. This technique is called polynomial secret sharing [30, 31].
How do the nodes obtain their token? Consider the case that a node in the network, which has already possessed a token, needs to renew its current token. The message handshake in the localized token issuing process is illustrated in Figure 4. Before its current token expiration time, a node broadcasts a TREQ (Token Request) packet to its neighbors, which contains its current token and a timestamp. Each node also keeps a Token Revocation List (TRL) learnt from the intrusion reaction component. When a node receives a TREQ packet, the TRL will be used to decide whether to serve the request or not. Specifically, when a node receives a TREQ packet from its neighbor, it extracts the token from the packet, it checks whether the TREQ packet comes from the owner of the token therein, and whether the token has already been revoked by comparing it with the TRL. If the token is still valid and the source of the TREQ packet matches the owner of the token, it constructs a new token in which owner identity is equal to that in the old token, in this new token the signing time is equal to the timestamp in the TREQ packet, and the expiration time is increased. It then signs this newly constructed token using its own share of SK, encapsulates the partially signed token in a TREP (Token Reply) packet, and then unicasts the TREP packet back to the node from which it received the TREQ packet. When the node which needs to renew its token receives k TREP packets from different neighbors, it can combine these partially signed tokens into a single token signed by SK.
There is another case of token issuing: a newly joined node needs to obtain its first token. This is similar to the token renewing case from the message handshake perspective. In order to join the network, a node also broadcasts a TREQ (Token Request) packet, containing its identity and the current time, in

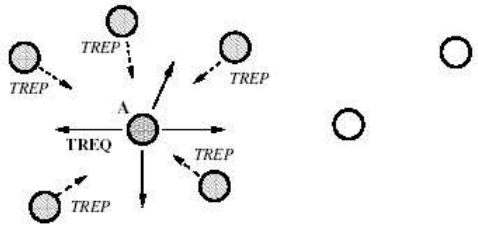

Figure 4: Localized token issuing[27]

its local neighborhood. Its neighbors apply the same rules as described above to determine whether to serve this request, if they decide to issue the token they apply the same process to construct and send back the partially signed tokens. However, the expiration time field in the first token is different from that in a renewed token.
*What is the period of the token's expiration ?*To decrease communication overhead, this period is increased by a constant T after each period of renewing, relying on the following assumption: "the node that stays and behaves well for more time will be more trusted". We think this is not inevitably true, and can present a vulnerability.

Neighbor monitoring: each node is responsible for monitoring its neighbors and detecting any attack or misbehavior in both routing or data packets, by using promiscuous mode and analyzing captured packets. For this purpose, the mechanism presented in [25], which we will explain latter, is used. We note that nodes cooperate in neighborhoods to improve accuracy.
We think this solution has some drawbacks, first it prevents a node which has less than k neighbors to communicate. Moreover, the choice of this parameter (k) is a critical issue, the choice of low K weakens the key (It will be more breakable), whereas the choice of high values requires high connectivity which is not always ensured in MANETs. Furthermore, this solution uses the notion of token expiration, this requires a timestamping mechanism which is not obvious in distributed systems. Anyway, there are some tolerable solutions as the use of logical clocks or GPS, but this was not treated by the others.

5. Securing against misbehavior

As we have seen, because of the lack of any central administration and their infrastructureless feature, MANETs rely on the nodes cooperation to ensure services. Because of their limitation, MANETs' nodes may *misbehave* and do not respect the general rules to access the services. For instance, a node may do not forward packets on behalf of other nodes. This represents a misbehavior, if a node disseminates its data packet through the network and relies on other nodes to route them, it must be cooperative, otherwise excluded from the network. Another example is the no-adherence of nodes vis-a-vis the channel access rules for the purpose of gaining high throughput. Generally speaking, misbehavior in MANET may be seen as *internal* nodes' *unfair selfish* behavior resulting in the no respect of rules required to ensure fair services. Even though this is not an attack, it is a severe threat that menaces one of the security requirements, namely the availability. Just few works have dealt with this problem and consider selfish behavior as a misbehavior. In this section we present and discus two kinds of misbehavior.

5.1. misbehaving on channel access

5.1.1. The problem

Since there is no central authority in MANETs, Wireless Medium Access Control (MAC) protocols, such as IEEE 802.11, use distributed contention resolution mechanisms for sharing the wireless channel. The contention resolution is typically based on cooperative mechanisms that ensure a reasonably fair share of the channel for all the participating nodes. In this environment, some selfish hosts in the network may misbehave by failing to adhere to the network protocols, with the intent of obtaining an unfair share of the channel. The presence of selfish nodes that deviate from the contention resolution protocol can reduces the throughput share received by conforming nodes.

IEEE 802.11 MAC protocol [32], which is the standard MAC protocol for MANET, has two mechanisms for contention resolution; a centralized mechanism called PCF (Point Coordination Function), and a fully distributed mechanism called DCF (Distributed Coordination Function). PCF needs a centralized controller (such as a base station) and can only be used in infrastructure-based networks, thus it is not to be considered in the ad hoc mode. Whereas, DCF is widely used in infrastructure-based wireless networks as well as in ad hoc wireless networks.

DCF uses CSMA/CA (Carrier Sense Multiple Access/ Collision Avoidance) option for resolving contention among multiple nodes accessing the channel. A node (sender) with data to transmit on the channel selects a *random backoff* value from range [0;CW], where CW (Contention Window) is a variable maintained by each node. While the channel is idle, the backoff counter is decremented by one after every time slot (time slot is a fixed interval of time defined in IEEE 802.11 standard), and the counter is frozen when the channel becomes busy. The node may access the channel when the backoff counter is decremented to zero. After the backoff counter is decremented to zero, the sender may reserve the channel for the duration of the data transfer by exchanging control packets on the channel. The sender first sends a RTS (Request to Send) packet to the receiver node, then the receiver responds with a CTS (Clear to Send). This RTS-CTS exchange is optional in IEEE 802.11, it aims to ensure the channel reservation for the duration of data transmission. Both of the packets contain the proposed duration of data transmission, other nodes which overhear either the RTS or the CTS (or both) are required to defer transmissions on the channel for the duration specified in RTS/CTS. After a successful RTS/CTS exchange, the sender transmits a DATA packet, then the receiver responds with an ACK packet to acknowledge a successful reception of the DATA packet. If a nodes data transmission is successful, the node resets its CW to a minimum value (CWmin), otherwise, if a nodes data transmission is unsuccessful (detected by the absence of a CTS or the absence of an ACK), CW is doubled, but it should not exceed a maximum value of CWmax. A misbehaving node may obtain more than its fair share of the bandwidth by:

- Selecting backoff values from a different distribution with smaller average backoff value than the distribution specified by DCF (e.g., by selecting backoff values from range [0,CW/4] instead of [0,CW]).
- Using a different retransmission strategy that does not double the CW value after collision.

We note that it is not beneficial for selfish to not delay at all or to choose a very small constant period, since this may result in a very high collision rate hence in the loss of the packets it sends.
Such selfish misbehavior can seriously degrade the throughput of well-behaved nodes. For instance, simulation results obtained by Kyasanur and Vaidya[33] show that for a network containing 8 nodes sending packets to a common receiver with one of the 8 nodes misbehaving by selecting backoff values from range [0,CW/4], the throughput of the other 7 nodes is degraded by as much as 50%.
To the best of our knowledge, there is no published solution proposed to this problem, expected the one of Kyasanur and Vaidya [33], in the following we present and discus this solution.

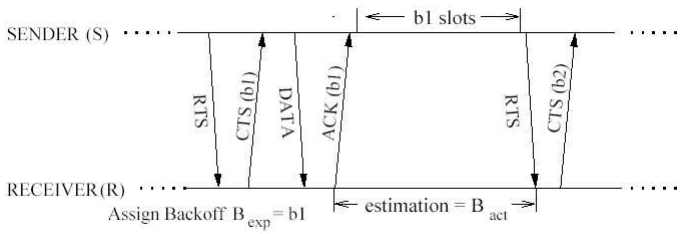

Figure 5: Receiver-Sender interaction [33]

5.1.2. The solution

Due to the random selection of the backoff, it is hard to distinguish between the legitimate selection of small backoff values and a misbehaving selection. Hence, detecting a MAC layer misbehavior is a complicated problem. Kyasanur and Vaidya [33] have proposed a scheme to resolve this problem.
It consists of modifications to the IEEE 802.11 protocol that enable a receiver to identify sender misbehavior within a small observation interval. Instead of the sender selecting random backoff values to initialize the backoff counter, the receiver selects a random backoff value, b1, and sends it in the CTS and ACK packets to the sender. The sender uses this assigned backoff value in the next transmission to the receiver. Figure 5 summarizes this exchange. With these modifications, a receiver can identify senders deviating from the protocol by observing the number of idle slots between consecutive transmissions from the sender, B_{act}. If this observed number of idle slots is less than the assigned backoff, then the sender may have deviated from the protocol. The magnitude of observed deviations over a small history of received packets is used to diagnose sender misbehavior with high probability. The proposed scheme also attempts to negate any throughput advantage that the misbehaving nodes may obtain. To achieve this and to discourage misbehavior, deviating senders are penalized,i.e when the receiver perceives a sender to have waited for less than the assigned backoff, it adds a penalty to the next backoff assigned to that sender.

5.1.3. Discussion

If we assume *the receiver is a well-behavior*, when the sender does not backoff for the duration specified by the penalty (or backs off for a small fraction of the duration), it significantly increases the probability to be detected as a misbehaving. On the other hand, a misbehaving sender which backs off for the duration specified by the penalty (or a large fraction of it) does not obtain significant throughput advantage over other well-behaved nodes. However, there some problems with this solution:

- Deviations observed are not inevitably a misbehavior, they may be caused by the channel condition difference between the sender and the receiver or what is known as the hidden node phenomena, to understand this we give the following example: consider the situation of three nodes A, S, and R, where R is the receiver and S the sender in our context, i.e R is monitoring S after sending it a backoff. We assume R is within the power range of A, but S is not so (figure 6), we also assume that A is sending packets. Hence the R's channel is busy, whereas S's channel is free, consequently, S decreases its counter upon each slot unlike R, then the former will be able to access to the channel when its backoff counter reaches 0, this access will be considered by R as a deviation, if such a situation lasts R will register enough deviations to consider wrongly S misbehaving. So this solution may lead to accuse *innocent* nodes.

- We define a new misbehavior on channel access that we call *cooperative misbehavior*, we consider

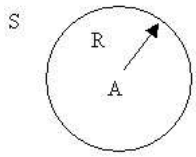

Figure 6: Nodes' positions

two nodes that wish to obtain unfairly high throughput to exchange packets, when they are in the same neighborhood (they are neighbors), the receiver may misbehave and assign unfair backoff values to the corresponding sender. The proposed scheme fails with this situation since it relays on receivers trustworthy.

5.2. Misbehavior on routing

5.2.1. The problem

MANETs' Routers are the hosts themselves, and the routing relies on nodes cooperation, the more nodes that participate in packets routing, the better the efficiency. However, a selfish node that excepts others to forward packets on its behalf but is unwilling to spend its battery life, CPU cycles, or available network bandwidth to forward packets not of direct interest to it, may misbehave by agreeing to forward packets and then does not do so. We think this must be considered as an unfair and unauthorized behavior.

5.2.2. The solution

The problem of non-cooperative nodes has been addressed lately. In [34, 35, 36, 37], the idea that users may not want to cooperate because of their battery constraints is introduced and simple rules are proposed, which can be used to determine whether a user should forward other nodes traffic or not. Snirivasan et al [37] define a parameter, called sympathy, that reflects the level of selfishness/ altruism of the nodes, and on which routing decisions rely. The purpose is to allow users to be selfish if they need to be so, since they consider selfish as a legal behavior required when the node needs to save its capacity.
In [34, 35], the authors propose a method based on the introduction of a virtual currency, the so-called nuglets. Every network node has an initial stock of nuglets, either the source or the destination of each traffic connection use nuglets to pay the relay nodes for forwarding data traffic. The cost of a packet may depend on several things, such as the required transmission power and the nodes battery status. Packets sent by or destined to nodes that do not have a sufficient amount of nuglets are discarded.
In [36], a simpler mechanism is proposed, which makes source nodes pay as many battery units as the estimated number of nodes on the path to the destination, and makes relay nodes earn as many battery units as the number of forwarded packets.
All these interesting solutions, however, do not consider selfish as misbehavior, hence, they do not aim to detect any node misbehavior, but they just introduce mechanisms to stimulate cooperation.
Marti et al [25] are the first who propose a solution to detect and mitigate selfish misbehavior on routing, in the following we overview this solution.
To mitigate this problem, Marti et al propose categorizing nodes based upon their dynamically measured behavior. They define two techniques which they called *watchdog* and *pathraterthat*, the first one is to identify misbehaving nodes whereas the second helps routing protocols to avoid routing through these nodes. These techniques are used along with DSR to built a misbehavior mitigating routing

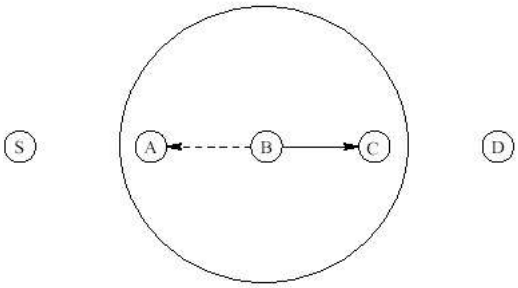

Figure 7: watchdog method

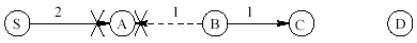

Figure 8: ambiguous collision problem

protocol

Watchdog: The watchdog method detects misbehaving nodes Figure 7 illustrates how the watchdog works, suppose there exists a path from node S to D through intermediate nodes A, B, and C. Node A cannot transmit all the way to node C, but it can listen in on node B's traffic, thus when A transmits a packet for B to forward to C, A can often check if B forwards the packet.
The watchdog is implemented by maintaining a buffer of recently sent packets and comparing each overheard packet with the packet in the buffer to see if there is a match. If so, the packet in the buffer is removed and forgotten by the watchdog, since it has been forwarded on. If a packet has remained in the buffer for longer than a certain timeout, the watchdog increments a *failure tally* for the node responsible for forwarding the packet, if the tally exceeds a certain threshold, it determines that the node is misbehaving and sends a message to the source notifying it of the misbehaving node.
The watchdog technique has advantages and weaknesses. DSR with the watchdog has the advantage that it can detect misbehavior at the forwarding level and not just at the link level. However, watchdog might not detect a misbehaving node in the presence of:

- The ambiguous collision problem: It prevents A from overhearing transmissions from B. As Figure 8 illustrates, a packet collision can occur at A while it is listening for B to forward on a packet, A does not know if the collision was caused by B forwarding on a packet as it should or if B never forwarded the packet and the collision was caused by other nodes in A's neighborhood. Because of this uncertainty, Marti et all [25] suggest that A should not immediately accuse B of misbehaving, but should instead continue to watch B over a period of time, if A repeatedly fails to detect B forwarding on packets, then A can assume that B is misbehaving. However, this suggestion do not prevent false accusation when collusion frequency at A is high, furthermore it makes the follwing vulnerability.

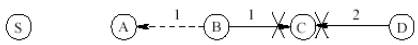

Figure 9: receiver collision problem

- Partial dropping: A node can circumvent the watchdog by dropping packets at a lower rate than the watchdog's configured minimum misbehavior threshold.
- Receiver collision problem: in this problem node A can only tell whether B sends the packet to C, but it cannot tell if C receives it (Figure 9). If a collision occurs at C when B first forwards the packet, A only sees B forwarding the packet and assumes that C successfully receives it. Thus, B could skip retransmitting the packet.
- False misbehavior: Another problem can occur when nodes falsely report other innocent nodes as misbehaving. By claiming that some nodes following it in the path are misbehaving, a selfish node could attempt to avoid getting packets to forward and a malicious could attempt to partition the network . For instance, node A could report that node B is not forwarding packets when actually it is, this will cause S to mark B as misbehaving when A is the culprit.
- Insufficient transmission power: Another problem is that a misbehaving node that can control its transmission power can circumvent the watchdog. A node could attempt to save its energy consumed for forwarding packets by adjusting its transmission power such that the power is strong enough to be overheard by the previous node but too weak to be received by the true recipient. This would require that the misbehaving node knows the transmission power required to reach each of its neighboring nodes, and to be closer to the previous node than the true recipient. Another issue we remark related to the transmission power control is the possibility of false detections (false positives). Assume three nodes A, B and C such that A monitors B's forwarding to C and B uses controlled power, we also assume the required power from B to C to be less than the one from B to A, thereby the packets sent from B to C will not be received at A. The node A may accuse wrongly B as misbehavior even though it forwards packets to C.
- Cooperated misbehavior: For example, B and C from Figure 7 could collude to cause mischief. In this case, B forwards a packet to C but does not report to A when C drops the packet.

Pathrather: The pathrater, run by each node in the network, combines knowledge of misbehaving nodes with link reliability data to pick the route most likely to be reliable. Each node maintains a rating for every other node it knows about in the network which reflects its movement stability and routing actively, more the node is active and forwards *successfully* packets more its rating increases. Each node computes a path metric by averaging the node ratings in the path, If there are multiple paths to the same destination, the path with the highest metric is chosen. Note that this differs from standard DSR, which chooses the shortest path in the route cache. Further note that since the pathrater depends on knowing the exact path a packet has traversed, it must be implemented on top of a source routing protocol. The pathrather excludes routes containing any node known as misbehaving. The full algorithm of rating assignment is avalaible in [25]

6. Key management

Key management system is an underlying mechanism for securing both networking functions (e.g., routing) and application services in mobile ad hoc networks. Public Key Infrastructure (PKI) has been recognized as one of the most effective tools for providing security for dynamic networks. However, providing such an infrastructure in MANETs is a challenging task due to their infrastructureless nature. Hence, the PKI in ad hoc networks are mobile hosts nodes (or a set of them), then the key management system should not trust nor rely on any *fixed* Certificate Authority CA, but should be self organized.

6.1. Required features for a successful key management system

Distribution: Since there is no fixed infrastructure, the CAs should be distributed over mobile nodes. As we will see later, the choice of these nodes is problematic.

Fault Tolerance: The main concern of fault tolerance is the capability to maintain correct operations in the presence of faulty nodes. Replication using threshold cryptography can be employed to provide tolerance of faulty nodes.
Availability: Traditionally, the term availability has been used in conjunction with fault tolerance, but in ad hoc networks, availability is also highly dependent on the connectivity of the network. In wired networks, if there are no faulty or compromised nodes, the system is by definition available for clients since connectivity is not a problem. whereas, in ad hoc networks, even when there are no faulty or compromised nodes, clients may not be able to contact the desired services due to inconsistent connectivity.
Security: Acting as the trust anchor for the whole network, the CA should be secure against malicious nodes or adversaries. While it may not be possible to be resistant to all levels of attacks, there should be a clear threshold of attacks a system can withstand while operating normally.

6.2. Proposed solutions

Many propositions to secure routing protocol have taken place, most of them rely on authentication, and assume the existing of a central CA, as we have seen, the existing of such a CA in MANET is really problematic. Few studies have been devoted to key management, whose solutions can be used for securing both networking functions (e.g., routing) and application services. In this section we present two recently proposed solutions to key management in ad hoc networks.

6.2.1. A fully distributed solution

Capkun et al [38] propose a fully distributed self-organizing public-key management system in which the nodes generate their keys, issue, store, and distribute public-key certificates. This system is similar to PGP (Pretty Good Privacy) [39] in the sense that public-key certificates are issued by the nodes. However, in order to remove the reliance on on-line servers (which are clearly incompatible with the philosophy of ad hoc networks), the system does not rely on certificate directories for the distribution of the certificates. Instead, certificates are stored and distributed by the nodes and each node maintains a local certificate *repository* that contains a limited number of certificates selected by the node according to an appropriate algorithm. When node u wants to verify the *authenticity* of v's public key, the two nodes merge their local certificate repositories, then u tries to find an appropriate certificate *chain* from u to v in the merged repository.
An algorithm for the construction of local certificate repositories is proposed such that any pair of nodes can find certificate chains to each other in their merged repository. The basic operations of this public-key management scheme are:

Creation of public keys: The public key and the corresponding private key of each node is created locally by the node itself.
Issuing public-key certificates: If a node u believes that a given public key K_v belongs to a given node v, then u can issue a public-key certificate in which K_v is bound to v by the signature of u. There may be many reasons for u to believe that K_v belongs to v, for instance u may receive K_v on a secure (possibly out-of-band) channel that is associated with v, or someone trusted by u claims that K_v belongs to v, etc.
Storage of certificates: Certificates issued in the system are stored by the nodes in a fully decentralized way. Each node maintains a local certificate repository that has two parts: First, each node stores the certificates that it issued. Second, each node stores a set of additional certificates (issued by other nodes) selected according to an appropriate algorithm given in [38]. This additional set of certificates is obtained from other nodes, for this purpose, some underlying routing mechanisms are assumed to exist.
Key authentication: When a node u wants to obtain the authentic public key K_v of another node v, it asks other nodes (possibly v itself) for K_v. In order to verify the authenticity of the received key,

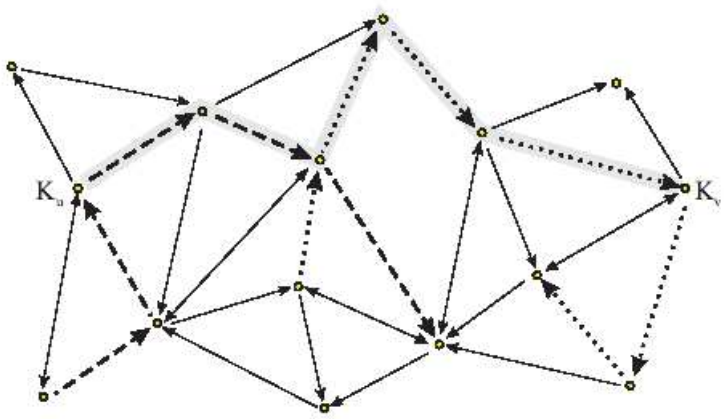

Figure 10: A path from $K_u to K_v$in the merged subgraph of u and v

v or the node which supplied the key to u also provides u with it local certificate repository, then u merges the received repository with its own repository and tries to find an appropriate certificate chain from K_u to K_v in the merged repository. If this fails, u may ask other nodes for further certificates.

Model: The public keys and the certificates of the system are represented as a directed graph G(V,E), where V and E stand for the set of vertices and the set of edges respectively, this graph is called the certificate graph. The vertices of the certificate graph represent public keys and the edges represent certificates. More precisely, there is a directed edge from vertex K_u to vertex K_w if there is a certificate signed with the private key belonging to u in which K_w is bound to an identity. A certificate chain from a public key K_u to another public key K_v is represented by a directed path from vertex K_u to vertex K_v in G. For any directed graph H, if two vertices x and y are in H, and there is a directed path from x to y in H, then we say that y is reachable from x in H. Thus, the existence of a certificate chain from K_u to K_v means that vertex K_v is reachable from vertex K_u in G.
As we have said, when user u wants to verify the authenticity of the public key K_v of user v, u and v merge their certificate repositories, and u tries to find an appropriate certificate chain from K_u to K_v in the merged repository. In the model, u and v merge their subgraphs and u tries to find a path from vertex K_u to vertex K_v in the merged subgraphs, an example is shown in figure 10.
In terms of this model, constructing a local certificate repository means selecting a subgraph of the full certificate graph of the system. In [38], a description of a proposed algorithm is available.
However, we should note that this approach provides only probabilistic guarantees.

6.2.2. A partial distributed solution

In [40], Yi and Kravets employ threshold cryptography to distribute the CA functionality over specially *selected* nodes based on the security and the physical characteristics of nodes. The selected nodes that collectively provide PKI functionality are called MOCAs (MObile Certificate Authorities). Using these MOCAs, an efficient and effective communication protocol for correspondence with them for certification services is presented.

Mobile nodes may be heterogeneous in many respects, especially in terms of their security, in this case, any security service or framework should utilize this environmental information. For instance, consider a battlefield scenario with a military unit consisting of nodes soldiers that have different

ranks, hence they may be equipped with computers that are different in power, capabilities, transmission ranges, levels of physical security, and so on. In such a case, Yi and Kravets propose to pick such nodes to provide any security service to the rest of the network, and in general, to exploit knowledge of heterogeneity to determine the nodes that will share the responsibility of the CA.

The communication pattern between a client and k or more MOCA servers is one-to-many-to-one (*manycast*), which means that a client needs to contact at least k MOCAs and receive at least k replies. To provide an effective and efficient way of achieving this goal, MP (MOCA certification Protocol) is proposed. In MP, a client that requires certification services sends Certification Request (CREQ) packets, any MOCA that receives a CREQ responds with a Certification Reply (CREP) packet containing its partial signature. The client waits a fixed period of time for k such CREPs. When the client collects k valid CREPs, it can reconstruct the full signature and the certification request succeeds. If too few CREPs are received, the clients CREQ timer expires and the certification request fails. On failure, the client can retry or proceed without the certification service. The CREQ and CREP messages are similar to Route Request (RREQ) and Route Reply (RREP) messages in ondemand ad hoc routing protocols.

The shape of a MOCA framework is determined by the total number of nodes in the network, the number of MOCAs, and the threshold value for secret reconstruction(number of MOCAs required to issue certificates). Although the total number of nodes in the network (M) can change dynamically over time, it is not a tunable parameter. The number of MOCAs (n) is determined by the characteristics of nodes in the network, such as physical security or processing capability and it is also not tunable. In this system, n defines the limits of the system as an upper bound for k; the minimum number of MOCAs a client must contact to get certification services. Given M and n, the last parameter k which is the threshold for secret recovery, is indeed a tunable parameter. Once k has been chosen and the system is deployed, it is expensive to change k. Therefore, it is important to understand the effects of varying k on a given system, k can be chosen between 1 (a single CA for the whole network) and n (a client needs to contact all MOCAs in the system to get certification services). Setting k to a higher value has the effect of making the system more secure against possible adversaries since k is the number of MOCAs an adversary needs to compromise to collapse the system. But at the same time, a higher k value can cause more communication overhead for clients since any client needs to contact at least k MOCAs to get certification services. Therefore, the threshold k should be chosen to balance the two conflicting requirements, it is clear that no one value will fit all systems.

It is possible that an ad hoc network does not have enough heterogeneity among the nodes, which may make it difficult if not impossible to choose MOCAs based on this heterogeneity assumption. In such cases, the solution proposed is to choose randomly a subset of nodes as MOCAs. We think this is not an efficient strategy, if a subset is chosen it should fulfil a given criterium, otherwise why do not distribute the task over all nodes. Moreover, we think that the choice of a *static* subset of nodes as MOCAs is not optimal, because conditions change along the time, and nodes that are not MOCAs in a given time will be able to be more suitable to act as MOCAs, hence the MOCAs set should be dynamic.

7. Intrusion Detection Systems IDSs

An intrusion may be defined as: "any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource" [41].
Intrusion prevention measures, proactive solutions, can be used in ad hoc networks to reduce intrusions, but they cannot eliminate them. For example, encryption and authentication cannot defend against compromised mobile nodes, which carry the private keys. Integrity validation using redundant

information from different nodes, such as those being used in secure routing, also relies on the trustworthiness of other nodes, which could likewise be a weak link for sophisticated attacks. The history of security research has taught us a valuable lesson, no matter how many intrusion prevention measures are inserted in a network, there are always some weaknesses in the systems that one could exploit to break in, these weaknesses are design and programming errors, or various social engineering penetration techniques (as illustrated in "I love you" virus). Hence, IDS presents a second wall of defense and it is a necessity in any high-survivability network.
The primary assumptions of intrusion detection are:

- User and program activities are observable, for example via system auditing mechanisms; and more importantly,
- Normal and intrusion activities have distinct behavior.

Therefore, intrusion detection involves capturing *audit data* and reasoning about the evidence in the data to determine whether the system is under attack. Based on the type of audit data used, traditional IDSs can be categorized as:

- Network-based IDS: normally it runs at the gateway of a network where it captures and examines network packets that go through the network hardware interface.
- Host-based IDS: Relies on operating system audit data to monitor and analyze the events generated by programs or users on the host.

Another classification of IDSs is based on the techniques used which are:

- *Misuse detection systems*, as IDIOT [42] and STAT [43]. They use patterns of well-known attacks or weak spots of the system to match and identify known intrusions. For example, a rule for the "guessing password attack" can be "there are more than 4 failed login attempts within 2 minutes". The main advantage of this technique is that it can accurately and efficiently detect instances of known attacks. Whereas, its main drawback is that it lacks the ability to detect the truly innovative (i.e., newly invented) attacks, whose patterns are unknown.
- Anomaly detection systems, like IDES [44], they flag observed activities that deviate significantly from the established normal usage profiles as anomalies, i.e. possible intrusions. For instance, the normal profile of a user may contain the averaged frequencies of some system commands used in his or her login sessions. If for a session that is being monitored, the frequencies are significantly lower or higher, then an anomaly alarm will be raised. The main advantage of anomaly detection is that it does not require prior knowledge of intrusions and can thus detect new intrusions. The main disadvantage is that it may not be able to describe what the attack is and may have high false positive rate, i.e detects normal operations as attacks.

Conceptually, an intrusion detection model, i.e., has these two components:

- *The features* (attributes or measures), e.g. the number of failed login attempts, and the averaged frequency of the gcc command that together describe a logical event, a user login session, etc.
- *The modelling algorithm*, it is a rule-based pattern matching that uses the features to identify intrusions.

Defining a set of predictive features that accurately capture the representative behaviors of intrusive or normal activities is the most important step in building an effective intrusion detection model, and it can be independent of the design of the modelling algorithm. These features should be established such that The main purpose of an IDS model is reached, which can be summarized as; decreasing low false positive rate, calculated as the percentage of normalcy variations detected as anomalies or intrusions, and increasing true positive rate, calculated as the percentage of anomalies or intrusions detected [45].

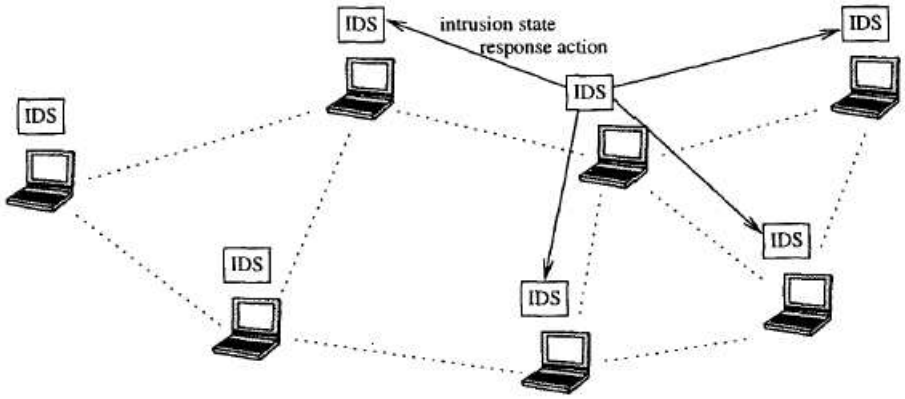

Figure 11: IDS architecture for MANET [45]

7.1. Traditional IDS problems

The vast difference between traditional networks and MANETs makes it very difficult to apply intrusion detection techniques developed for the former to the later. The most important difference is that MANETs does not have a fixed infrastructure, and today's network- based IDSs which rely on real-time traffic analysis can no longer function well in the new environment. Compared with wired networks where traffic monitoring is usually done at switches, routers and gateways, *MANETs do not have such traffic concentration points* where the IDS can collect audit data for the entire network.

The second big difference is in the communication pattern. In MANETs wireless users tend to be stingy about communication due to slower links, limited bandwidth, higher cost, and battery power constraints. Disconnected operations [46] are very common in wireless network applications, and so is location-dependent computing or other techniques that are solely designed for wireless networks and seldom used in the wired environment. All these suggest that the anomaly models for wired networks cannot be directly used in this new environment.

Furthermore, an other big problem with MANETs is that there may not be a clear separation between normalcy and anomaly. For instance, A node that sends out false routing information could be the one that has been compromised, or solely the one that is temporarily out of sync due to volatile physical movement. Intrusion detection may find it increasingly difficult to distinguish false alarms from real intrusions.

7.2. A New architecture

We must answer the following questions in developing a viable IDS for MANETs:

- What is a good system architecture for building IDSs that fits the MANETs' features?
- What are the appropriate audit data sources?
- How do we detect anomaly based on partial and local audit traces if they are the only reliable audit source?
- What is a good model of activities in a wireless communication environment that can separate anomaly when under attacks from the normalcy?

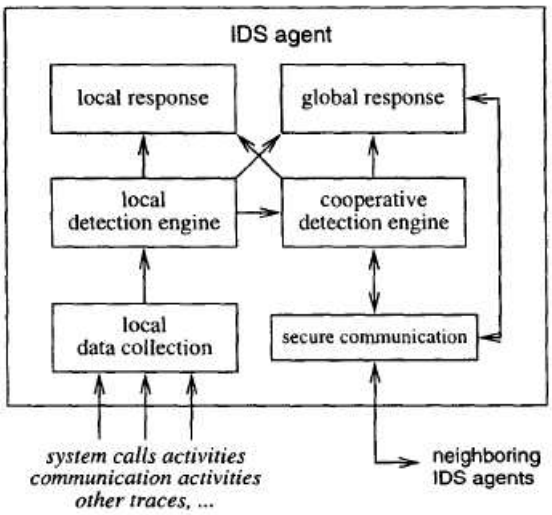

Figure 12: A conceptual model for an IDS agent [45]

IDSs should be both distributed and cooperative to suite the needs of wireless ad hoc networks. In [45] Zhang and Lee propose a novel architecture (figure 11) that can be considered as a general framework to build MANETs' IDSs. Every node in the MANET participates in intrusion detection and response. Each node is responsible for detecting signs of intrusion locally and independently, but neighboring nodes can collaboratively investigate in a broader range.
In the systems aspect, individual IDS agents are placed on each and every node. Each IDS agent runs independently and monitors local activities, including user and systems activities, and communication activities within the radio range. It detects intrusion from local traces and initiates response. If anomaly is detected in the local data, or if the evidence is inconclusive and a broader search is warranted, neighboring IDS agents will cooperatively participate in global intrusion detection actions. These individual IDS agent collectively form the IDS system to defend the MANET.
The internal of an IDS agent can be fairly complex, Zhang and Lee [45] conceptually structure it into six pieces (Figure 12).

- The data collection module: It is responsible for gathering local audit traces and activity logs.
- The local detection engine: It uses data collected by The data collection module to detect local anomaly.
- Cooperative detection engine: It is used by the detection methods that need broader data sets or that require collaborations among IDS agents
- The local response module: It triggers actions local to the mobile node.
- The global response module: It coordinates actions among neighboring nodes, such as the IDS agents in the network electing a remedy action.
- The secure communication module: It provides a high-confidence communication channel among IDS agents.

8. Conclusions

In this paper we have studied the different MANET's security issues, and have shown that the features of this new environment make it more vulnerable to threats and that the solutions developed for standard networks are often unsuitable in this environment. We have divided threats into two categories; attacks and misbehavior, then we have presented how they can affect the MANET's security in different layers, especially in the network and the Medium Access Control (MAC) layers. For the network layer we have presented different kinds of attacks on routing protocol and we have classified and discussed the proposed solutions. We have also presented attacks and misbehavior on data forwarding which have received relatively less attention in literature, we think securing data forwarding is a fertile field of research. Regarding the medium access layer, we have presented the non-respect on the channel access misbehavior that can affect hugely the network efficiency, and we have presented and discussed the lonely solution proposed. In our discussion we have shown how this solution may accuse wrongly a well-behaving node, and how it is unable to detect what we have called cooperative misbehavior. We have also presented the key distribution issue that can be an underlying mechanism for securing both lower and upper layers, and finally Intrusion Detection Systems (IDSs) that are essential when preventive measures fail have been presented. We think securing ad hoc networks is a great challenge that includes many opened problems of research, and receives more and more attention among ad hoc networks community.

Acknowledgment

We are grateful to Prof Abdelmadjid Bouabdallah for his advises, support, and reception at the Heudiasyc laboratory of the university of compiegne, where we have prepared this survey. Many thanks are also due to our colleagues Abdelouahid Derhab and Souad Benmeziane for their reviews and helpful comments.

References

[1] William Stallings. *Cryptography and Network Security principles and practices*. Pearson Education Inc, third edition edition, 2003.

[2] Frank Stajano and Ross Anderson. The resurrecting duckling: Security issues for ad-hoc wireless networks. In *7th International Security Protocols Workshop, Cambridge, UK*, April 1999.

[3] Dan Nguyen, Li Zhao, Pra ornsiri Uisawang, and John Platt. Security routing analysis for mobile ad hoc networks. *Technical report, University of Colorado, Boulder*, 2003.

[4] B.Johnson David and A.Maltz David. Dynamic source routing in ad hoc wireless networks. *Mobile Computing, Chapter 5*, pages 153–181, 1996.

[5] Y.-B. Ko and N.H. Vaidya. Location-aided routing, LAR, in mobile ad hoc networks. In *ACM/IEEE MOBICOM'98, Dallas, Texas*, pages 66–75, October 1998.

[6] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *INFOCOM '97*, pages 1405–1413, Aprile 1997.

[7] Charles.E. Perkins and Elizabeth.M. Royer. Ad hoc on demand distance vector (AODV) algorithm. In *the 2nd IEEE Workshop on Mobile Computing Systems and Applications (WMCSA'99)*, pages 90–100, 1999.

[8] Chai-Keong Toh. A novel distributed routing protocol to support ad hoc mobile computing. In *IEEE 15th Annual International Phoenix Conference on Computers and Communications*.

[9] Charles.E. Perkins and Pravin Bhagwat. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computer. In *ACM SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*, pages 234–244, March 1994.

[10] S.Murthy and J.J.Garcia-Luna-Aceves. An efficient routing protocol for wireless networks. In *CM Mobile Networks and Application Journal, Special Issue on routing in mobile Communication Networks*, pages 183–197, October 1996.

[11] N.Badache, D.Djenouri, A.Derhab, and T.Lemlouma. Les protocoles de routage dans les rseaux mobiles ad hoc. *RIST Revue d'Information Scientifique et Technique, Volume 12 No 2*, pages 77–112, 2002.

[12] Nadjib Badache, Djamel Djenouri, and Abdelouahid Derhab. Mobility impact on mobile ad hoc networks. In *ACS/IEEE conference proceeding, Tunis, Tunisia*, July 2003.

[13] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A secure on-demand routing protocol for ad hoc networks. In *In Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking MobiCom 2002*, pages 12–23, september 2002.

[14] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. Secure efficient distance vector routing in mobilewireless ad hoc networks. In *Fourth IEEEWorkshop on Mobile Computing Systems and Applications WMCSA 02*, June 2002.

[15] Claude Castelluccia and Gabriel Montenegro. Protecting AODV against impersonation attacks. In *ACM SIGMOBILE Mobile Computing and Communications Review archive Volume 6, Issue 3*, pages 108–109, July 2002.

[16] Panagiotis Papadimitratos and Zygmunt J. Haas. Secure routing for mobile ad hoc networks. In *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, January 2002.

[17] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, and J. D. Tygar. Spins: Security protocols for sensor networks. In *Seventh Annual ACM International Conference on Mobile Computing and Networks (MobiCom 2001). Rome, Italy*, July 2001.

[18] Kimaya Sanzgiri, Bridget Dahill, Brian Neil Levine, Clay Shields, and Elizabeth Belding-Royer. A secure routing protocol for ad hoc networks. In *Proceedings of the 10th IEEE International Conference on Network Protocols (ICNP 02)*, November 2002.

[19] Manel Guerrero Zapata and N. Asokan. Securing ad hoc routing protocols. In *Proceedings of the ACM Workshop on Wireless Security (WiSe 2002)*, September 2002.

[20] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. In *Proceeding of the ACM workshop on WIreless SEcurity WISE 2003, San diego, CA, USA*, september 2003.

[21] Seung Yi, Prasad Naldurg, and Robin Kravets. Security-aware ad-hoc routing for wireless networks. In *ACM Workshop on Mobile ad hoc networks, Mobihoc*, 2001.

[22] F. Wang, B. Vetter, , and S. Wu. Secure routing protocols: Theory and practice. In *Technical report, North Carolina State University*, May 1997.

[23] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Packet leashes: A defense against wormhole attacks in wireless ad hoc networks. In *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, April 2003.

[24] Djamel DJENOURI and Nadjib BADACHE. An energy efficient routing protocol for mobile ad hoc network. In *The second proceeding of the Mediterranean Workshop on Ad-Hoc Networks, Med-Hoc-Nets 2003, Mahdia, Tunisia*, pages 113–122, 25-27 June 2003.

[25] S. Marti, T. Giuli, K. Lai, and M. Baker. Mitigating routing misbehavior in mobile ad hoc networks. In *ACM Mobile Computing and Networking, MOBICOM 2000*, pages 255–65, 2000.

[26] Panagiotis Papadimitratos and Zygmunt J. Haas. Secure data transmission in mobile ad hoc networks. In *the 2003 ACM workshop on Wireless security, San Diego, CA, USA, session: Secure routing*, pages 41–50, 2003.

[27] X. Meng H. Yang and S. Lu. Self-organized network layer security in mobile ad hoc networks. In *ACM MOBICOM Wireless Security Workshop (WiSe'02)*, September 2002.

[28] M.O. Rabin. Efficient dispersal of information for security, load balancing, and fault tolerance. *Journal of ACM, Vol.36, No.2*, pages 335–348, April 1989.

[29] S. Bouam and J. Ben Othman. Securing data protocol using multipath routing in ad hoc networks. In *The second proceeding of the Mediterranean Workshop on Ad-Hoc Networks, Med-Hoc-Nets 2003, Mahdia, Tunisia*, Jun 2003.

[30] P. Zerfos J. Kong, H. Luo, S. Lu, and L. Zhang. Providing robust and ubiquitous security support for manet. In *The 9th IEEE International Conference on Network Protocols ICNP 2001 Mission Inn, Riverside CA, USA*, pages 251–260, November 2001.

[31] A. Shamir. How to share a secret. *Communications of the ACM, Vol.22, No.11*, pages 612–613, November 1979.

[32] Matthew S.Gast. *802.11 Wireless Networks*. O'Reilly and Assiciation, Inc, first edition, isbn 0-596-00183-5 edition, 2002.

[33] Pradeep Kyasanur and Nitin H. Vaidya. Detection and handling of mac layer misbehavior in wireless networks. In *IEEE International Conference on Dependable Systems and Networks (DSN'03), San Francisco, California*, Jun 2003.

[34] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J.P. Hubaux, and J. Y. Le Boudec. Self-organization in mobile ad-hoc networks: the approach of terminodes. *IEEE Communications Magazine, Vol.39, No.6*, June 2001.

[35] L. Buttyn and J.-P. Hubaux. Nuglets: a virtual currency to stimulate cooperation in selforganized mobile ad hoc networks. *Technical report No. DSC/2001/001, Swiss Federal Institution of Technology, Lausanne*, January 2001.

[36] L. Buttyn and J.-P. Hubaux. Stimulating cooperation in self-organizing mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications (MONET) Special Issue on Mobile Ad Hoc Networks, Vol. 8 No. 5*, October 2003.

[37] V. Srinivasan, P. Nuggehalli, C. F. Chiasserini, and R. R. Rao. Energy efficiency of ad hoc wireless networks with selfish users. In *EW2002 (European Wireless conference), Florence, Italy*, 26-28 February 2002.

[38] Srdjan Capkun, Levente Buttyan, and Jean-Pierre Hubaux. Self-organized public-key management for mobile ad hoc networks. *IEEE Transactions on Mobile Computing, Vol.2, No.1*, pages 52–64, January 2003.

[39] P. Zimmermann. *The Official PGP Users Guide*. MIT Press, 1995.

[40] Seung Yi and Robin Kravetso. Moca : Mobile certificate authority for wireless ad hoc networks. In *The second anunual PKI research workshop (PKI 03), Gaithersburg*, 2003.

[41] R.Heady, G.Luger, A.Maccabe, and M.Servilla. The architecture of a network level intrusion detection system. In *Technical report, Computer Science Department, University of New Mexico*, August 1990.

[42] K. Ilgun, R. A. Kemmerer, and P. A. Porras. State transition analysis: A rule-based intrusion detection approach. *IEEE Transactions on Software Engineering 21(3)*, pages 181–199, March 1995.

[43] S.Kumar and E. H.Spafford. A software architecture to support misuse intrusion detection. In *the 18th National Information Security Conference*, pages 194–204, 1995.

[44] T. Lunt, A. Tamaru, F. Gilham R. Jagannathan, P. Neumann, H. Javitz, A. Valdes, and T. Garvey. A real-time intrusion detection expert system (ides) final technical report. *Technical report, Computer Science Laboratory, SRI International, Menlo Park, California*, February 1992.

[45] Y. Zhang and W. Lee. Intrusion detection in wireless adhoc networks. In *Mobile Computing and Networking, MOBICOM 2000, Boston, MA, USA*, pages 275–283, 2000.

[46] M.Satyanarayanan, J.J. Kistler, L.B.Mummert, P.Kumar M.R.Ebling, and Q.Lu. Experiences with disconnected operation in a mobile environment. In *USENIX Symposium on Mobile and Location Independant Computing, Cambridge, Massachusetts*, pages 11–28, August 1993.