On the Limitations of Security Concepts for Mobile Adhoc Networks Routing Protocols

A. Festag, J. Girao, T. Melia und D. Westhoff

NEC Network Laboratories Heidelberg, NEC Europe Ltd.
melia@netlab.nec.de

Abstract: Recently, a number of approaches proposing various security aspects for routing protocols in mobile ad hoc networks have been proposed. In this work we do not introduce another proposal aiming at security in the context of ad hoc networks. Instead, we argue from a generalized view and answer the question: what security objectives are in principle achievable and what security features are not achievable at all, even when using unacceptable heavy weighted security features. Our contribution in this work is to derive from the available security mechanisms and from a reasonable ad hoc network specific attacker model a set of theoretically achievable security objectives. In particular we will also denote the majority of non practically achievable security objectives.

1 Introduction

Before starting to design any security solution we need to understand (1) against whom we want to protect the system and (2) what should be the protection aims of the system itself. (1) relates to the assumed attacker model. The attacker model describes in an either formal or informal way the attacker's capabilities taking into account resources, power computation and/or colluding behavior. Ideally an attacker is equipped with considerable computation resources, but in a more relaxed scenario an attacker will probably deploy a laptop or PDA. Also, we have to face scenarios in which a set of attackers (colluding/distributed attacks) try to disrupt the network. Once we identify the attacker the system may not tolerate, we have to fix the requirements to provide different levels of security (2). Routing protocols, in general, should aim at:

- routing signaling cannot be spoofed,
- forged routing messages cannot be injected into the network,
- routing messages cannot be altered except normal protocol functionalities,
- routing loops cannot be formed via malicious behaviors,
- routes cannot be redirected from the shortest path, and, as a prevention principle, non authorized nodes should be excluded from the route computation.

Existing ad hoc routing protocols (e.g. [PERD03], [CP03]) have been designed under the assumption of a trusted and cooperative environment. As a matter of fact, nodes may misbehave driven by different reasons. Selfishness,for instance, attempts to break the philosophy of the cooperative environment by means of saving resources, i.e. forwarding is an expensive operation in terms of battery lifetime, or not sharing available resources within

the network itself. Malicious behaviors can derive as well from attackers looking for sudden feelings of great excitement; apparently this is another considerable motivation. The idea to prevent users to fetch data from the Internet, to use the available services or, even worse, to inject into the network forged and false information, pushes malicious actors to be more and more effective and precise in detecting victims and their flaws. Furthermore, since ad hoc routing protocols could carry confidential information in the payload of both data messages and routing messages, a misbehaving node may collect confidential information about other nodes such as position, velocity and movement traces. A combination of such data could give room to creation of users profiles and habits. Ad hoc networks are particularly vulnerable to all such kind of attacks since they are open by nature.

Moreover, spoofing combined with impersonation allows a misbehaving node to assume somebody else credentials and therefore more than one hop far away nodes would believe to be direct neighbors. The effects of such actions could lead simply to DoS attacks or could compromise the functionalities of the routing algorithm itself. Fabrication and forging of datagrams aim at injecting into the network new or partially new routing control traffic. These features could be exploited to produce loops in the routing process, to make a path more attractive and shorter than what is in reality or to exclude nodes from any communication path. As we can see the more we compose different actions the more the attack become efficient and difficult to be detected.

In the following Section we present the categories of considered attacks. In Section 3 we present the protocols listed in table 1 and their major flaws and in Section 4 we discuss achievable and not achievable countermeasures. Finally, we conclude our work and identify future steps.

2 General Attack Categories

We now look at the attacks that are currently known and try to categorize them in a way that is coherent with out analysis on the routing protocols.

We begin by providing a list of well known generic attacks and contextualize them in the ad hoc scenario. We then take these concepts to a higher level and search for ways how to provide a common base for analyzing distinct routing protocols.

2.1 Generic Attacks

In this section we try to enumerate different attacks and their impact on the ad hoc wireless network.

Firstly we look at passive sniffing/eavesdropping and IP/MAC spoofing.Sniffing can be seen as being an even more adequate problem in a wireless shared medium. A malicious node can try and obtain information from the payload of the packets, communication patterns, involved parties or any other information it can listen to. The second problem mentioned occurs when a node uses an IP or layer 2 address that is not his own. This way he may impersonate or gain access to services he would otherwise not have. As to ad hoc specific attacks we face those that derive from the flexibility requirements of the

routing protocols, the restricted resource capacity of the nodes and the medium itself. Routing based attacks fall into many other categories such as *sinkhole* attacks (also known as *blackhole*) in which nodes do not forward packets they receive, when a node only forward some packets (this is known as the Grey hole or selective forwarding attack), *wormhole* attacks in which the nodes manipulate routes in which they are involved or even hybrid attacks that involve IP spoofing (as such is the *Sybil* attack). The restricted capacity of the nodes also requires attention as some attacks can be motivated by the saving of resources. Such attacks interfere with the functioning of the network when a node refuses to forward packets or do computation for other nodes. Nevertheless, they can be prevented by Intrusion Detection Systems, based on watchdogs that determine which nodes are not cooperating, or by incentive based systems that lead to interaction between all nodes by providing mutual benefit. These need to be shaped to the network and, even so, only a subset of attacks can be detected. All of these contribute to the problem of labeling a node as malicious. The last set of attacks are in relation to Denial of Service (DoS) that either prevent the functioning of the network by flooding or take advantage of defects in the underlying protocols, or simply do not allow the node to conserve energy and force it to do unnecessary computation (sleep deprivation attacks).

2.2 Attack Categories specific to routing protocols

We now divide the specific attacks on the ad hoc routing protocols we intend to analyze into three categories:

- **Route Manipulation**
  We first look at attacks that directly affect the route established between two nodes that wish to communicate. These attacks normally take place in the route discovery phase of the protocol. They can be divided into two groups:
  – *Route Diverting:* Route is extended to a non-optimal path by either insertion or substitution of a node in the optimal route.
  – *Route Subtracting:* The route is subtracted of an element (possibly the forwarding node itself) not to contemplate some nodes in the final path.
- **Impersonation**
  In this type of attacks we consider the attacker tries to occupy a position in the protocol that does not belong to it. We contemplate the case in which the impersonator is pretending to be the source, the destination or any of the forwarding nodes.
- **Injection of Signaling Messages**
  In this case the attacker sends messages outside of the protocol definition to. We distinguish between *fabrication* and *replay*.
  – *Fabrication:* The packet being injected in the network is created by the attacker.
  – *Replay:* The originator of the packet is not the attacker. The attacker simply sends a copy of the packet at a different instance of time.

3 Protocol Description and Attacks

The two most commonly used classifications for ad hoc routing protocols reactive or proactive routing protocols. A reactive protocol functions on an event driven model. When

a node needs to communicate with another node and does not possess a route, a request message is sent and the network reacts. Due to the nature of the protocol, nodes have only partial information on the network. In a proactive routing protocol routes are constantly updated and all the topology of the network is known. Because the routes are known *a priori*, no messages other than the data packets are exchanged in the event of initiating communication.

Reactive	Proactive
DSR	OLSR
AODV	
GeoRouting	

Table 1: Protocols analyzed more thoroughly in this paper.

We have chosen to look at four of the most well known routing protocols for ad hoc networks.. The first is *Dynamic Source Routing* (DSR), a reactive protocol based on source based routing.The second is *Ad Hoc On-demand Distant Vector* (AODV). While it is still a reactive protocol, it presents other mechanisms that are common to this category. As an proactive routing protocol we have chosen Optimized Link-State Routing (OLSR) as our candidate due to its popularity in the research community and employment of proactive routing mechanisms. Finally, we take the example of Geo-routing as a special case of a reactive protocol based on location services. We briefly describe the protocols as far as it is relevant to describe the attacks that are specific to the protocols.

3.1 DSR

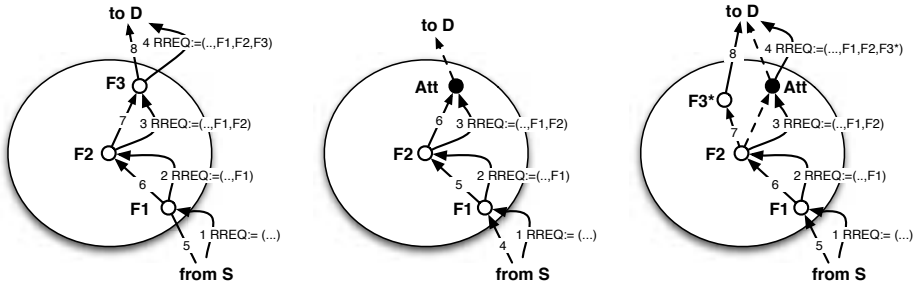
In DSR every packet has a route path consisting of the addresses of nodes that have agreed to participate in routing the packet. DSR can be divided into two main functions: Route discovery and route maintenance. In the discovery phase (see Fig. 1(a)) a source S wishing to communicate with destination$D$, but not having any path to D, initiates a route discovery bybroadcasting a route discovery message $RREQ := (S, D)$ to its neighborthat contains the destination address D. The neighbors N_i,$i = 1, \ldots, n$ in turn receive$IN(N_i) : RREQ := (S, D||N_1, \ldots, N_{i-1})$ append their own addressesto the route discovery packet, and re-broadcast$OUT(N_i) : RREQ := (S, D||N_1, \ldots, N_i)$. The process continues until aroute request packet reaches destination D. D now sends back a routereply $RREP := (D, S||N_1, \ldots, N_n)$ packet to inform S of thediscovered route. Since the route request packet that reaches Dcontains a path from S to D, D may choose to use the reversepath to send back the reply (bi-directional link is required here) orto initiate a new discovery back to S. Since there can be manyroutes from a source to a destination, a source may receive multiple route replies from a destination. DSR caches these routes in the route cache for future use. The second main function in DSR is route maintenance, which handles path breaks. A path break occurs when two nodes on a path are no longer in transmission range. If N_i detects a link break when forwarding a packet to the next node in the route path, it sends back a message to the source notifying it of the

link break. The source must try another path or do a route discovery if it does not have another path.

DSR-specific attacks aim at the route discovery. A malicious node may manipulate the route by attacking the source routing mechanism of DSR. In particular, an attacker may

- not forwarding a route discovery message so that the attacker never belongs to the path for data forwarding with $OUT(Att) : -$ (see Fig. 1(b)) and D may not receive any packet
- forwarding a route discovery message but not including its own identity with[1] $OUT(Att) : RREQ := (S, D||N_1, \ldots, N_{i-1}, *)$
- manipulating the prefix of the path (route diverting) $OUT(Att) : RREQ := (S, D||N_1^*, \ldots, N_{i-1}^*)$
- adding some unnecessary node information at the end of the current node list $OUT(Att) : RREQ := (S, D||N_1, \ldots, N_i, N_1^*, \ldots, N_l^*, N_i)$,
- including another forwarding node into the RREQ in order to selfishly re-route packets via this node (see Fig. 1(c)) (also impersonation) $OUT(Att) : RREQ := (S, D||N_1, \ldots, N_{i-1}, X_*)$

Similar attacks can be applied to RREP packets. This can be regarded as more clever attacks since RREP messages are sent by unicast.



(a) Regular forwarding process via forwarding nodes F1, F2, and F3.

(b) Attacker Att simply drops the RREQ and is not included in path

(c) Att impersonates another node, includes this node in the RREQ, and is excluded from forwarding.

Figure 1: Two attacks on route discovery in DSR

[1] The * represents falsified fields in a message.

3.2 AODV

AODV is a reactive routing protocol where nodes have only a partial knowledge of the network topology. Similar to DSR, AODV has the two basic functions, route discovery and route maintenance. Unlike DSR based on source routing, in AODV routing table entries are created in each node along the discovered route and data packets are forwarded based on these routing table entries. For route discovery, an expanding ring search mechanism is used that makes use of the TTL field in the IP header. If a node S needs to discover a route to another node D, it issues $OUT(S) : RREQ := (TTL, S, D, SN_S, SN_D, HC, ID)$, where TTL is the time to live and defines the radius of the ring mechanism, SN define sequence numbers for S and D, the hop count is set in HC and initialized at 0 and the ID, together with S provides identification for the message (see Fig. 2(a)). At each hop, TTL is incremented andHC decremented. A route reply message RREP is then sent by unicasteither from the destination of the RREQ or an intermediate node that contains a cached route to the destination. When a node receives a RREQ and has a route to D or is in fact D, it creates a $OUT(D) : RREP := (HC, D, S, L, SN_D)$. HC is now set with the distance in hops from S to D retrieved from the RREQ and L refers to the lifetime of the route in question. If a node detects a link break that invalidates a route, it sends a $OUT(N_i) : RERR := (D, SN_D)$ to destroy the route in question. More than one destination D may be present in this message. If an error in the link occurs, a route error message RERR can be used to destroy a previously discovered route (route maintenance).

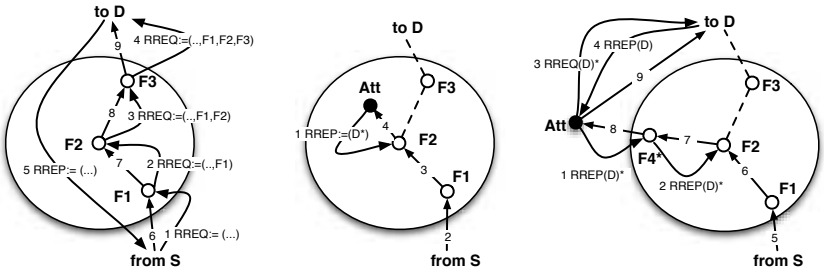
Attacks on the route discovery are specific to AODV, where an attacker Att may manipulate the route as follows:

- Reply to a RREQ for which the attacker does not have a route to the destination (Fig. 2(b))
  $IN(Att) : RREQ := (TTL, S, D, SN_S, SN_D, HC, ID)$
  $OUT(Att) : -$
- Reply to a RREQ for which the attacker does not have a route to the destination, discover a new route and proceed with forwarding operations (now the attacker is in the data path) (Fig. 2(c))
  $IN(Att) : RREQ := (TTL, S, D, SN_S, SN_D, HC, ID)$
  $OUT(Att) : RREP* := (HC, D, N_i, L, SN_D)$
  $OUT(Att) : RREQ* := (TTL, N_i, D, SN_S, SN_D, HC, ID)$
- Provide always fresher SN or smaller HP in RREQ messages so that route is chosen over all others
  $IN(Att) : RREQ := (TTL, S, D, SN_S, SN_D, HC, ID)$
  $OUT(Att) : RREP := (HC^*, D, N_i, L, SN_D^*)$ where $HC^* < HC$ or $SN_D^* << SN_D$

3.3 Georouting

Geographical routing or position-based routing[2] is an on demand routing protocol that is based on information about the physical position of the participating nodes. For the forwarding decision at each node the destination's position contained in the packet and the

[2] We use both terms simultaneously.



(a) Regular forwarding process via forwarding nodes F1, F2, and F3.

(b) Attacker Att pretends having a route to D such that F2 forwards packets to the attacker.

(c) In addition to the attack in b), Att discovers the route to D and starts forwarding.

Figure 2: Two exemplary attacks on route discovery in AODV

position of the forwarding node's neighbor is used. Geographical routing thus does not require the setup and maintenance of routes. Geo-routing can be divided into four main functions: beaconing, location discovery, forwarding, and recovery. With beaconing a node N_i frequently broadcasts beacons to its one-hop neighbors $OUT(N_i) : B := (N_i, L_{N_i})$ with its address N_i and its current location L_{N_i} (see Fig. 3(a)). A source Swishing to communicate with destination D, but not having thecurrent location of the node in its location cache, initiates alocation discovery by broadcasting a location request message $LREQ := (S, D, L_S, T_{SL})$ to its neighbors that contains the source and destination addresses S and D, as well as the source's location L_S and source location time-stamp TS_{SL} as piggybacked location information. The neighbor N_i receives the message $IN(N_i) : LREQ := (S, D, L_S, L_{N_{i-1}}, TS_{SL})$ that also contains the current location of the sender $L_{N_{i-1}}$, and re-broadcasts $OUT(N_i) : LREQ := (S, D, L_S, L_{N_i}, TS_{SL})$. When the location query reaches the destination D, D issues a location reply which is forwarded by N_i as $OUT(N_i) : LREP := (S, D, L_D, L_S, L_{N_i}, TS_{SL}, TS_{DL})$. Both, the request and the reply message piggyback the location of the source and sender that intermediate nodes may learn the current location of the nodes. This also pertains to data messages, where an intermediate node receives a data message $IN(N_i) : Data := (S, D, L_S, L_D, L_{N_{i-1}}, TS_{SL})$ and forwards $OUT(N_i) : Data := (S, D, L_S, L_D, L_{N_i}, TS_{SL})$. The final function is recovery, which handles the forwarding of data messages if greedy routing fails.[3] Exist-

[3] This occurs in topologies in which the only route to a destination requires to send a packet temporarily farther away in geometric distance from the destination.

ing approaches are based on distributed graph planarization, for example, GPSR [KK00] provides a perimeter mode in which a planar graph of the topology is generated.[4]

Attacks for route manipulation specific to geo-routing are based on location spoofing, where an attacker node pretends to be at a different location than in reality. An attacker might falsify the location in signaling and data messages. It is noted that intermediate nodes "learn" the wrong route and therefore, subsequent data packets originated or forwarded by these nodes are steered to the wrong location as well. Furthermore, once a node has cached a location falsified by an attacker, it may wrongly update the contained location in other messages, not knowing that the location is wrong. In order to pretend that a falsified location is newer than existing ones, an attacker may even use a wrong times-tamp (which eventually lies in the future). Even so, wrong location information may result in fail of topology planarization and *faces* are wrongly calculated such that routes around faces do not exists or routing loops occur.

Attack on beaconing. An attacker Att may falsify its own location $OUT(Att) : B := (Att, L_{Att})$ and is selected as a forwarder instead of another node (Fig. 3(b)). Alternatively, Att is not selected as a forwarder although it is in a more favorable position (DoS attack, selfish behavior).

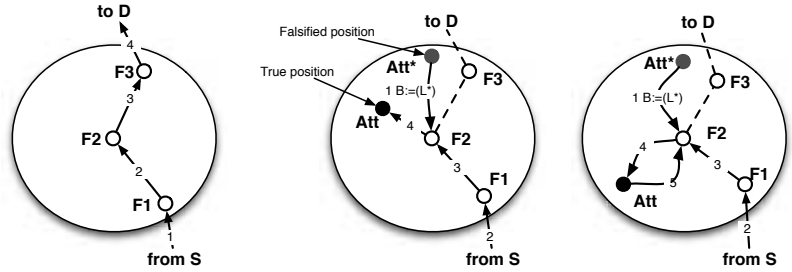
Attack on location service. Att falsifies the source, sender, or destination location in LREQ $OUT(Att) : LREQ := (S, D, L_D*, L_S*, L_{Att}*, TS_{SL}, TS_{DL})$. With a wrong destination location, the LREQ is sent into wrong direction and never reaches D. Using a falsified source location, LREP and reply data messages to the source will steered to wrong location. With a wrong sender location, the neighbor learns wrong location of attacker. Falsifying the location in LREP can be regarded as a more clever attack. It has similar consequences, but unlike LREQ, LREP messages are not broadcasted and therefore a single node attack is sufficient. Again, source, sender, and destination location can be spoofed.
$IN(Att) : LREQ := (S, D, L_S*, L_D*, L_{Att}*, TS_{SL})$
$OUT(Att) : LREP := (S, D, L_D*, L_S*, L_{Att}*, TS_{SL}, TS_{DL})$

Attack on forwarding and recovery. Similarly, at forwarding of data messages an Att may falsify source, sender (its own), and destination location. With a falsified destination location, a packet is simply sent to the wrong location not reaching the destination. With a falsified source location the reply data packets are mis-directed (probably to the attacker itself). With a wrong sender location, a neighbor learns a wrong location of attacker.

3.4 OLSR

OLSR is a proactive link state routing protocol, designed specifically for mobile ad-hoc networks. OLSR employs an optimized flooding mechanism for diffusing link-state information, and diffuses only partial link-state to all nodes in the network. Routing control traffic deploys HELLO messages and exchange of topology information using TCP messages. HELLO messages serves to three main tasks: link sensing, neighbor detection and

[4] A planar graph is a graph in which no two edges cross) and data packets are routed detour around a region (called 'face').



(a) Regular forwarding process via forwarding nodes F1, F2, and F3

(b) Attacker Att falsifies its location and is selected as the next hop instead of node F3

(c) Att causes a routing loop

Figure 3: Two exemplary attacks in georouting

Multipoint Relay (MPR) selection signaling. They are exchanged periodically between neighbors only. A node X sends periodically to every neighbor N_i, $i = 1, \ldots, n$

$$OUT(X) : HELLO := (X, TTL, HC, MSN, W, LC || N_1, \ldots, N_{i-1}, N_{i+1} \ldots, N_n) \quad (1)$$

where TTL is the time to live, HC the hop counter, MSN the message sequence number, W (willingness) indicates the intention of a node to forward data on behalf of others and, therefore, to be elected as MPR node. LC contains information about symmetric neighbors and MPR neighbors. HELLO messages are never forwarded, and upon the reception of such a datagram nodes populate the link set, 1-hop and 2-hop neighbor set and MPR set. MPR nodes are used to reduce the number of retransmissions that occur in a specific region. Each node in the network selects, independently, its own set of MPR among its symmetric 1-hop neighborhood. That means that the union of symmetric link sets of the MPR nodes contain the symmetric 2-hop topology of the network. A source announces the result of the heuristic MPR process setting the parameter LC to MPR_NEIGH in the HELLO packet. The parameter W, an integer value between 0 and 7, is used to elect MPR nodes. A selfish node announcing the value 0 (WILL_NEVER) never acts as an MPR. A node announcing value 7 (WILL_ALWAYS) has higher probability to be elected as an MPR. Therefore, since MPR nodes distribute topological changes in the network, this feature could be exploited to build up others attacks (e.g sinkhole). Moreover, HELLO packets cannot be forwarded. If a malicious node forwards these datagrams the network is flooded with wrong information about neighbors. This action, by affecting the population of the link set, poisons the 1-hop neighborhood. An attacker, by means of spoofing, could also mount impersonation attacks. The originator (X) indicates the source of the message. This value is fixed and cannot be changed along the path in the network. Furthermore,

a node may decide to announce a wrong set of neighbors $N_1, \ldots, N_{i-1}, N_{i+1} \ldots, N_n$ (link spoofing). This may result in unreachable nodes from the network (i.e. if the set is incomplete a node might ignore the sender). Finally, message sequence numbers (MSN) are used to provide freshness of the datagrams. Unfortunately, if the sequence number mechanism is not properly designed the protocol suffers of replay attacks. Different solutions have been proposed , e.g. [CAJ+03], however, a right tradeoff between complexity and computing requirements must be tuned. TC messages are periodically broadcasted by MPRs

$$OUT(MPR) : TC := (S, TTL, HC, MSN, AN_SN || AN_MAi, \ldots, AN_MAn) \quad (2)$$

The purpose of a TC message is to diffuse topological information from a source S to the entire network. Thus, a TC message contains a set of bi-directional links between a node and a subset of its neighbors (Advertised Neighbor Main Address). TC messages are diffused via MPRs in order to populate the topology information base on every node of the ad hoc cloud. Attacks against the set of announced $ANMA$ are possible. Finally, the parameter $ANSN$ (Advertised Neighbor Sequence Number) is associated with the advertised neighbor set. A malicious node can exploit this feature to change the freshness of a datagram.

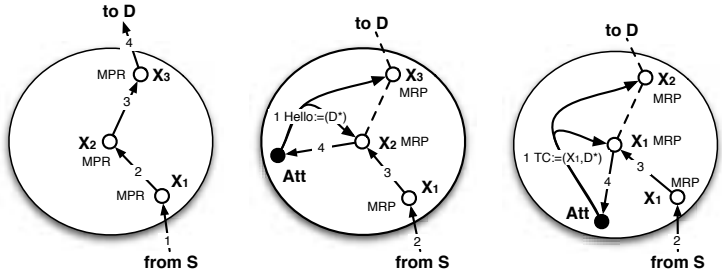
The OLSR-specific attacks include:

Attack using HELLO messages. HELLO packets are never forwarded. That means an attacker cannot prevent a node to send advertisements to its neighbors. The main chance a node has to manipulate a route is to be elected as MPR. To do that an attacker should announce its willingness to forward traffic. Unfortunately, a node cannot prevent somebodyelse to be elected as an MPR; in fact a node elects locally its set of MPR and then it spreads the info to the neighbors. A node X may also claim to be Y (impersonation):
$OUT(X) : HELLO := (Y, TTL, HC, MSN, W, LC || N_1, \ldots, N_{i-1}, N_{i+1} \ldots, N_n)$

Attack using TC messages. A malicious MPR can announce a wrong set of symmetric links:
$IN(MPR) : TC := (S, TTL, HC, MSN, AN_SN || AN_{MAi}, \ldots, AN_{MAn})$
$OUT(MPR) : TC := (S, TTL, HC, MSN, AN_SN || AN^*_{MAi}, \ldots, AN^*_{MAn})$

4 Countermeasures and limitations

The major motivation of this Section is the deeper insight that for protocols like DSR, AODV, OLSR, or Geo-routing prevention or detection of all the above listed attacks is impossible. Nevertheless, an efficient countermeasure against the different attack classes is a proper selection of the routing protocol itself although clearly, the selection may be restricted due to the specific requirements of the scenario.



(a) Regular forwarding process.

(b) Attacker Att claims to have D as a neighboring node and is potentially included in the route.

(c) Similar to Fig. b, but Att alters a TC packet and is again included in the route

Figure 4: Two exemplary attacks on link state exchange in OLSR

In Table 4 we list major characteristics of available security approaches for mobile ad hoc networks routing protocols. Note that we do not claim this tabular to offer the complete list of all security work done in this area.

Next we will list a set of thesis with respect to the already 'by nature' integrated protection level of the compared protocols and address countermeasures for remaining attacks in general. For each thesis the subsequent defense gives an explanation. Nevertheless the explanation should be understood more to stimulate the discussion than to be an ultimate statement.

Thesis-1: Attacks on proactive routing protocols tend to have a 'fuzzy' impact...

Defense-1: Since faking a broadcast message has only impact for a very limited period of time also the influence of the local or global view of the network is time restricted. In addition, since malicious manipulations of proactive messages are not directly addressed to any particular payload transmission process, even the attacker cannot foresee the exact impact of his attack. Thus, attacks tend to address more DoS attacks.

Thesis-2: Proactive protocols (OLSR) are more robust against active attacks than reactive routing protocols (DSR, AODV, Georouting)....

Defense-2: Case 1: An attacker which is not an MPR is not forwarding foreign traffic. Thus, the attacker is not enabled to divert, manipulate and/or spoof traffic. If the attacker's nature is selfish he feels fine not being an MPR since this position already saves as much energy as possible. Here we note that selfish behavior does not cause any kind of misbe-

	[MGLB00]	[YML02]	[PW02]	[BB02]	[PH02]	[BH02]	[LPW03]	[ZYC03]	[HPJ02]	[Za01]	[CAJ+03]
protocol	DSR	DSR	DSR	DSR	any	any	any	DSR	DSR	AODV	OLSR
strategy **d**etect/ **m**otivate	d	d	d	d	d	m	m	m	d	d	d
topology **p**ure/ **s**tub	p	p	p	p	p	p	s	s	p	p	p
scheme **c**rypto/ **p**romis.	p	c/p	c/p	p	c	c	c	c	c	c	c
reactive											
request/	-	r	r	r	r	r	r	r	r?	r	-
reply/	-	r	r	r	r	r	r	r	r?	r	-
forward/	f	f	-	f	-	f	f	f	-?	-	-
maint.	-	m	m	m	m	-	-	-	m?	-	-
proactive											
hello/	-	-	-	-	-	-	-	-	-	-	h
topology	-	-	-	-	-	-	-	-	-	-	t

Table 2: Security concepts for MANET routing protocols

havior for the rest of the ad hoc cloud. If the attacker's nature is malicious and he wants to perform active or passive attacks, he aims to be elected as an MPR. To get more influence on the routing the attacker sends HELLO messages with high willingness. But even local impact is difficult to achieve since the MPR election is basically based on majority voting from the whole set of neighbors. In addition we note that an attacker cannot keep locally broadcasted HELLO messages from honest neighbors to be received by others since they are transmitted single hop.

Case2: Once the attacker is an MPR he is in the position to actively manipulate forwarding processes of foreign data. Nevertheless, he is only ensured to be MPR for the current election period.

Thesis-3: Route manipulation attacks on reactive routing protocols have a direct impact on a particular transmission process...

Defense-3: Active attacks of an intermediate node like deleting, adding or modifying entries in an RREQ message directly result in an faked routing behavior of other nodes in the subsequent forwarding process. The attacker can foresee the consequence of an attack with reasonable probability, e.g. that the route is not chosen, the path is diverted, etc.

Thesis-4: "Watchdog mechanisms" have only limited value for reactive routing protocols...

Defense-4: In a network where one cannot foresee which particular nodes will be the forwarding nodes next, all nodes need to perform the watchdog tasks to subsequently communicate its observation results to other nodes. In addition, such concept may cause wrong accusations and tremendous additional transmission overhead. The latter is true since observations need to be communicated to nodes that are multi hops away from the attacker. Only in case the whole ad hoc cloud has a knowledge on misbehaving nodes, the watchdog mechanisms is not only enabled to detect but also to react. To summarize, a watchdog mechanism for reactive routing protocols is heavyweight with only limited benefit.

Thesis-5: "Watchdog mechanisms" are valuable for proactive protocols (OLSR)...

Defense-5: By applying the promiscuous mode single hop neighbors can act as watchdogs and observe each others behavior to detect misbehavior. Extending OLSR with a watchdog mechanism perfectly fits since the major disadvantage of watchdog - its heavyweightness in terms of data and computation overhead -, can be restricted in this context. First, due to Thesis 2, Case 2, watchdog is only applied to MPRs and not to all nodes of the network. Second, communication of the watchdog result only needs to be done single hop and can be synchronized with broadcast period of the HELLO message. Thus the results of the watchdog mechanism are considered for the next election period and nearly come for free.

Thesis-6: Injection of signaling messages is always possible, both for reactive and proactive protocols...

Defense-6: First of all there is no way to prevent an attacker from transmitting fabricated or replied signaling messages. Consequently only detection based mechanisms may help. Since we feel for the general case watchdogs cause too much overhead, only detection with the help of authentication plus timestamps remains a possible solution. Timestamps provide freshness at the cost of an overall (coarse granular) synchronization of the network which is not trivial but necessary to detect reply attacks. Even in presence of an authentication mechanism with and an overall time synchronization, fabrication of signaling messages is still not detectable.

Thesis-7: Motivation based cooperation schemes increase forwarding participation for reactive and proactive routing protocols...

Defense-7: Beside detection based approaches like [YML02], [PH02], [HPJ02], [ZA02] there is also the trend to some incentive based approaches that do not claim to detect misbehavior but motivate the community to play fair. Examples are [BH02], [LPW03], [ZYC03]. Generally these approaches cause much less overhead in terms of computation and transmission to achieve non-selfish behavior in the network. Their appliance is totally independent from the underlying routing protocol. Such approaches may use cryptographic primitives, either symmetric or asymmetric.

Thesis-8: Symmetric crypto schemes are energy saving but do not scale...

Defense-8: Compared to asymmetric crypto schemes we observe that symmetric schemes are in the order of 10^3 less computation intensive. On the other hand the symmetric key is established per pair of nodes. In large ad hoc networks where an security association to all other nodes may in principle be necessary, this approach will not scale.

Thesis-9: Asymmetric crypto schemes have considerable overhead but scale...

Defense-9: The overhead of asymmetric crypto schemes is twofold. On the one hand they cause much more computation overhead like also noted in Thesis-8. On the other hand also the data overhead for the storage of certificates and transmission of digital signatures (plus certificates) is not negligible over the wireless. Depending on the security level and the chosen digital signature scheme the size of a digital signature is approximately 150 bytes whereas the certificate size is about 600-700 bytes. Nevertheless asymmetric schemes are most attractive in large ad hoc networks since public keys can be easily broadcasted over the wireless and each receiver of somebody's public key is enabled to very the originator.

Thesis-10: The value of "Watchdog" and authentication schemes on active attacks differs...

Defense-10: Although watchdog and crypto authentication schemes both aim at detection, they provide different results: With a watchdog in place the neighbor of an attacker can doubtlessly detect the attacker. However it is a problem to reveal this information to other ad hoc nodes in non-repudiative manner. With an authentication protocol in place in case

of an attack one can infer that an attack happened. The Verifier of an incoming message can not infer who was the attacker.

5 Conclusion

The above observations may help to get a more detailed picture of the problems arising when designing 'secure' ad hoc routing protocols. Note that we did not mention well known problems in infrastructure-less environments, e.g. the absence of a trusted certification authority and possible solutions to overcome this problem, e.g. by threshold cryptography. As a first recommendation for designing new security solutions in the context of ad hoc networks we should restrict ourselves trying to reach the elementary security objectives with a reasonable message overhead and delay. We should not waste time trying to achieve unrealistic security and reliability issues that result in unacceptable signaling overhead on a meta level by only limited gain under some particular circumstances.

In the Introduction the authors sketch out what routing protocols in general should routing aim at. Apparently most of the requirements seem to be fulfilled, but, according to thesis 6, we notice that injection of forged control messages is always possible. Also dropping, partial or complete, is an operation that strictly depends on the 'human' behavior of the attacker. In both cases the only countermeasure available is detection but even a detection scheme is a limitation for the system itself. In fact, detection is prerequisite for reaction. Intuitively, once misbehaving nodes have been identified, we need to propagate and share this knowledge with the other member of the ad hoc cloud. How can we reach an acceptable level of trust?

This reasoning lead us to the conclusion that when proposing a security architecture, researchers and protocol engineers have to take in account always a degree of vulnerability. The goodness of the solution will then depend on how much the system can tolerate malicious behaviors. For instance, if we consider two ad hoc clouds **a** and **b** composed by twenty nodes each, and we fix the level of tolerance $T(a) = 0.8$ and $T(b) = 0.7$ it means network **a** can tolerate not more than four misbehaving nodes and network **b** not more than six. The fine tuning of this parameter is probably the best countermeasure we can deploy against known and unknown attacks.

References

[BB02] Buchegger, S. und Boudec, J.-Y. L.: Performance analysis of the confidant protocol (cooperation of nodes: Fairness in dynamic ad-hoc networks). In: *In ACM MobiHoc, Lausanne, Switzerland.* June 2002.

[BH02] Buttyan, L. und Hubaux, J.: Stimulating cooperation in self-organizing mobile ad hoc networks. In: *In ACM Journal for Mobile Networks (MONET), special issue on mobile ad hoc networks.* Summer 2002.

[CAJ+03] Clausen, T., Adjih., C., Jacquet, P., Laouti, A., Muhltaler, P., und Raffo, D.: Securing the olsr protocol. In: *Proceedings of IFIP Med-Hoc-Net 2003.* June 2003.

[CP03] Clausen, T. und P.Jacquet: Optimized link state routing protocol (olsr). In: *EXPERIMENTAL RFC 3626.* October 2003.

[HPJ02] Hu, Y.-C., Perrig, A., und Johnson, D.: Ariadne: A secure on-demand routing protocol for ad hoc networks. In: *In 8th Conference on Mobile Computing and Networking (ACM Mobicom2002), Atlanta, Georgia*. S. 12–23. September 2002.

[KK00] Karp, B. und Kung, H.: Gpsr: Greedy perimeter stateless routing for wireless networks. In: *Proceedings of the Sixth Annual ACM/IEEE International Conference on Mobile Computing and Networking, MobiCom*. S. 243–254. Boston, MA, USA. August 2000.

[LPW03] Lamparter, B., Paul, K., und Westhoff, D.: Charging support for ad hoc stub networks. In: *In Elsevier Journal of Computer Communications, Special Issue on Internet Pricing and Charging: Algorithms, Technology and Applications, Elsevier Science*. August 2003.

[MGLB00] Marti, S., Giuli, T., Lai, K., und Baker, M.: Mitigating routing misbehaviour in mobile ad hoc networks. In: *6th International Conference on Mobile Computing and Networking ACM MobiCom Conference pp. 255-265*. August 2000.

[PERD03] Perkins, C., E.Belding-Royer, und Das, S.: Ad hoc on-demand distance vector (aodv) routing. In: *EXPERIMENTAL RFC 3561*. July 2003.

[PH02] Papadimitrados, P. und Haas, J.: Secure routing for mobile ad hoc networks. In: *In SCS Communication Networks and Distributed Systems Modelling and Simulation Conference, Texas, USA*. January 2002.

[PW02] Paul, K. und Westhoff, D.: Context aware detection of selfish nodes in dsr based ad-hoc networks. In: *IEEE Globecom 2002, Taipei, Taiwan*. November 2002.

[YML02] Yang, H., Meng, X., und Lu, S.: Self-organized network layer security in mobile ad hoc networks. In: *ACM Workshop on Wireless Security (WiSe) in conjunction with ACM MobiCom 2002*. September 2002.

[Za01] Zapata, M.: Secure ad hoc on-demand distance vector (saodv) routing. In: *Internet Draft draft-guerrero-manet-saodv-00 (work in progress)*. October 2001.

[ZA02] Zapata, M. und Asokan, N.: Securing ad hoc routing protocols. In: *In ACM Workshop on Wireless Security (WiSe'02), in conjunction with ACM Mobicom 2002, Atlanta, Georgia*. 2002.

[ZYC03] Zhong, S., Yang, Y., und Chen, J.: Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. In: *In Proceedings of IEEE Infocom 2003, San Francisco, USA*. 2003.