AntSec, WatchAnt, and AntRep: Innovative Security Mechanisms for Wireless Mesh Networks

Parag S. Mogre, Kálmán Graffi, Matthias Hollick, and Ralf Steinmetz
Email: {pmogre,graffi,mhollick,rst}@kom.tu-darmstadt.de
Multimedia Communications Lab (KOM), Technische Universität Darmstadt, Merckstr. 25, 64283 Darmstadt, Germany

***Abstract*— Wireless Mesh Networks (WMNs) build on user nodes to form the network's routing infrastructure. In particular, the correct forwarding behaviour of each intermediate node on a multihop path from a source node to a destination node is crucial for the functioning of the mesh network. However, current secure routing solutions and misbehaviour detection mechanisms are not sufficient and are mostly inapplicable in mesh networks based on state-of-the-art wireless technology. In particular, hop-by-hop per-link encryption mechanisms break solutions that are based on the overhearing of the wireless channel, which leads to severe problems in the presence of misbehaving nodes. We present *AntSec*, *WatchAnt*, and *AntRep*, which together address the above security gap. AntSec guarantees integrity and authenticity of routing messages, WatchAnt detects misbehaviour in forwarding data messages as well as routing messages and in addition is able to cope with per-link encryption at the MAC layer. AntRep is a reputation management system and helps take punitive action against misbehaving nodes. AntSec, WatchAnt, and AntRep are well suited for WMNs with a quasi-static network topology. Through a thorough evaluation we show the improved routing performance of AntSec working together with WatchAnt and AntRep.**

I. Introduction

In the recent years we have seen continuously increasing demand for fast and reliable ubiquitous network access. Providing anytime-anywhere broadband network connectivity is, thus, becoming important for network providers. Wireless broadband networks are a viable solution for satisfying the above demand. However, network providers are continuously aiming towards a reduction in the maintenance costs for the wireless networks. Wireless Mesh Networks (WMN), where part of the network's infrastructure is maintained by user controlled devices, allow the network providers to supply ubiquitous broadband network coverage and, at the same, time reduce maintenance cost for the network infrastructure.

The term WMN usually refers to a network formed by a set of wireless nodes, which collaborate to forward data from a source node to the desired destination node. In this aspect WMNs are similar to the so-called ad hoc networks, where a set of nodes come together spontaneously to form a network. Unlike ad hoc networks, the nodes participating in a WMN are considered to be stationary and, thus, provide a stable wireless network backbone. Routing protocols developed for ad hoc networks have to cope with high mobility of the nodes and have to ensure correct routing in the presence of rapidly changing network topology. Currently, research in the area of ad hoc networks focuses mainly on challenges such as scalability, performance, etc. Unfortunately, security issues are often neglected and not addressed during the design of new protocols. In contrast to ad hoc networks, WMNs are usually not considered to have power limited or even mobile nodes. This opens up interesting avenues for optimization in the WMN (see [1] for a detailed survey of WMNs).

WMNs rely on individual user nodes in the network to correctly forward data from the source to the destination over a multihop path. Each node participating in the network must act as a router and forward messages on behalf of other nodes in the network. This poses several security challenges for the network infrastructure. Malicious routing behaviour by a node may be profitable for the node, e.g. resulting in resource savings at the node. This leads to several challenging issues which have to be solved. Especially security in routing needs further research. Participating nodes have to be authenticated, they should be forced to participate in routing and attacks have to be detected. It is important to introduce the necessary security features in this early state of development of protocols for WMNs to avoid costly security patches or even a lack of security in the WMN.

In the current paper we address the above security needs in WMNs. In particular, we consider networks that operate using encrypted wireless links. Our contribution is as follows:

- *AntSec*: A novel stigmergy-based probabilistic routing solution is presented. Our solution, AntSec, guarantees integrity and authenticity of the routing messages. AntSec is a probabilistic, proactive multipath routing algorithm, which is resilient against forging, modification and dropping attacks.
- *WatchAnt*: A novel mechanism to detect forwarding misbehaviour (for both routing as well as data messages) is presented. WatchAnt is able to cope with encrypted links, although, encrypted links complicate the monitoring of behaviour of neighbouring nodes. WatchAnt detects both malicious packet dropping as well as packet modification at misbehaving nodes.
- *AntRep*: The reputation management system, AntRep, is presented to interoperate seamlessly with AntSec and WatchAnt. AntRep serves as a local database for each node to manage the obtained reputation information and to trigger corrective actions if a misbehaving node is detected.

The stigmergy-based solutions presented are able to opti-

0742-1303/07 $25.00 © 2007 IEEE
DOI 10.1109/LCN.2007.102

IEEE computer society

mize their performance with increasing time of operation of the WMN. Thus, the presented solution is highly suitable for WMNs where the network topology is quasi-static. In Section II we outline the security goals for our mechanisms and list the assumtions we made while designing our solutions. In Section III we look at related work, classify it and show why existing solutions are not applicable for state-of-the-art WMNs. In Section IV we present our solution. In Section V we provide a thorough evaluation of our proposed solutions. This is followed by a conclusion and pointers for further research in Section VI.

II. Problem Statement and Assumptions

We have identified routing security as a critical issue in WMNs. We next describe the scenario as well as the assumptions behind our work. We then state the security goals for our mechanisms (AntSec, WatchAnt, and AntRep).

We identify two application scenarios for our security mechanisms, namely subscription-based and subscription-less, open networks. The term open network refers to the possibility that new nodes can join the existing WMN in an organic manner. The term subscription-based identifies a WMN which is deployed by a network provider and only nodes which are registered with the network provider are allowed to join the network. In contrast, in a subscription-less network there exists no network provider, but arbitrary nodes are allowed to join. We focus on open, subscription-based mesh networks and assume the existence of a trusted third party (the network provider). Each node that wants to gain access to the network has to be authorized out-of-band to allow us to punish misbehaviour. An example for such a WMN could be a mesh network using the IEEE 802.16 standard's MeSH mode (see [2] for an introduction to the MeSH mode of the IEEE 802.16 standard [3]).

We further assume that the communication takes place over a shared wireless medium, where the nodes are able to both send as well as receive data. Links between nodes are assumed to be bidirectional, i.e. given a link *L(A,B)* between nodes *A* and *B* in the WMN there exists the link *L(B,A)*. The WMN may deploy per-link encryption at the Medium Access Control (MAC) layer. Thus, data transmissions on link *L(A,B)* can be encrypted by *A* such that only neighbour *B* is able to decode the transmitted data. Please note that our solution is applicable to mesh networks with and without per-link encryption. We also assume the existence of a mechanism to broadcast data to all neighbours without encryption. The above assumptions are inline with the conditions and assumptions made by state-of-the-art MAC standards supporting mesh networks (e.g. the IEEE 802.16-2004 standard). We design the routing protocol such that for each packet a node *N* receives, the node is able to obtain information about the previous two hops along which the packet was forwarded to the node *N*. For example if the packet travels along a path *S-N1-N2-N3-N* to an intermediate node *N* then the node *N* knows that the two previous nodes which forwarded the packet are nodes *N2* and *N3* in this example.

The security goals we consider when designing our solution are as follows. The security goals encompass goals for nodes individually as well as goals for the entire routing system/network (control-plane and data-plane).

1) Authenticity and authorization of the source and destination node can be verified by all nodes on the route.
2) Authenticity and authorization of neighbours can be verified by nodes en-route.
3) Correct routing functionality shall be maintained (e.g. loop free routing, up-to-date routes, etc.).
4) Forging of routing messages in the name of other nodes shall have no effect on routing.
5) Manipulation or dropping of routing messages shall be detected.
6) Manipulation or dropping of data packets shall be detected.
7) Misbehaving nodes shall be detected and identified, so that various punishment methods can be applied.

In addition to these security goals, our solution should be as efficient as possible, i.e., in terms of computational effort the usage of asymmetric cryptography should be avoided for frequent operations such as packet forwarding due to its computational complexity. Routing overhead shall be kept as low as possible, unnecessary transmissions shall be avoided, etc. To harness the power of decentralized operation of the network, our solution shall base routing and security decisions on local information, wherever this is possible.

III. Related Work

The detection of malicious nodes that refuse to forward messages is a challenging task in decentralized, open networks. This is particular true for wireless multihop networks such as WMNs. In the area of mobile ad hoc networks, recently a number of security solutions have been discussed. Various approaches depart from today's de-facto standard routing protocols for wireless multihop networks: Dynamic Source Routing (DSR) [4] and Ad Hoc On-demand Distance Vector (AODV) [5] and reach a variety of specialized security goals, typically by utilizing asymmetric and/or symmetric cryptography in combination with protocol mechanisms. See Djenouri et al. [6] for an up-to-date survey on security issues and solutions in such networks.

To detect misbehaviour only few approaches exist. Marti et al. introduced in 2000 the idea of *Watchdog* and *Pathrater* [7] in order to solve the problem of malicious nodes. The main idea of the mechanism is to store an identifier for every packet forwarded to a neighbour and, by overhearing, validate whether the neighbour forwards this packet or not. Promiscuous listening on the wireless channel comes with several limitations. Scenarios exist, where transmission collisions occur on the physical layer and the behaviour of neighbouring nodes cannot be verified. Also, efficient bandwidth utilization might rely on planning of simultaneous transmissions within a two-hop neighbourhood, which also prohibits reliable promiscuous listening. Despite these limitations, Watchdog is still one of

the most common building blocks in various proposed security frameworks.

Alternatives to Watchdog are rare. *Nuglets* [8] and *Sprite* [9] are both incentive-based solutions that are based on accounting of the forwarding service. In Nuglets, accounting is done locally, but requires tamper-proof hardware, while in Sprite receipts for delivered packets are saved and a trusted accounting server is used for accounting based on these receipts. This results in high computational and storage requirements. Kargl presents in [10] *Iterative Probing*, a mechanism to detect blackholes on routes. Kargl assumes a field in each packet that contains encrypted information only decipherable by a single node on the route. This node has to acknowledge the receipt of the packet. Starting from the destination node, the source node iteratively adresses every single node on the route. Malicious nodes that drop messages can, thus, be identified by means of the probing mechanism: the last acknowledgement is either from the malicious node in the route or from its precursor. However, Kargl's approach is limited to source routing and necessitates changes on the network layer. In [11] Djenouri et al. propose the usage of signed two-hop acknowledgments. This results in high traffic overhead even after optimization.

Current state-of-the-art wireless technology poses strong constraints that have not sufficiently been considered in related work. In particular, hop-by-hop encryption on link layer prohibits overhearing of forwarded messages. Moreover, there exist reservation-based MAC layers, which might only possess limited broadcast-capabilites, thus, making the use of flooding-based reactive routing protocols prohibitively expensive. We identified that stigmergy-based routing algorithms fulfil most of the requirements of current and upcoming WMNs. Stigmergy-based routing protocols imitate the routing behaviour of insects such as ants, which randomly explore the landscape until they discover food resources. To inform their colony, they return on the path travelled towards the food and lay a pheromone trail that attracts further ants to use this route. Routing algorithms based on this principle were introduced in [12][13] by Di Caro et al. The advantage of this class of routing algorithms is that they do not demand broadcasting capabilities from the underlying MAC layer and routing decisions are made entirely localized, only based on probabilistic ratings of individual links. With this approach link quality aspects and security considerations can be considered in parallel. Security aspects for this class of algorithms are not discussed in sufficient detail, yet. In [14] Zhong and Evans have explored some of the security issues for stigmergic systems in general.

In summary, we witness a lack of feasible security solutions to detect node misbehaviour for the emerging research area of WMNs. This is especially true, if we consider state-of-the-art wireless technology. We propose a security framework to deal with the aforementioned challenges. In particular, we propose AntSec, a secure stigmergy-based routing protocol. AntSec works in close collaboration with WatchAnt, which provides a mechanism similar to Watchdog. However, it can also cope with encrypted links on the MAC layer. Moreover, it is designed to synergistically exploit the characteristics of the underlying stigmergy-based routing protocol. As a connector between both schemes we further introduce a reputation management system, AntRep.

IV. AntSec, WatchAnt and AntRep

In this section we first outline the components of our security framework and give an overview of the interactions among the different components. This is followed by a detailed description of the working of the individual components.

Fig. 1 shows the components of our security framework and the interaction among the different components. AntSec is a stigmergy-based routing algorithm which builds up on AntNet 1.1 [15] and provides several security extensions. WatchAnt is a challenge-response based misbehaviour detection mechanism, which is inspired by the work of Marti et al. [7]. Unlike contemporary watchdog mechanisms found in literature, WatchAnt is able to detect misbehaviour of neighbouring nodes even in the presence of encrypted wireless links. AntRep, the reputation management system that complements AntSec and WatchAnt, uses a multiple-threshold based system to classify neighbouring nodes into different categories based on their (mis)behaviour observed by WatchAnt. Fig. 1 shows the interaction among the above components. AntSec is responsible for updating the routing tables at the node and for acquisition and maintenance of the routes. The routing table contains entries per-destination and neighbour; these values denote the probability that the respective neighbour is selected as next hop for the particular destination. AntSec additionally uses information from AntRep to adapt these probabilites. WatchAnt observes the routing (control) packets sent by AntSec as well as data transmissions sent to neighbouring nodes and uses the routing information to perform checks to detect misbehaving neighbours. The misbehaviour information is then fed to AntRep, the reputation management system as shown in Fig. 1. We will next discuss the functioning of the individual components.

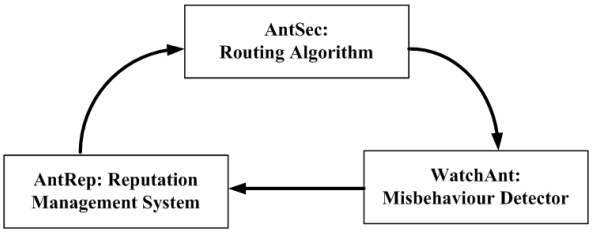


Fig. 1. Components of our security framework

A. AntSec: Securing Ant Routing

AntSec is a proactive, probabilistic, multipath, stigmergy-based, distributed, non-broadcast based, secure routing algorithm. AntSec is one of the first contributions for securing stigmergy-based routing algorithms (see the work of Zhong and Evans [14] for earlier work to study security vulnerabilities of stigmergic systems). Our work has to be seen in this

context. Here, we present the vital components of the routing algorithm.

Using AntSec each node maintains a routing table where for all tuples (*Destination Node*, *Neighbouring Node*) a routing probability is maintained denoting the probability of choosing the *Neighbouring Node* as the next hop for a packet destined to the *Destination Node*.

AntSec uses the following routing messages:
Discovery Forward Ant (DFANT): These are periodically sent to random destinations to find and establish new routes. DFANTs contain a registration certificate, and a public key hash authenticating the source node. The registration certificate and keys are obtained by nodes from a trusted third-party (network provider) as stated in our assumptions. In addition DFANTs have a path list containing all visited nodes. Flags in the DFANT allow the nodes on the route to request the registration certificate and public key of the destination node. Maintenance Forward Ant (MFANT): MFANTs are sent periodically to keep the current routes active, reinforce active routes, and adapt the routing probabilities to the current state of the network. Similar to DFANTs, MFANTs contain a list of visited nodes. However, as all the nodes on the route have received the registration certificate of the source and the destination during route setup from the DFANTs, MFANTs contain only a unique hash of the source registration certificate. Backward Ant (BANT): BANTs are sent by the destination nodes in response to received forward ants (DFANTs and MFANTs). Critical parts of the forward ants are signed and added to the corresponding BANT. Additionally, the public key and registration certificate of the destination node is added to the BANT if requested in the forward ants. BANTs also contain a complete list of visited nodes. All types of ants have an AntID, which uniquely identifies the ant. Due to the proactive nature of AntSec, routes are established before they are used. As only authenticated and authorized nodes shall participate, their registration certificate is contained in every DFANT.

Only authorized nodes are allowed to participate in the network. Therefore, upon receiving a DFANT, the registration certificate of the source node is checked for validity and stored if valid. In MFANTs the registration certificate is not contained in order to save bandwidth. During route establishment the certificate is propagated to and stored by every node on the route, so it is not needed to be sent repeatedly. The same holds for the Public Key and registration certificate of the destination node. Once a route is established, the validity of the destination node is checked and the Public Key of the destination node is distributed, there is no further need to provide this information in MFANTs. Invalid certificates and Public Keys can be detected easily, as the certificates may contain a hash of the Public Key and the information about the current network identifier of the corresponding node. With this approach only authenticated and authorized peers can establish routes.

In order to guarantee the integrity of routing messages a two step mechanism is used. BANTs are signed by the destination node, so that every node on the way back to the source node (using the path history) can verify the integrity of the BANT. To achieve this, the Public Key of the destination node is provided in the BANT upon request. The integrity of FANTs is guaranteed by a second look at the critical fields. Before forwarding a FANT, each node stores a hash of the critical fields of the FANT, i.e. the path history up to the current node, the certificates, source and destination identifiers. Each node, upon receipt of a BANT, checks the BANT's path history up to this node's occurrence and other immutable parts of the routing message, to see whether the corresponding FANT is known and has not been changed invalidly. Any invalid modification of the FANT results in a BANT which cannot be associated to its corresponding FANT. In case of a recognition, the integrity of the FANT is assured. The matching entry is then deleted from the stored memory. In case a corresponding FANT cannot be found, the integrity of the original FANT must have been compromised or a new FANT has been forged, in this case the BANT is dropped. For any unexpected error the previous hop (node) is punished with a reputation decrease. In case of a valid BANT the routing tables are updated.

We see that forging and invalid modification of routing messages have no effect. Even replays of old FANTs and BANTs do not cause harm, as replayed BANTs are dropped due to the missing corresponding FANT. Replayed FANTs have the effect of new FANTs, so benefit is gained from this attack (by helping to update routing tables). Only routing information taken from the BANT is used to update the routing tables, when the integrity and authenticity of the routing messages is assured.

Malicious nodes may try to cause inefficient routes or even loops, but effects of such attacks are limited, because each node can only determine the next hop of the routing packet.

One additional advantage of the stigmergy-based security approach is that attacks have to be performed several times to have effect. Routing probabilities change only significantly after several routing table updates. With increasing number of attacks, malicious behaviour is easier to detect.

B. WatchAnt: Watching the Extended Neighborhood

In this subsection we present details about WatchAnt, a novel misbehaviour detection mechanism for WMNs. To the best of our knowledge, WatchAnt is one of the first schemes proposed which can detect node misbehaviour even in the presence of per-link encryption at the MAC layer. Detection of misbehaviour in forwarding data by a neighbouring node is a difficult task in WMNs. Even when the wireless links are not encrypted, parallel transmissions scheduled within a two-hop neighbourhood to increase spatial reuse (as done by the TDMA based MeSH mode of the 802.16 standard) make promiscuous listening difficult, if not impossible.

WatchAnt is a challenge-response based scheme for detecting forwarding misbehaviour of neighbouring nodes. To make the presentation intuitive we will explain the functioning of WatchAnt with the help of the schema shown in Fig. 2.

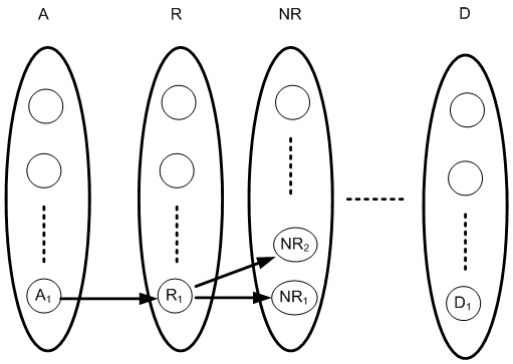


Fig. 2. WatchAnt working principle

Consider the sets of nodes *A*, *R*, *NR*, and *D* as shown in Fig. 2. *A* will represent the set of nodes generating the packet or forwarding the packet and wanting to verify the forwarding behaviour of the next hop for the packets. *R* will denote the set of relay nodes (next hops for packets transmitted by nodes in set *A*). *NR* denotes the set of next hops for the set of nodes in *R*; i.e. packets transmitted by nodes in set *A* to nodes in set *R* will be forwarded to nodes in the set *NR* on their way towards the destination (nodes in set *D*). The set *D* is assumed to be the set of destinations for the packets originated or initially transmitted/forwarded by the nodes in set *A*. Note that, just to keep the discussion simple, we assume that the sets are disjoint, they could in practice be non-disjoint. We use *P* to denote the set of packets originated or initially forwarded by nodes in set *A*. P_x^{ab} denotes the last packet transmitted by a node a to node b. P_{x-i}^{ab} thus, denotes the i^{th} packet preceeding the current packet transmitted by the node a to node b. We assume the presence of a mathematical function $hashsum(h1, h2)$ which is basically a mapping ($\{H\}$,$\{H\}$) $\longmapsto H$, where $H = B^{160}$ $h_i \in H$ for $i \in \mathbb{N}$ and $B = \{0, 1\}$. The set H can be considered to be a set of 160 bit hashes computed for individual packets using a hash function. Assume that the function $hashsum()$ is commutative as well as associative. If $hashsum()$ is in addition a one-way hash function it is beneficial for our mechanism, however, this is an optional feature. To simplify the notation we will use $hashsum(P_i, P_j, ..., P_k)$ to denote the repeated application of hashsum, $hashsum(h_i, hashsum(hashsum(P_j, ...$ $hashsum(P_{k-1}, P_k))))$, where h_i corresponds to the hash value for packet P_i.

We next give an overview of the challenge-response mechanism. The node (A_1) wishing to verify the forwarding behaviour of its neighbour (R_1) transmits a challenge (WatchAnt Request [challenged node's ID (R_1), packet count (i)]). The challenge identifies the addressed node, and asks it to reply with the forwarding information for the last i packets sent by the challenger to the challenged node. In this example, R_1 is requested to reply specifying information about the forwarding for the packets $P_{x-i}^{A_1R_1}$ ... $P_x^{A_1R_1}$.

R_1 then sends a response (WatchAnt Reply [(previous hop (A_1), next hop node ID (IDs from set *NR*, num. of packets forwarded (j), hash value for the packets)*]). As seen the response consists of a set of tuples identifying in each case the previous hop (the challenger), the next hop for a set of packets, the number of packets forwarded to the next hop, the hash value for all the packets sent to the next hop. To make the example more clear, assume that A_1 had sent a challenge as specified above. Further assume that the node R_1 has forwarded the packets only to a single next hop (*NR*$_1$) and the challenge had asked for the last 2 packets. The response then looks like [A_1,*NR*$_1$,2,$hashsum(P_x^{A_1R_1}, P_{x-1}^{A_1R_1})$]. This response is transmitted as a broadcast message without encryption. We denote the number of previous packets for which the $hashsum()$ is to be computed as $WaReqNum$. In the above example, $WaReqNum = 2$. This parameter determines the probability of detecting forwarding misbehaviour of neighbouring nodes.

Thus, in our scenario, when R_1 transmits the response, it will be received by its neighbours in the sets *A* and *NR*. Nodes which are addressed in the response will process the reply. In the above example, the nodes A_1 and *NR*$_1$ will process the received reply. By verifying the reply, either A_1 or *NR*$_1$ or both will be able to detect forwarding misbehaviour of node R_1 in case it is misbehaving. In general, we can say the the WatchAnt reply sent by the relays (set of nodes *R*) will be verified by the challengers (set of nodes *A*) and the reported next hops after the relays, i.e. the set of nodes *NR*. Both these sets of nodes need to be able to verify the reply, the set of nodes *R* needs to generate a reply. Hence, each of the above set of nodes needs to maintain certain data structures, which are described next.

Each node maintains two lists *InList* (for information about packets received from the neighbour), and *OutList* (for storing information about packets sent to the neighbour) for each neighbouring node. Let $hashIm(P)$ denote the hash of the immutable parts of packet P. An entry in the *InList*, for a packet received (P), contains $hashIm(P)$, the node identifier of the neighbour which transmitted this packet (*previous node*), and the node identifier of the node which transmitted the packet prior to the *previous node*. In addition, a field in the *InList* can be used to enter the information about the next hop for the packet.

The *OutList* contains for the packet transmitted (P), $hashIm(P)$, and the node identifier of the node to which the packet was forwarded. The number of entries in the above two lists can be limited to some maximum value. In addition, to be able to verify the WatchAnt reply, we need information about the previous two hops for the packets. Therefore, for each transmitted packet, a previous node identifier field is set in addition to the transmitter's node identifier. On receiving or transmitting a packet, the *InList* or the *OutList* are updated and all the fields in these lists are set as specified previously. A node periodically issues WatchAnt requests (challenges) as explained previously asking for the information about the previous $WaReqNum$ packets sent to the neighbour. The challenged node then uses the *InList* to find out the next hops for the last $WaReqNum$ packets received from the challenger.

Using the hash values for the packets in question found from the *InList* and the corresponding next-hops, the challenged node uses the $hashsum$ function to generate the WatchAnt reply. The WatchAnt reply is then transmitted. The challenger uses its *OutLists* to determine whether the *hashsum* reported by the challenged node matches the *hashsum* for the packets sent to the node. Other nodes (corresponding to the set *NR*) receiving the WatchAnt reply and addressed in the reply use their *InList* to check if the node really forwarded the packets it reports as forwarded. It is seen that a malicious node which lies and tries to manipulate the reply can fool only the challenger or the next hop but not both simultaneously and, hence, its forwarding misbehaviour will be detected.

The parameter $WaReqNum$ plays an important role in determining the ability of the WatchAnt mechanism to detect forwarding misbehaviour. Consider that a malicious node drops packets with a probability of P_{Drop} instead of forwarding them. Now the probability that a packet is not dropped is given by (1-P_{Drop}). Given a WatchAnt request asking for information about the last $WaReqNum$ packets is addressed to a malicious node, its malicious behaviour will not be detected if and only if it has not dropped a single packet in the last $WaReqNum$ packets. The probability that the node has not dropped a single packet in the last $WaReqNum$ packets is given by $(1 - P_{Drop})^{WaReqNum}$, which is equal to the probability that a malicious node will go undetected. Thus, the probability that a malicious node, dropping packets with a probability P_{Drop}, will be detected is $1 - (1 - P_{Drop})^{WaReqNum}$.

C. AntRep: Managing Reputation for Stigmergic Systems

To maintain a current state of its neighborhood, each node relies on AntRep as a reputation management system. For our scenario, reputation management is carried out in a distributed and decentralized fashion. In particular, AntRep represents all information gathered by WatchAnt about the behaviour of the extended neighborhood of a node as node – value pairs. These reputation values are updated periodically when positive or negative observations are made by WatchAnt. We next describe the parameterization of the reputation values, the system policies to react if certain thresholds are reached and the detailed process of updating reputations. We summarize the subsection by highlighting similarities and differences with related work in the area of repuation management.

If a node joins the network, its reputation value is 0. We define the following thresholds to apply for our reputation system.

- 25 Maximum reputation value
- 0 Initial reputation value
- -25 No-reputation-bonus threshold
- -40 Punishment threshold
- -60 Minimum reputation value

The symmetric range from [-25;25] describes the notion of normal operation of neighbours. The nodes' strategy is to perform normal routing operations to neighbours within this reputation range. Also, positive as well as negative observations lead to the below detailed change of the reputation value if within this range. Reputations below -25 indicate that a neighbour behaves maliciously. Reputation changes towards a better reputation value are no longer commenced as feedback to WatchAnt, but the reputation is only allowed to increase according to the restoration process described below. As soon as the reputation is under -40, two changes take effect. (1) The node is excluded from routing, i.e. its probability of being chosen as next hop to arbitrary destinations is set to the minimal value. (2) The node is denied service, i.e. messages generated by this node are no longer processed.

We distinguish between the thresholds at -25 and -40 to be able to adequately treat selfish behaviour of nodes, which might try to constantly operate with a bad reputation to avoid forwarding of packets for other nodes. From the threshold -25 on, these nodes rely on the (slow) mechanism of reputation fading to get back into normal operation, they are living on the edge of exclusion. In contrast, inactivity of a node is not considered harmful. To enforce continuous positive behaviour from benign nodes and to allow nodes identified as malicious to (slowly) recover their reputation, the reputation of a node is periodically updated as follows:

- If current reputation value $oldRep$ is positive: $newRep = 0.9 \cdot oldRep$
- If current reputation value $oldRep$ is between -40 and 0: $newRep = 0.98 \cdot oldRep$
- If current reputation value $oldRep$ is less than -40: $newRep = 0.99 \cdot oldRep$

The above models the reputation fading (or second chance) mechanism in our misbehaviour detection systems. Misbehaving nodes can return to normal service after an appropriate timeout. Thus, without any other triggering change, the reputations of all neighbouring nodes converge to the initial reputation value (0). As seen in this subsection, reputations are maintained locally, representing the subjective view of one node observing its neighbours. The reputation value of a single node maintained by two different neighbours can be completely different (also this single node can behave differently with respect to its neighbouring nodes). Each node decides based on its local reputation table, how to cope with each single of its neighbouring nodes. Mechanisms can be devised to use the local subjective observations and spread them as second-hand reputations (see Buchegger [16]). Second-hand reputations have been shown to increase the speed of detection of malicious nodes. For the results presented in this paper, we do not employ second-hand reputations, but rely only on local-observations and decisions to mimimize the protocol overhead.

In addition to positive or negative reputation updates based on the WatchAnt replies, we identified elements of the AntSec protocol which can be used to update the reputation of neighbouring nodes. Routing loops in received , e.g. FANTs, invalid BANTs received, BANTs received for which no corresponding FANT has been seen are all symptoms for node misbehaviour, and can be detected by a benign node. Benign nodes receiving

invalid protocol packets will not forward them. Thus, when an invalid packet is received from a neighbour, the neighbour is punished by decrementing its reputation value. There are only two events which trigger a rise in the reputation. (1) The referenced outgoing node receives a WatchAnt reply with correct information. (2) The node starting the examination process (WatchAnt requestor) receives a WatchAnt reply containing authenticated nodes and correct information about the forwarded packets. In this case the reputation rises, too.

In summary, AntRep provides a localized view of the reputation of the nodes in our stigmergy-based routing system. The necessary reputation input is provided by AntSec and WatchAnt feedback.

V. Evaluation

In this section we present a thorough evaluation of our security solution w.r.t. the obtained routing performance. To this end we perform a simulation study for two selected experimental designs. First, we analyse the most important performance indicators in a setup that models a realistic WMN. Second, we use an artificial topology to investigate the performance of our scheme for nodes that exhibit only probabilistic misbehaviour. We implemented our security framework using an extended and consolidated version of the JiST/SWANS [17] discrete event simulator. The extended simulator contains an implementation of the IEEE 802.16-2004 MeSH mode (MAC and PHY), which has been used for our study. We next present the individual experimental setups alongwith the results obtained. The duration of the individual tests allowed to reach the steady state of the network after the initial network entry procedure and the flow and route setup. We perform 20 replications for each experiment.

A. Simulation Results: Scenario I

Scenario I comprises a randomly generated WMN topology with 50 nodes. The nodes are stationary and the node degree is bounded to at most 5 neighbours. To stress-test our protocol, we simulated 25 Constant Bit Rate (CBR) flows between randomly selected (source, destination) pairs, each flow generating packets of size 512 byte ten times a second.

In a first test, we varied the fraction of malicious nodes in the network from 0% to 50%. Goal is to observe the effects on the packet loss rate as well as the effectiveness of our scheme to identify misbehaving nodes. As described in Section IV-A AntSec is by design immune to attacks on the routing (control) data itself. Hence, we model the malicious nodes to drop all data packets not related to themselves, but process routing packets. The routing scheme acts as our predictor variable. In particular, we study AntSec, AntNet (denoted as Ant) and Dynamic Source Routing (DSR). Ant represents a stigmergy-based proactive routing protocol, while DSR represents a reactive routing protocol from the ad hoc domain. The protocols were modified such that they could operate on top of the reservation-based MAC protocol of IEEE 802.16.

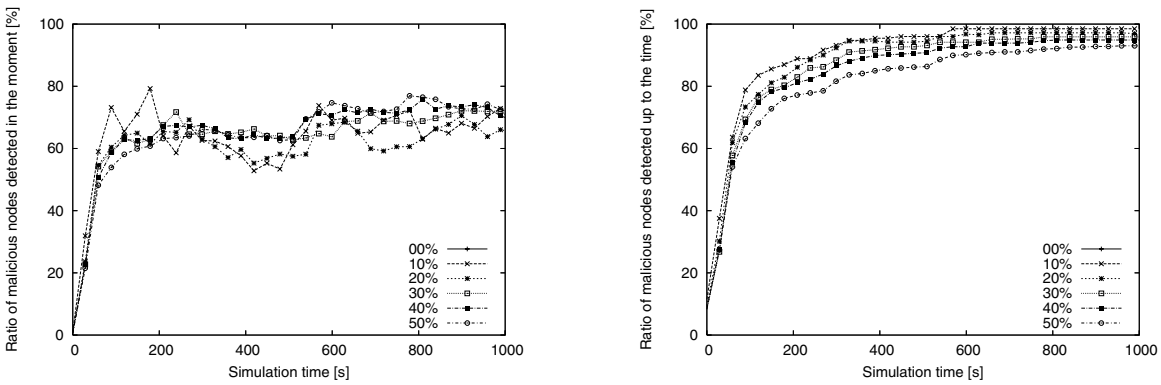
TABLE I
Scenario I: Mean percentage of data delivered

Routing	Percentage of malicious nodes in the network				
	10%	20%	30%	40%	50%
AntSec	81.2%	65.4%	46.7%	35.5%	25.2%
Ant	68.4%	48.7%	32.9%	27.0%	21.9%
DSR	63.2%	44.4%	30.7%	23.2%	17.7%

Table I shows the mean delivery ratio vs. a varying fraction of malicious nodes in the network. If malicious nodes are absent, all three routing protocols feature delivery ratios close to 100%, i.e. the loss was consistently below 1%. Please note that the stigmergy-based protocols show a negligible amount of packet loss if the time-to-live for packets is exceeded, which can happen due to the probabilistic nature of these schemes. For an increasing fraction of malicious nodes, AntSec outperforms the other routing protocols, which demonstrates the effectiveness of our security framework to detect and exclude malicious nodes from the routing. Despite this fact, the routing performance for all protocols deteriorates with an excessive amount of malicious nodes in place. Due to its probabilistic nature, Ant is inherently able to constantly adapt its routes, thus, avoiding malicious nodes to some extent. This results in a slightly better performance compared to DSR. We can conclude that the features of stigmergy-based protocols such as self-stabilisation need to be complemented with security mechanisms to combat malicious nodes. AntSec lays the groundwork for further research in this area.

The improved performance comes at a price, however. The average routing overhead of Ant is around 20% of the total number of bytes transported in the WMN (measured over all experiments). The security mechanism in AntSec leads to a mean overhead of 35%. DSR serves as a baseline for the overhead: because of the static setup, routes are discovered only once. It's overhead was less than 2% for all the above simulations. Optimization of Ant and AntSec, e.g. using the adaptation of the emission rate of DFANT and MFANT messages, can lead to a significant reduction in overhead, which is outside the scope of this work.

We perceive the detection quality of malicious nodes to be of high interest. Fig. 3(a) shows the fraction of identified malicious nodes at the given time. The network shows a stable detection quality of around 70% of all malicious nodes (i.e. in 70% of the cases at least one neighbour finds out a node is malicious). The remaining 30% account for nodes that are either false negatives (i.e. non-detected malicious nodes) or malicious nodes that have re-obtained an acceptable reputation over time. As described earlier, we consider such a second chance as vital to allow nodes to revert to positive behaviour, thus also mitigating false positives (wrongly excluded well-behaving nodes). Fig. 3(b) shows the cumulative number of correctly identified malicious nodes up to a given point in time. It can be seen that nearly 100% of the malicious nodes are detected over time for all setups. The results shows that AntSec is able to detect malicious nodes even for very high



(a) Fraction of malicious nodes identified at a given time

(b) Fraction of malicious nodes identified up to a given time

Fig. 3. Scenario I: quality of detection of malicious nodes over time

fractions of misbehaving nodes and, as a result, to adapt and improves the routes in the network.

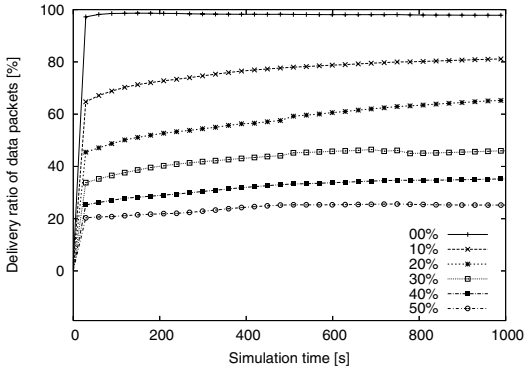


Fig. 4. Mean fraction of data delivered up to a given time

The benefit of the successive identification of malicious nodes by AntSec can be observed in Fig. 4, which shows the mean delivery ratio cumulated until the given point in time. The figure shows an increasing trend in the obtained delivery ratio over time for all tests. This implies that with increasing deployment time AntSec selects improved routes (avoiding malicious nodes) and, thus, improves the delivery ratio. In fact, we observed that when one considers the delivery ratio for the last 100 seconds of the simulation, AntSec shows delivery ratios that are up to eight percent higher than the average delivery ratio.

B. Simulation Results: Scenario II

The goal of Scenario II is to study the performance of AntSec for malicious nodes that exhibit probabilistic (mis-)behaviour. I.e. these nodes do not drop 100% of the data packets, but try to avoid detection by dropping fewer packets. Again we are discussing the effectiveness of WatchAnt in detecting malicious behaviour. Moreover, we investigate the working of AntRep in detail.

For Scenario II we used the artificial WMN topology shown in Fig. 5. We considered two CBR flows with the same data rate as in Scenario I. The two flows considered are $N_S \rightarrow N_D$, and $N_D \rightarrow N_S$. Node X marks the malicious node. It is active in the time-interval [200s;1200s] and acts maliciously in the interval [300s;1100s]. Thus, the node is malicious 80% of the time. We vary the degree of maliciousness during the latter period of time. Again, we perform 20 replications for each test.

TABLE II
DATA DELIVERY RATIO FOR ANTSEC IN SCENARIO II

Drop Ratio	20%	40%	60%	80%	100%
Delivery Ratio	92.2%	91.0%	87.8%	82.2%	82.5%
Std.Dev.	3.3%	5.0%	7.6%	9.5%	8.6%

Table II shows the mean fraction of data delivered to the destination using AntSec for differing drop rates of node X. It is seen that even with 100% packet drop rate of the malicious node AntSec achieves a mean delivery ratio of 82.5%. As a baseline, for the latter example Ant routing produces a mean delivery ratio of only 21.9% (which is only marginally better than the sustained delivery ratio of 20% that can be reached solely during the non-malicious interval). Fig. 6 shows the value for the reputation for the malicious node as computed by node N_2. The figure also illustrates the detection speed of WatchAnt.

We observed that the drop ratio of the malicious node makes essentially no difference to its detection probability in AntSec. For all studied drop rates, on an average 4.3 and

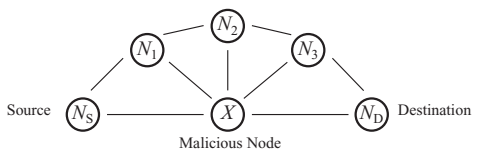


Fig. 5. Topology for Scenario II

4.55 nodes detected node X to be malicious, i.e. almost all neighbours. This effect is due to the design of the WatchAnt mechanism as shown in the following example. As discussed earlier, the parameter $WaReqNum$ is used during the creation of WatchAnt requests, and describes the number of packets for which a reception report is requested. If one of these $WaReqNum$ packets has not been forwarded, the WatchAnt reply is false, and hereby it is not relevant whether 20% or 100% of these packets have been dropped. Thus, by adjusting the parameter $WaReq$ it is possible to influence the detection quality of WatchAnt. For the above simulation we have chose $WaReqNum = 12$.

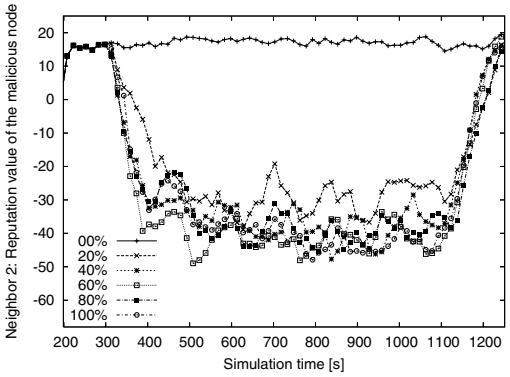


Fig. 6. Reputation of the malicious node X as computed by node N_2

C. Simulation Results: Summary

We have analyzed the performance of AntSec, WatchAnt, and AntRep for different scenarios. Our results are very promising and show that our framework is able to achieve the intended goal, namely to detect malicious nodes in WMNs that operate with the constraint (feature) of encrypted links between mesh nodes. We can conclude that stigmergy-based secure routing mechanisms are a viable alternative to existing secure routing schemes, especially if we consider organically growing networks. Additional results and a more detailed analysis can be found in [18].

VI. Conclusion

State-of-the-art Wireless Mesh Networks feature sophisticated hop-by-hop security mechanisms such as link-layer encryption. Being designed to secure the wireless transmission in the first place, these features complicate a wide range of current security solutions on the network layer. In particular, if we consider networks that operate with non-trusted mesh nodes, we lose the ability to transfer solutions from the domain of wireless multihop networks that require the overhearing of the wireless channel to identify misbehaving nodes. Moreover, hop-by-hop mechanism cannot replace end-to-end security mechanisms. Our security framework comprises the components AntSec, WatchAnt, and AntRep and is able to address the aforementioned research problems. Based on the principle of stigmergy, our solution shows very good performance in mesh networks with static topologies, even though the misbehaviour can be dynamic. We evaluate the proposed mechanisms by means of a simulation study atop of IEEE 802.16 MeSH mode. Our selected results show the feasibility of our approach. However, this work clearly marks only the beginning of the research in the area of mesh networks that operate under the provision of advanced mechanisms such as hop-by-hop link-layer encryption.

References

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless Mesh Networks: A Survey," *Computer Networks*, vol. 47, no. 4, pp. 445–487, 2005.

[2] P. S. Mogre, M. Hollick, and R. Steinmetz, "The IEEE 802.16-2004 MeSH Mode Explained," TU-Darmstadt, Germany, ftp://ftp.kom.tu-darmstadt.de/pub/TR/KOM-TR-2006-08.pdf, Tech. Rep. 08, 2006.

[3] IEEE Computer Society and IEEE Microwave Theory and Techniques Society, "802.16 IEEE Standard for Local and metropolitan area networks, Part 16: Air Interface for Fixed Broadband Wireless Access Systems," IEEE Std. 802.16-2004, October 2004.

[4] D. Johnson, D. Maltz, and J. Broch, *DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks*. Addison-Wesley, 2001, ch. 5, pp. 139–172.

[5] C. E. Perkins, E. Belding-Royer, and S. R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF RFC 3561, July 2003.

[6] D. Djenouri, L. Khelladi, and A. N. Badache, "A Survey of Security Issues in Mobile Ad Hoc and Sensor Networks," *Communications Surveys & Tutorials, IEEE*, vol. 7, no. 4, pp. 2–28, 2005.

[7] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks," in *MobiCom 2000*, 2000, pp. 255–265.

[8] L. Buttyan and J.-P. Hubaux, "Enforcing Service Availability in Mobile Ad Hoc WANs," in *ACM MobiHoc*, Boston, MA, USA, 2000.

[9] S. Zhong, Y. Yang, and J. Chen, "Sprite: A Simple, Cheat-proof, Credit-based System for Mobile Ad Hoc Networks," Department of Computer Science, Yale University, Tech. Rep. Yale/DCS/TR1235, 2002.

[10] F. Kargl, S. Schlott, A. Klenk, A. Geiss, and M. Weber, "Advanced Detection of Selfish or Malicious Nodes in Ad Hoc Networks," Proceedings of 1st European Workshop on Security in Ad-Hoc and Sensor Networks (ESAS 2004), September 2004.

[11] D. Djenouri and N. Badache, "Cross-Layer Approach to Detect Data Packet Droppers in Mobile Ad-Hoc Networks," in *IWSOS 2006*, vol. 4124. Springer, 2006, pp. 163–176.

[12] G. Di Caro and M. Dorigo, "Antnet: a Mobile Agents Approach to Adaptive Routing," Universite Libre de Bruxelles, IRIDIA, Tech. Rep. 12, 1997.

[13] G. Di Caro and M. Dorigo, "Antnet: Distributed Stigmergetic Control for Communications Networks," *Journal of Artificial Intelligence Research*, vol. 9, pp. 317–365, 1998.

[14] W. Zhong and D. Evans, "When Ants Attack: Security Issues for Stigmergic Systems," University of Virgina, Department of Computer Science, Tech. Rep. CS-2002-23, Apr. 2002.

[15] B. Baran, "Improved AntNet Routing," *SIGCOMM Comput. Commun. Rev.*, vol. 31, no. 2 supplement, pp. 42–48, 2001.

[16] S. Buchegger, "Coping with misbehavior in mobile ad-hoc networks," Ph.D. dissertation, Swiss Federal Institute of Technology (EPFL), April 2004.

[17] R. Barr, "JiST/SWANS manual," http://jist.ece.cornell.edu/docs.html, 2004.

[18] K. Graffi, "A Security Framework for Organic Mesh Networks," Master's thesis, TU-Darmstadt, May 2006.