

# Secure Communication over Heterogeneous Networks with Clustered Mobile Ad hoc Extensions

Dimitrios Vogiatzis, Spyridon Vassilaras and Gregory S. Yovanof  
Athens Information Technology  
e-mail: {dvog, svas, gyov}@ait.edu.gr

**Abstract** — In addition to classical security issues, clustered ad hoc networks face the possibility that some nodes may exhibit uncooperative behaviour. Therefore, misbehaviour detection and reputation mechanisms need to be implemented in order to reinforce node cooperation. In this paper, we address the issue of detecting non-cooperative behaviour during packet forwarding in heterogeneous networks with clustered mobile ad hoc extensions. The proposed solution incorporates end-to-end authenticated acknowledgments for each transmitted packet, combined with explicit authenticated alarms sent by legitimate nodes along the path to the source, every time they encounter a suspicious event. Low computational overhead is achieved by employing an adapted version of the TESLA symmetric key broadcast authentication protocol.

**Index Terms** — Ad hoc Networks, Packet Forwarding, Clusters, Misbehaviour Detection, TESLA.

## 1. Introduction

The emergence and adoption of wireless standards, such as the WLAN WiFi (IEEE 802.11a/b/g), WPAN Bluetooth (802.15.1) and 3G (UMTS/HSDPA) has boosted a vast range of applications and services, including Internet browsing, file transfer, messaging, news, games, entertainment, location based services, etc. At the same time, the wide deployment of these wireless technologies has contributed a lot to the advance and realization of the “always connected, best connected” vision. Unfortunately, despite this explosion in wireless connectivity, there is still increased need for capacity improvement, even in areas where coverage exists. This is especially true, for large urban areas and hotspots (airports, shopping malls, cafes, etc.), where multiple mobile users with different user profiles coexist and compete for network resources, trying to gain access to a variety of wireless services.

It is widely acknowledged that the 4G generation of networks will not be based on the prevalence of a single – entirely new or existing – technology but rather on the coexistence, synergy and efficient cooperation of heterogeneous systems and networks. One of the most popular and promising type of network architecture is the clustered ad hoc architecture (Fig. 1). Due to its inherent organizational features, clustered ad hoc

networks exhibit many benefits with respect to capacity, interference, frequency reuse and scalability.

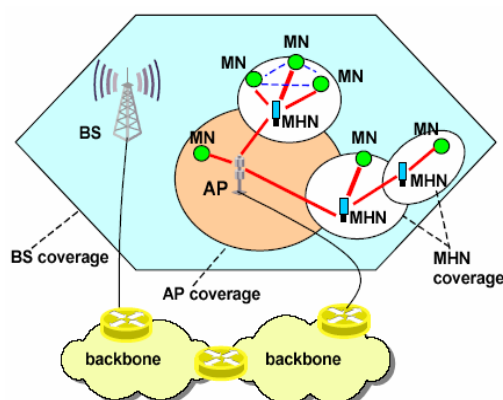


Fig. 1: Clustered Ad hoc architecture

A typical scenario describing the deployment of a clustered ad hoc network is in hotspot areas, where a large number of mobile users with high traffic needs are in the transmission range of each other. To increase the total capacity of such highly dynamic networks, a clustered mobile ad hoc network architecture can be used. In such a setting, a specific set of Mobile Nodes (MNs) that are closely located and want to exchange data are organized into a cluster. Each cluster operates in a different frequency channel to avoid interference with neighbouring clusters. Through the use of power control, MNs limit their transmissions in shorter range. Thus, the network is capable of accommodating more users within the same area and transmissions inside a cluster can achieve higher bit rates.

Many of the existing wireless systems operate already in a clustered fashion. For example, in Bluetooth data communication is performed through piconets interconnected with each other to form scatternets. Other systems, such as CANA [1] and CAMA [2] also operate in a clustered way, while at the same time future UWB systems, are believed to adopt a similar clustered type of architecture. The authors in [3] also envision heterogeneous ad hoc networks (using 802.11, Bluetooth and GPRS technologies) that are formed under the central supervision of a cellular network infrastructure. Therefore, it is obvious that clustered

networks will have a dominant position in the next generation networks, constantly expanding and widely deployed as local and metro access solutions.

Yet, despite the relative advantages that the clustered network architecture provides, from the security point of view it introduces several vulnerabilities and flaws that may jeopardize the overall network operation. The fact that in clustered Mobile Ad hoc Networks (MANETs) MNs move, communicate and exchange data in a dynamic environment, where connectivity constantly varies and strict power limitations exist, gives rise to the development of uncooperative behaviour on behalf of the nodes. In clustered ad hoc networks it is quite common that some nodes, urged by selfish motives in order not to spend power and save valuable energy resources, refuse to forward packets. In some other occasions, in order to deliberately disrupt the normal network operation, nodes may exhibit malicious behaviour by modifying on purpose the content of packets. Therefore, in clustered MANETs, more than any other type of networks, cooperation between MNs is essential for the proper and efficient network operation.

In this paper, we develop a new approach for monitoring the forwarding behaviour of nodes in heterogeneous networks with clustered pure ad hoc extensions. In this way, we are able to rate their behaviour and take actions against the emergence of any type of misbehaviour. Special emphasis is given on the prompt detection of a packet drop, modification or misroute. Our method is using cumulative, end-to-end authenticated acknowledgements (ACKs) based on the TESLA [4],[5] broadcast authentication protocol. We also make use of alarms generated by the MNs and propagated to the source, each time a MN detects some type of protocol misbehaviour. The key features of the proposed scheme rely on its simplicity and the limited overhead that it imposes to the network in terms of exchanged messages and processing at the nodes. Furthermore, it is free from any assumptions regarding the trustworthiness of the MNs, (source and destination included) and can be used in real time, in the sense that the source can identify a potential misbehaviour at some point in the transmission path dynamically, without waiting for the packet to reach at the destination in order to get the feedback from all intermediate nodes. The proposed scheme can be applied simultaneously by all nodes in a transmission path and not only the source. In this way, all intermediate nodes that get affected by the potential misbehaviour of a node, can detect the misbehaviour and use this information to rate the conformance of this node to the network functions.

The rest of the paper is organized as follows: In Section 2 we present related work. In Section 3 a brief description of the security issues in a clustered ad hoc network architecture is provided. In Section 4 we present and describe analytically the proposed detection mechanism, based on cumulative end-to-end authenticated acknowledgments, authenticated source alarms and the TESLA protocol. In Section 5 we briefly

discuss the use of reputation ratings to enforce the cooperation of individual nodes. We conclude the paper in Section 6.

## 2. Related work

In most reputation schemes for single channel MANETs, MNs continuously listen to the wireless channel to make sure that each packet they send to their neighbours gets forwarded without being altered. On the other hand, when the forwarding MN sends and receives packets on different channels, there is no single MN that can police the forwarding node. This situation is very common in wired networks where routers cannot use common channel monitoring to rate their neighbours. In [6] a method for detecting routers that drop or misroute packets based on the conservation of flow principle is developed. This method is not intended to detect malicious routers that alter the contents of packets. To achieve this, authenticated ACKs must be used. For example, the authors in [6] use such a scheme to allow the source of a packet to detect a *faulty link* in a route. The term *faulty link* indicates a link for which at least one of its two edges or the link itself is faulty, i.e., fails to correctly execute the protocol. The scheme uses source routing, packet authentication, destination (end-to-end) ACKs, timeouts and fault announcements (FAs). Packets (including ACKs and FAs) must be authenticated in such a way that intermediate nodes can verify the originator of the packet but be unable to impersonate him (in the following, we will use the term *authenticated packet* to mean just that). The most straightforward way to achieve this is with digital signatures. However, digitally signing each ACK using asymmetric cryptography would put a tremendous overhead on the network, since asymmetric cryptography (even the lightweight Elliptic Curve algorithms) is very expensive computationally. For this reason, the authors in [7] use a symmetric cryptography protocol (revised in [8]) that can be used to authenticate packets sent from source to destination as well as ACKs and FAs sent from the destination or intermediate nodes back to the source. The correctness of this protocol is based on the assumption that the source is trusted. Hence, the information gathered about faulty links in a route is valid only at the source node and used only by this node in future routing decisions. In contrast, the use of asymmetric digital signatures would permit all nodes in the path to gain accurate information about faulty links.

Nevertheless, assessing and rating the forwarding behaviour of a node in mobile wireless systems is an inherently complex problem. This is ought to the fact that unsuccessful reception of a packet by the intended destination does not necessarily denote a selfish or malicious behaviour. Packet drop could easily be a result of a link failure due to mobility or bad channel condition. In [9], we have used the TESLA protocol to detect node misbehaviour in a centralized network of clusters. In this setting, each node maintains two

counters for forwarded and unauthenticated packets respectively and uses authenticated ACKs from neighbouring nodes to update these counters and periodically send their values to a trusted Central Authority (CA). This, allows the CA to have a general overview of each node's forwarding behaviour, identify potential misbehaving nodes and take measures against them. TESLA is a broadcast authentication protocol which relies on symmetric key cryptography, loose clock synchronization, hash chains and delayed key disclosure. Compared to digital signatures TESLA trades in bandwidth, delay and buffer space overheads for a significant gain in computational performance. Packets cannot be authenticated immediately, but have to be stored in a buffer until their associated key is disclosed (alas, this opens the door to DoS attacks aiming at overflowing this buffer<sup>1</sup>). Furthermore, a node is supposed to forward unauthenticated packets to their destination (and waste bandwidth in case these packets prove to be forged later on). Albeit these drawbacks, TESLA can be used to authenticate packets and ACKs in a way equivalent to digital signatures in the sense that information about faulty links in the path obtained by intermediate nodes does not rely on the trustworthiness of the source.

### 3. Security issues in clustered Mobile Ad hoc Networks

As the name suggests, in a clustered MANET, the network is organized in clusters of nodes. Each cluster operates in a different frequency channel to avoid interference with neighbouring clusters. Communication between mobile nodes that belong to different clusters is achieved with the help of Forwarding Nodes (FNs). FNs are nodes which belong simultaneously to two adjacent clusters and serve as bridges to forward data packets among them. An FN is able to communicate in both communication channels. The decisions about cluster formation can be made either on an ad hoc basis or via the use of a central controller.

Although a legitimate node can easily verify whether a packet has been successfully forwarded or not when all transmissions take place in the same frequency channel, this task is not so trivial when the packet forwarding is done through a different channel. For instance, in heterogeneous networks a node can receive a packet from its neighbour through an 802.11 channel and forward it using 802.16.

In most cases, MNs forming a MANET are not necessarily trustworthy. The fact that a MN gets authenticated during its association with an Access Point (AP) or other MNs, does not necessarily guarantee that it will always behave correctly and execute properly all network functions. On the contrary, in clustered MANETs, nodes do have many reasons to exhibit

uncooperative behaviour both in the data transmission and in the cluster formation phase. Strict power limitations and constant need to save energy discourage nodes from handling data traffic that it is not originated from or destined to them. This selfish type of misbehaviour results to dropping of packets that use them as relay nodes to reach the destination. Additionally, malicious nodes could try to disrupt network operation by misrouting or modifying packets or by providing false connectivity information. In both cases the result is the same; the network performance deteriorates, the throughput suffers severely due to the drop of packets and connections may face arbitrarily long end-to-end delays.

In this paper, we present an end-to-end security mechanism for detecting non-cooperative behaviour during multi-hop data transmission in clustered MANETs. An insight on how to deal with the problem of misbehaviour in the cluster formation phase is given in [10]. To this end, we introduce the combined use of cumulative end-to-end authenticated *ACKs* and authenticated alarms explicitly sent by intermediate nodes to the source, to notify it of possible misbehaviour in the network. Packet authentication is based on the TESLA protocol. The types of misbehaviour that the proposed scheme deals with include packet dropping, modification and misrouting and attempts to break the TESLA protocol by manipulating the TESLA keys or generating non-valid message authentication codes. The proposed protocol works in the context of a source routing scheme, where routes are predetermined by the source. The goal of the protocol is that the source will be always able to detect links in which either of the two nodes may have misbehaved. An important aspect of the protocol is that not only the source, but also all the nodes in a transmission path can identify misbehaving links and maintain this information in case they act as source nodes at some time in the future. The correct execution of the protocol does not rely on the trustworthiness of any node, in the sense that if misbehaviour takes place at some point in the path, this will be somehow reflected back to the source. Of course, since the collected information will be processed and used by the source, there is no reason for the source node not to act according to the protocol.

### 4. Cumulative ed-to-end authenticated ACKs with authenticated source alarms in clustered MANETs

In order to describe the exact protocol for the detection of a misbehaving link, consider a data transmission path  $n_0, n_1, \dots, n_m$ . We assume that routing paths have been determined in an earlier stage through the use of some secure source routing protocol (e.g., ARIADNE[11], the secure version of DSR). Let

<sup>1</sup> This attack is not possible if TESLA is used for ACK authentication only, since ACKs that do not correspond to a transmitted packet can be immediately dropped.

us denote by  $d_{i,j}$  an estimate of the delay<sup>2</sup> of transmitting a packet from  $n_i$  to  $n_j$ ,  $0 \leq i, j \leq m$ . We also denote by  $d_i$  the Round Trip Time (RTT) from the moment when  $n_i$  transmits/forwards a RREQ until it receives the corresponding RREP. We assume that each node  $n_i$  in the path can estimate<sup>3</sup>  $d_{i,i+1} = 0.5 \cdot (d_i - d_{i+1})$ . The calculated  $d_{i,i+1}$ 's should be then communicated to all other nodes in the path.

In order to use TESLA for ACK authentication, the destination  $m$ , upon receiving a RREQ packet, chooses a random initial key  $K_N^m$  and generates a one-way key chain by repeatedly computing a one-way hash function  $H(\cdot)$   $N$  times (i.e.,  $K_{N-1}^m = H(K_N^m), \dots, K_{k-1}^m = H(K_k^m), \dots, K_0^m = H(K_1^m)$ ).

The destination then communicates  $K_0^m$  to all nodes in the path, either by including this information in the RREP message or by issuing separate control packets. The same procedure is also followed by the intermediate nodes in the path, communicating their own TESLA keys  $K_0^i$  ( $0 < i < m$ ) upstream (i.e., towards the source). By default, there is no reason for the source to issue its own TESLA keys.

Each transmitted packet is assigned by the source a unique serial number  $SN$ . The source sends the packet (SN included) downstream without any form of authentication. Upon reception of a packet  $p$  by an intermediate node  $n_i$  or the destination node  $n_m$ , the node computes a Message Authentication Code (MAC)  $M$  based on the received packet and the ID of the node (i.e.,  $M = MAC_{K_k^i}(p, i)$ ), using one of its own TESLA symmetric keys  $K_k^i$ ,  $0 < k \leq N$ . The node then sends back to the source an ACK consisting of the node ID  $i$ , the ID of the TESLA key  $k$ , the SN of the received packet and its associated MAC  $M$ , i.e.  $ACK(i, SN) = \langle i, k, SN, MAC_{K_k^i}(p, i) \rangle$ . A packet

containing  $k$  and  $K_k^i$  is sent from  $n_i$  to the source according to the TESLA protocol based on the key disclosure schedule ([4]). Confirmation of the received MAC is only possible after the reception of the associated key. The disclosure of the key can happen only after the estimated time for the propagation of the ACK to the source has expired. Instead of a pessimistic upper bound to the end-to-end network delay (as in the original TESLA), node  $n_i$  can use the time information

$\tau_i = \sum_{j=0}^{i-1} d_{j,j+1}$  that it has already obtained during the route discovery phase and the establishment of the path.

<sup>2</sup> We assume fixed delays as in a TDMA wireless MAC. If CSMA is used maximum delays have to be used instead.

<sup>3</sup> For simplicity we assume symmetric delays, i.e.,  $d_{i,i+1} = d_{i+1,i}$ .

The key publication interval should be kept relatively small so that only a few packets are authenticated with the same key.

The source node  $n_0$  starts  $m$  timers  $t_0^{j,SN}$  ( $0 < j \leq m$ ) for each transmitted packet from the source to the destination, one for each hop in the path. The purpose of a timer  $t_0^{j,SN}$  is to count the time that has elapsed from the generation of a packet with sequence number  $SN$  that has not yet been acknowledged by node  $j$  in the path. Each timer  $t_0^{j,SN}$  expires at

$2 \sum_{i=0}^{j-1} d_{i,i+1}$ . Similarly, each intermediate node  $n_i$  ( $0 < i < m$ ) upon forwarding a packet starts  $m-i-1$  timers  $t_i^{j,SN}$  ( $i < j \leq m$ ), one for each of the remaining hops in the path. Each of these timers is scheduled to expire at  $2 \sum_{k=i}^{j-1} d_{k,k+1}$ .

The expiration of any timer indicates dropped or excessively delayed data packets or ACKs. The expiration of a timer at node  $n_i$  due to a not received yet ACK for a packet with sequence number  $SN$  from node  $n_j$ , generates a Source Alarm  $SA(i, j, SN)$ ,  $0 \leq i \leq m, 0 < j \leq m$ . This source alarm  $SA(i, j, SN)$  is sent upstream from node  $n_i$  to the source  $n_0$ , notifying the source that this particular data packet has not been acknowledged by node  $n_j$ .  $SAs$  get forwarded to the source by intermediate nodes. An intermediate node  $n_k$  receiving a  $SA(i, j, SN)$ , ( $k < i < j$ ) is obliged to generate its own  $SA(k, j, SN)$  as well, in case the corresponding timer  $t_k^{j,SN}$  expires. In the ideal case, where the delay estimations were accurate and the synchronization was perfect, the reception of  $SA(i, j, SN)$  and the generation of  $SA(k, j, SN)$  should be taking place at the same time. We indicate the relative time difference of these two events in a node by  $\delta$  and we assume that it is relatively small. If node  $n_k$  generates  $SA(k, j, SN)$  and doesn't receive a valid  $SA(i, j, SN)$  then it has to generate  $SA(k, i, SN)$  as well.

In order for the source to be able to verify the authenticity of the received  $SA$ , the transmitted  $SA$  also includes a MAC based on the received packet and the ID's of the nodes  $n_i$  and  $n_j$ , i.e.,  $SA(i, j, SN) = \langle i, j, k, SN, MAC_{K_k^i}(p, i, j) \rangle$ .

Well behaving nodes are supposed to forward valid packets, ACKs and SAs unaltered to the appropriate node. If a node (including the destination) receives a packet whose SN has been already encountered, it has to drop this packet. ACKs and SAs should be also dropped if they have been already seen, if they correspond to packets that have not been received yet or if they indicate a key that might have been already published

(according to the TESLA protocol). Because packets containing keys can also be dropped or excessively delayed, another timer  $t_i^{j,K_k}$  ( $0 < k \leq N$ ) should expire when  $n_i$  has not received ahead of time the key  $K_k^j$  by  $n_j$ . The expiration time should be set at  $t_k + (j-i) \cdot \Delta + \sum_{k=i}^{j-1} d_{j,j+1}$  where  $t_k$  is the scheduled time for  $K_k^j$  to be published and  $\Delta$  a time synchronization slack, allowing for a small ACK transmission jitter in each hop of the path. If  $t_i^{j,K_k}$  expires then all packets from node  $n_j$  whose ACKs are associated with  $K_k^j$  are considered unauthenticated. Upon reception of a packet containing a key by node  $n_i$ , if  $t_i^{j,K_k}$  has expired, the packet is dropped. Otherwise the key is checked for validity using the one way hash function  $H(\cdot)$ . If it is valid, then the MAC in the associated ACKs is checked for correctness. A received ACK is considered valid only when both the TESLA key  $K_k^j$  is fresh and valid and when the MAC of the associated packet received by a node is correct. In any other case the packets are considered unauthenticated and a source alarm  $SA$  is generated at node  $n_i$  for each packet.

A schematic for the conceptual operation of the proposed scheme is shown in Fig. 2. Dashed lines indicate virtual links, in the sense that  $ACKs$  and  $SAs$  are propagated to the source via the intermediate hops.

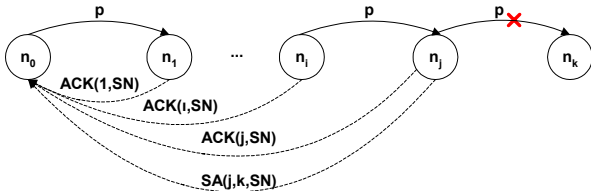


Fig. 2: The proposed scheme with end-to-end authenticated ACKs and SAs

Before we move on to prove the validity and correctness of the proposed scheme we introduce the following terms:

**Definition 1:** A *suspicious link* is a link that either of its two adjacent nodes may have misbehaved or the packet was lost due to a link failure.

**Definition 2:** A *suspicious path* is a path that has at least one suspicious link.

A misbehaving node may be either selfish or malicious. In the first case, data packets,  $ACKs$  or  $SAs$  are not forwarded, while in the latter data packets,  $ACKs$  or  $SAs$  are modified before they get forwarded.

The purpose of the protocol is to identify suspicious links in the path according to the following propositions:

**Proposition 1:** If the source node has received and verified all the  $ACK(i,SN)$  ( $0 < i \leq m$ ) associated with a specific data packet with sequence number  $SN$ , then packet transmission is considered successful; otherwise at least one  $SA$  arrives at the source (it might have been generated by the source itself) and if  $SA(i^*,j^*,SN)$  is the  $SA$  which satisfies  $\max_i \left( \min_j SA(i,j,SN) \right)$ , then link  $\langle n_{i^*}, n_{i^*+1} \rangle$  is a suspicious link.

**Proposition 2:** All legitimate nodes upstream of a suspicious path can identify a suspicious link in the path; if  $SA(i^*,j^*,SN)$  is the  $SA$  which satisfies  $\max_i \left( \min_j SA(i,j,SN) \right)$ , then link  $\langle n_{i^*}, n_{i^*+1} \rangle$  is a suspicious link.

To prove the correctness of the propositions we first present and prove the following properties:

**Property 1:** The reception of a valid  $SA(i,j,SN)$  indicates that the path  $\langle n_i, n_{i+1}, n_{i+2}, \dots, n_j \rangle$  is a suspicious path.

**Proof:** A received  $SA(i,j,SN)$  indicates one of the two following possible scenarios: i) One of the nodes in the path  $\langle n_{i+1}, n_{i+2}, \dots, n_j \rangle$  is a misbehaving node, or ii) node  $n_i$  has deliberately generated a fake  $SA$ . In either case, a misbehaving node lies in the path  $\langle n_i, n_{i+1}, n_{i+2}, \dots, n_j \rangle$ . ■

**Property 2:** The reception by the source of both a valid  $ACK(i,SN)$  and a valid  $SA(i,j,SN)$  indicate that the packet has been successfully received by node  $n_i$ .

**Proof:** The MAC of a valid ACK and SA has to be computed on the actual data packet; therefore the packet has arrived at  $n_i$  unaltered, i.e., all intermediate nodes  $n_k$  ( $0 < k < i$ ) have successfully received and forwarded the packet. ■

**Property 3:** A node  $n_i$  cannot generate a valid  $SA(j,k,SN)$ ,  $i < j < k$ , even after receiving  $ACK(j,SN)$  and  $ACK(k,SN)$ .

**Proof:** With cryptographically secure MACs the knowledge of  $MAC_{K_i^j}(p,j)$  and/or  $MAC_{K_i^k}(p,k)$  is not enough for an attacker to generate a valid  $MAC_{K_i^j}(p,j,k)$  without knowing  $K_i^j$ . ■

**Proof of Proposition 1:** Property 2 implies that if the source node has received and verified all the  $ACK(i,SN)$  ( $0 < i \leq m$ ), then the packet successfully reached its destination. By default, if this doesn't happen at least one  $SA$  arrives at the source. Let us prove by contradiction that the link  $\langle n_{i^*}, n_{i^*+1} \rangle$  defined in Proposition 1 is indeed a suspicious link.

If the link  $\langle n_{i^*}, n_{i^*+1} \rangle$  is not suspicious then both the nodes  $n_{i^*}$  and  $n_{i^*+1}$  and the link  $\langle n_{i^*}, n_{i^*+1} \rangle$ , have adhered to the protocol. Now, since  $SA(i^*, j^*, SN)$  was received by the source, Property 1 implies that the path  $\langle n_{i^*}, n_{j^*} \rangle$ , is suspicious and combined with the hypothesis we conclude that  $j^* > i^*+1$  and that the path  $\langle n_{i^*+2}, n_{j^*} \rangle$  and/or the link  $\langle n_{i^*+1}, n_{i^*+2} \rangle$  are suspicious. But then, (the legitimate) node  $n_{i^*+1}$  should have generated  $SA(i^*+1, j^*, SN)$  as well. Therefore, since  $i^*$  is the maximum index of nodes that generated a SA about  $n_{j^*}$  which was received by the source,  $SA(i^*+1, j^*, SN)$  must have been dropped upstream of  $n_{i^*}$ . In that case however, the source has generated  $SA(0, j^*, SN)$  and hasn't received  $SA(i^*+1, j^*, SN)$  hence (according to the protocol) it must have generated  $SA(0, i^*+1, SN)$  as well. But,  $i^*+1 < j^*$  which contradicts the initial hypothesis that  $j^*$  is the minimum suspicious node index in all SA's received by the source. ■

A proof for Proposition 2 can follow along the same lines and is omitted due to space limitations. The two propositions suggest that the combined use of cumulative end-to-end authenticated ACKs and SAs uniquely identifies at least one suspicious link in the path. Note, that under certain conditions multiple suspicious links could also be identified by different nodes in the path.

## 5. Employing reputation ratings to reinforce cooperation

So far we have explained how common channel monitoring and authenticated end-to-end ACKs and SAs can be used to identify suspicious links in a path. However, in the latter case where the source or a legitimate intermediate node has identified the misbehaving link, it has no way of knowing which one of the edge nodes in the link has misbehaved. Of course, the source may find an alternative route to the destination to "fuse" the suspicious link; similarly, intermediate hops, when acting as source nodes in the future, may avoid routing their traffic through that link. The ultimate goal for legitimate nodes is to build and obtain node reputation ratings, by constructively processing the link reputation ratings. In this way, every node in the network will be capable of inferring each other node's forwarding behaviour. Link and node reputation ratings can be jointly used by an appropriately designed reputation mechanism to rate nodes' conformance to network's functions.

## 6. Conclusions

This paper has introduced a security mechanism for detecting misbehaviour in heterogeneous networks with clustered mobile ad hoc extensions. When compared to single channel MANETs, clustered MANETs have the disadvantage of a clustered architecture where neighbouring clusters use different communication channels to avoid interference. This complicates the task of rating the forwarding service provided by a node's neighbours. End-to-end authenticated ACKs can be used to resolve this problem. However, using an asymmetric cryptography scheme for authenticating all ACKs is very expensive computationally. In this paper, we proposed a solution that incorporates end-to-end authenticated ACKs and source alarms based on the TESLA symmetric key broadcast authentication protocol. The proposed scheme guarantees that in case of misbehaviour, at least one suspicious link will be identified by all legitimate nodes upstream of the suspicious link. Furthermore, the fact that DoS attacks, which is the main weakness of TESLA, is not possible when TESLA is used to authenticate ACKs and SAs only, makes the proposed scheme a dependable and efficient solution to rating packet forwarding services in clustered ad hoc mobile networks.

## 7. References

- [1] K. Oikonomou et al., "A Centralized Ad-Hoc Network Architecture (CANA) Based on Enhanced HiperLAN/2," IEEE PIMRC 2003, Beijing, China, September 7-10, 2003.
- [2] B. Bhargava et al. "Integrating Heterogeneous Wireless Technologies: A Cellular Aided Mobile Ad Hoc Network (CAMA)," Mobile Networks and Applications 9, 393-408, 2004 Kluwer Academic Publishers.
- [3] M. Danzeisen, et al., "Heterogeneous Network Establishment Assisted by Cellular Operators", 5<sup>th</sup> IFIP TC6 Int'l Conference on Mobile and Wireless Communication Networks, Singapore, October 2003.
- [4] A. Perrig, et al., "Efficient and secure source authentication for multicast," Network & Distributed System Security Symposium, NDSS '01, San Diego, CA, 2001.
- [5] A. Perrig, et al., "Efficient Authentication and Signing of Multicast Streams over Lossy Channels," 2000 IEEE Symposium on Security and Privacy (S&P 2000), May 2000.
- [6] K. Bradley, et al., "Detecting Disruptive Routers: A Distributed Network Monitoring Approach", IEEE Symposium on Security and Privacy, May 1998.
- [7] I. Avramopoulos, et al., "Highly Secure and Efficient Routing," INFOCOM 2004, Hong Kong, March 2004.
- [8] Amendment to: "Highly Secure and Efficient Routing," [lambda.cs.yale.edu/~arvind/papers/amendment.pdf](http://lambda.cs.yale.edu/~arvind/papers/amendment.pdf)
- [9] S. Vassilaras, D. Vogiatzis, G. Yovanof, "Misbehaviour Detection in Clustered Ad-hoc Networks with Central Control", ITCC 2005, Las Vegas, USA, April 2005.
- [10] S. Vassilaras, D. Vogiatzis, G. Yovanof, "Cooperation Enforcement in Mobile Ad hoc Networks with Centralized Supervision," European Wireless 2005, Nicosia, Cyprus, April 2005.
- [11] Hu et al., "Ariadne: A secure On-Demand Routing Protocol for Ad Hoc Networks", MOBICOM 2002.