DESIGN AND IMPLEMENTATION OF A TRUST-AWARE ROUTING PROTOCOL FOR LARGE WSNS

Theodore Zahariadis[1], Helen Leligou[1], Panagiotis Karkazis[5],

Panagiotis Trakadas[2], Ioannis Papaefstathiou[5],

Charalambos Vangelatos[3], Lionel Besson[4],

[1] Technological Educational Institute of Chalkis, Greece
{zahariad, leligou}@teihal.gr
[2] Hellenic Authority for Communications Security and Privacy (ADAE), Greece
trakadasp@adae.gr
[3] Hellenic Aerospace Industry S.A., Greece
VANGELATOS.Charalampos@haicorp.com
[4] Thales Communications, France
Lionel.BESSON@fr.thalesgroup.com
[5] Technical University of Crete, Greece
pkarkazis@isc.tuc.gr, ygp@ece.tuc.gr

ABSTRACT

The domain of Wireless Sensor Networks (WSNs) applications is increasing widely over the last few years. As this new type of networking is characterized by severely constrained node resources, limited network resources and the requirement to operate in an ad hoc manner, implementing security functionality to protect against adversary nodes becomes a challenging task. In this paper, we present a trust-aware, location-based routing protocol which protects the WSN against routing attacks, and also supports large-scale WSNs deployments. The proposed solution has been shown to efficiently detect and avoid malicious nodes and has been implemented in state-of-the-art sensor nodes for a real-life test-bed. This work focuses on the assessment of the implementation cost and on the lessons learned through the design, implementation and validation process.

KEYWORDS

Wireless sensor networks, trust management, secure routing, implementation cost

1. INTRODUCTION

Wireless Sensor Networks (WSNs) offer solutions which cover a wide range of application domains, including homeland security and personal healthcare, building and urban surveillance, industrial operations and environmental monitoring ([1], [2], [3], [4]). Their increasing penetration mainly stems from three important advantages: wireless operation, low cost and easy installation/ self-organisation. These advantages however, inherently introduce security issues [5].

The wireless operation of WSNs renders them vulnerable to privacy attacks while the nodes' low cost is tightly related to low capabilities in terms of processing, memory and energy resources, which limits the functionality that can be implemented to defend against the security

DOI : 10.5121/ijnsa.2010.2304

attacks. Thus, mature security solutions designed for legacy wired and wireless networks are unfortunately not applicable in WSNs. Another intricacy further complicating the security problem is that the nodes need to cooperate in order to accomplish certain networking tasks (like e.g. routing) to meet the random deployment requirement, introducing additional vulnerabilities.

Numerous security attacks have been presented in the literature ([6], [7]) with a significant subset targeting the routing process [8]. Once an adversary node manages to participate in the network, it can damage the routing process by simply dropping the packets it receives for forwarding, i.e. denying to sincerely cooperate in the routing procedure. Another easily implementable attack is packet modification. A taxonomy of routing attacks can be found in [9]. To defend against the majority of routing attacks, an approach borrowed from the human society has been proposed [10]: nodes monitor the behavior of their neighbours in order to evaluate their trustworthiness, regarding specific behaviour aspects called trust metrics. Although a plethora of such models has been proposed and shown to efficiently mitigate routing attacks, trust models are themselves vulnerable to specific attacks [11]. The need to defend against these attacks further increases the complexity of the functionality that needs to be implemented on the sensor nodes for security purposes.

Although the design of mechanisms to enhance security at all layers of the networking protocol stack has attracted the interest of the research community (e.g. [12], [13]), very limited implementation effort has been reported. In [14], the implementation of a link-layer security architecture is presented, while in [15] experience regarding the implementation of hash-based encryption schemes in TinyOS operated sensor nodes is reported. In [16], the efficiency of a set of routing protocols is compared based on real test-bed experiments. Finally, in [17] very limited information regarding the implementation of a trust model is provided.

In this paper, we present results and experience gained through the implementation of a location-based trust-aware routing solution called Ambient Trust Sensor Routing (ATSR). It incorporates a distributed trust model which relies on both direct and indirect trust information to protect the WSN from a wide set of routing and trust-related attacks. Our focus here is not on the design of a radically new trust model; instead, we concentrate on the implementation of ATSR and discuss the lessons learned through the design and implementation procedure in an attempt to balance the achieved performance benefits with the introduced implementation cost. Our target is to assess the node resources required to realize an efficient trust model and to reach guidelines for prospective designers and implementers of trust models used for routing purposes.

The rest of the paper is organized as follows: we first present the related work on trust-aware WSN routing and in section 3 the designed ATSR protocol while in section 4 we evaluate its performance based on results from computer simulations and analyse the performance benefits brought by each design choice. In section 5 we present the implementation architecture and in section 6 we analyse the implementation cost. In section 7 we discuss the lessons learned and draw guidelines while finally, in section 8, conclusions are reached.

2. RELATED WORK

Trust-based enhancements on the routing protocols for WSN have been widely addressed in the literature. The most important research results in this direction include:

Trusted AODV: The well-known AODV routing protocol has been extended by Xiaoqi Li et. al. [18] to perform routing by taking into account trust metrics. A trust recommendation mechanism is first introduced and then the routing decision rules of AODV are modified to take into account trust. Of particular interest is that a set of policies is derived for a node to update its opinions towards others since, it is necessary to design a trust information exchange mechanism

when applying the trust models into network applications. More specifically, three procedures (Trust Recommendation, Trust Judgment, Trust Update) are defined as well as the accompanying Route Table Extension, Routing Messages Extensions, Trusted Routing Discovery.

Trust-aware Dynamic Source Routing: To secure the Dynamic Source Routing (DSR) protocol, a mechanism involving the "watchdog" and "pathrater" modules has been designed and incorporated in the routing protocol [19]. This scheme is applicable to routing protocols where the source defines the route to be followed by the packets. The mechanism basically consists of two components: Watchdog and Pathrater. The Watchdog is responsible for detecting selfish nodes that do not forward packets. To do so, each node in the network buffers every transmitted packet for a limited period. During this time, each node places its wireless interface into promiscuous mode in order to overhear whether the next node has forwarded the packet or not. The Pathrater assigns different ratings to the nodes based upon the feedback that it receives from the Watchdog. These rating are then used to select routes consisting of nodes with the highest forwarding rate. The dynamic source routing (DSR) protocol that has been proposed to discover routes in wireless ad-hoc networks has been extended by Pirzada et. al [20] to also take into account the trust levels (reputations) of the nodes. Exactly as happens in trusted AODV, it improves the achieved security although it cannot deal with all the possible attacks.

CONFIDANT (Cooperation Of Nodes, Fairness In Dynamic Ad-hoc NeTworks) [21] adds a trust manager and a reputation system to the Watchdog and Pathrater scheme. The trust manager evaluates the events reported by the Watchdog (monitor in this case) and issues alarms to warn other nodes regarding malicious nodes. The alarm recipients are maintained in a friends-list which is configured through a user-to-user authentication mechanism. The reputation system maintains a black-list of nodes at each node and shares them with nodes in the friends-list. The CONFIDANT protocol implements a punishment based scheme by not forwarding packets of nodes whose trust level drops below a certain threshold.

CORE (COllaborative REputation) [22] is similar to CONFIDANT, however it employs a complicated reputation exchange mechanism. CORE divides the reputation of a node into three distinct components: Subjective Reputation, which is observed through own observations; Indirect Reputation, which is a positive report by another node; and Functional Reputation, which is based upon behaviour monitored during a specific task. These reputations are weighted for a combined reputation value.

Trusted GPSR: In [23] the greedy perimeter stateless routing (GPSR) is enhanced so as to take into account node trust levels. To do so, each time a node sends out a packet it waits until it overhears its neighboring node forwarding the packet. Based on this information (correct and prompt forwarding) it maintains a trust value for its neighbors. This information is then taken into account in the routing decisions.

TRANS (Trust Routing for Location aware Sensor Networks): TRANS is a routing protocol that selects routes among nodes based mainly on trust information and not on hop-count or other metric to avoid insecure locations [24]. The protocol relies on the assumptions that the sensors know their (approximate) locations and that geographic routing (e.g., GPSR) is used. For TRANS, a trusted neighbor is a sensor that can decrypt the request and has enough trust value (based on forwarding history as recorded by the sink and other intermediate nodes). A sink sends a message only to its trusted neighbors (i.e. its trust value is higher than specified trust threshold) for the destined location. Those neighbors correspondingly forward the packet to their trusted neighbors that have the nearest location to destination. Thus, the packet reaches the destination along a path of trusted sensors. An important feature of TRANS is that the blacklisting is distributed (or embedded in data packet) by the sink. However, this comes at the

assumption that the sink will not be compromised. The sink identifies misbehavior (by observing replies), probes potential misbehaving locations, and isolates insecure locations. After excessive packet drops, the sink initiates a search for insecure locations along the path. On discovery of such location the sink records the insecure location and advertises this information to the neighboring nodes.

SPINS: A suite of security protocols optimized for sensor networks (SPINS) has been designed [25] to provide data confidentiality, two-party data authentication, and evidence of data freshness. It involves two secure building blocks: SNEP and μTESLA. SNEP introduces a small overhead of 8 bytes, it maintains a counter but no counter values are exchanged (protecting the network from eavesdropping) and achieves semantic security. μTesla provides authentication for data broadcast. Emphasis has been placed on the limited processing and memory resources available in sensor networks environment. SPINS claim to provide trusted routing ensuring data authentication and confidentiality. However, it does not deal with Denial of Service Attacks or compromised nodes. It only ensures that a compromised node does not reveal all the keys of the network.

ARIADNE: A secure on-demand ad hoc network routing protocol, which prevents attackers or compromised nodes from tampering with uncompromised routes consisting of uncompromised nodes, and also prevents a large number of types of Denial-of-Service attacks is Ariadne which is presented in [26]. Ariadne is efficient, using only highly efficient symmetric cryptographic primitives and uses per-hop hashing functions. It also assumes the use of TESLA and MAC authentication mechanisms.

Trusted cluster head election: Scalability is a requirement for the wide deployment of sensor networks and can be achieved through clustering architectures. However, when it comes to security and trust, clustering architectures present the intricacy that the dynamic election of the cluster head has to be performed in a highly secure way, i.e. security should be higher in this procedure since the election of a compromised or malicious node as cluster head the network will collapse. In [27], a reputation metric based on successful vs. unsuccessful network events is communicated between the cluster members and the current cluster head, to drive the election of the next cluter head. This procedure is triggered by the current cluster head when certain threshold (e.g. battery levels) are reached. The new cluster head is elected based on the majority of votes and to ensure that it is not a compromised node, it is challenged (through a key exchange procedure). As the research work on routing attacks in WSNs has continued, trust models fitting different network architectures and different levels of security requirements have been proposed. A more elaborate review of the trust models is provided in [28] and [29].

3. THE ATSR DESIGN

Secure routing mandates that a message travels to the destination through benevolent nodes (avoiding malicious ones). In ATSR, a fully distributed trust management system is realised in order to evaluate the reliability of the nodes. Following this approach, nodes monitor the behaviour of their neighbours, regarding specific behaviour aspects (trust metrics) and can thus reach a direct trust value per neighbour. For example, a node may monitor whether its neighbour forwards the received traffic (forwarding trust metric) evaluating its trustworthiness regarding forwarding. ATSR takes also into account indirect trust information, i.e. each node additionally requests trust information from its neighbours regarding third nodes to accelerate the trust build-up procedure. This indirect trust information is also called reputation. As this process is also vulnerable to specific attacks [11], special measures against them are realized in our protocol. Direct and indirect trust information is combined to reach the Total Trust

information. Finally, the routing decisions are based on geographical information (distance to the base-station) and Total Trust information.

The routing and trust overhead introduced by ATSR includes the Beacon (broadcast) message which is used by each node to periodically announce its location coordinates, node id and remaining energy, the reputation request (multicast) message used to periodically request indirect trust information and the reputation response (unicast) message which is used to provide indirect information as a reply to a reputation request message.

Starting from the direct trust, each neighbour is evaluated based on a set of trust metrics which include:

- ***Packet forwarding:*** To detect nodes that deny to or selectively forward packets, acting in a selfish (malicious or not) manner, each time a source node sends a packet to a neighbour for further forwarding, it enters the promiscuous mode and overhears the wireless medium to check whether the packet was actually forwarded by the selected neighbour.

- ***Network layer Acknowledgements (ACK):*** To detect the successful end-to-end forwarding of the messages (and detect colluding adversaries), we suggest that each source node waits for a network-layer ACK per transmitted message to check whether the message has successfully reached a higher layer node (i.e. the base station). It is stressed that this check is performed only for trust evaluation purposes and does not necessarily trigger any message retransmission.

- ***Packet precision:*** Each time a source node transmits a packet for forwarding and then overhears the wireless medium to ensure that the packet was forwarded, it additionally processes it to check the packet's integrity, i.e. that no unexpected modification has occurred.

- ***Authentication***: The trust management module receives information from other (higher layer) blocks related to the trustworthiness of the neighbours. For example, in case a node may choose among neighbours supporting different authentication mechanisms, the one with better security features should be preferred. Although this is not an event or behaviour aspect monitored by the source node, it is listed here as an input to the trust evaluation system.

- ***Reputation Response:*** To check the sincere execution of the reputation exchange protocol, the node that requests reputation information, calculates for each neighbour the number of received reputation responses divided by the number of times this neighbour was asked for reputation information. This way, nodes that do not cooperate in the execution of the reputation protocol (acting in a selfish manner) are assigned lower trust values and are avoided for forwarding co-operations as a penalty.

- ***Reputation Validation:*** To protect against wrong (either bad or good) reputations being spread around (called hereafter bad-mouthing attack) and conflicting behaviour attacks [11] (i.e. a malicious node behaves differently towards different neighbours in different timespans), each time a reputation response message is received, the received reputations are validated. Each time node A receives a reputation response message from node C regarding node B, it compares it with the trust value node A has calculated for node B (if node A is confident about the direct trust value) and with the reputations provided by other neighbours. If the difference between the received value and the others exceeds a certain threshold, then the node that provided this value is considered malicious and the reputation is considered wrong; otherwise it is a "correct reputation".

- ***Remaining Energy:*** Although the energy level of each neighbour is not a pure trust metric, taking into account the remaining energy level, apart from extending the network lifetime, contributes towards load balancing (partially defending against the traffic analysis attack). In our novel routing protocol, the remaining energy travels piggy-backed in the Beacon message used to indicate the node availability and position.

On each sensor node, a trust repository is used to store trust-related information per neighbour. Dividing the number of successfully forwarded messages to the total number of messages A sent to B for forwarding, a trust value regarding forwarding is reached. Each trust metric targets the detection of one (or more) routing attack(s), as explained in [28] and will also be shown in the performance evaluation section. For example, the first trust metric (forwarding) allows for the detection of black hole and grey hole attackers, i.e. nodes that drop all or part of the received traffic respectively. Low values for this metric reveal black- and grey-hole attackers which can then be avoided for routing purposes. Similarly, comparing successfully performed interactions over the total number of attempted interactions provides a trust value for each of the monitored behaviours (except for energy and authentication) and leads to a verdict regarding the related attack. The remaining energy is expressed as a percentage of the initial energy of the node while the authentication metric is either equal to '1' (for nodes supporting high security schemes) or equal to '0' for no special measures. (Finer granularity is possible depending on the security levels achieved. This metric is actually left as a hook for communication of trust-related information with higher layer.) The values calculated for each metric are then summed up in a weighted manner to form the direct trust value per neighbour, as follows:

$$DT^{i,j} = \sum_{1}^{7} (W_m * T_m^{i,j}) \qquad (1)$$

The weight assigned to each metric represents the importance of detecting and avoiding the related attack since it affects the number of interactions (and thus the time) required for the detection of this attack type. Assigning higher weight value to the forwarding metric results in detection of black and grey-hole attackers after a really small number of cooperation attempts.

Coming to the reputation exchange protocol which is used to accelerate the trust information build-up procedure, in ATSR each node periodically requests reputation information from one randomly selected neighbour per quadrant. The received reputation information is combined with the direct trust value to reach the total trust value (trustworthiness) for each neighbour. In this concept, every node can build a trust relation with its neighbours, based on actions (events) performed by other nodes in the neighbourhood since node A may be informed that node B is malicious by their common neighbours before any direct interaction between nodes A and B. This is useful when a node arrives in a new neighbourhood which is usually the case for mobile nodes. The received reputations are summed up in a weighted manner with the weight representing the relevant trustworthiness of the node that provided it. Also, as a measure towards reducing the effects of bad-mouthing attacks, the indirect trust information provided by a non trusted neighbour is dropped.

Finally, the Total Trust (TT) value for a neighbor j is produced combining direct and indirect trust values in the following formula:

$$TT^{i,j} = C^{i,j} * DT^{i,j} + (1 - C^{i,j}) * IT^{i,j} \qquad (2)$$

where $C^{i,j}$ is the confidence factor which increases with the number of performed interactions.

The presented trust model has been integrated with a location-based routing protocol, which offers significant scalability advantages due to its localized routing decisions. A distance-related metric was defined and calculated per one-hop neighbour. This metric is maximized for the node closest to the destination and is combined with the total trust value in the ATSR Routing Function:

$$RF^{i,j} = W_d * T_d^{i,j} + W_t * TT^{i,j} \qquad (3)$$

Where *Wd* and *Wt* represent the significance of distance and trust criterion respectively with $W_d + W_t = 1$. The one-hop neighbour that maximizes the trust-distance routing function is selected as the next hop node for forwarding. Although it is possible to define a trust threshold and route packets through nodes with trust higher than this threshold, we have opted for balancing trust with distance to avoid node isolation. In the case that no malicious node exists in the network, i.e. the Total Trust is almost equal to 1, the ATSR behaves similarly to the Greedy Perimeter Stateless Routing (GPSR) protocol [30], which forwards the packet to the node closest to the destination.

To sum up, the ATSR solution combines a distributed trust model with a location-based routing approach to offer protection against routing attacks and scalability advantages.

4. PERFORMANCE EVALUATION

To fine tune the ATSR parameters and evaluate its efficiency, we modelled it using the JSIM open simulation platform [31]. Exploiting the JSIM capabilities in supporting simulation of large sensor networks, we performed simulation tests for topologies consisting of up to 1000 nodes.

4.1. Combining trust with distance for routing purposes

We first investigated the impact of the *Wd* on the performance of the presented ATSR protocol which are used to combined trust with distance in the routing cost function. We have run a scenario set for different values of the *Wd* parameter. (It is mentioned that *Wt=1-Wd*.) The results in terms of packet loss ratio in the presence of 50 malicious nodes randomly dropping half of the received packets (i.e. issuing grey-hole attacks) are shown in Figure 2. The best performance (lowest loss ratio) is achieved for *Wd=0.4* which indicates that the distance criterion should be almost equally balanced to trust. When low values are assigned to the distance criterion, packets travel more trusted but longer paths to the destination, risking more hops. The longer paths results in higher mean packet latency in this case, as shown in the corresponding curve in Figure 1. On the contrary, for *Wd* close to 1, trust plays a minor role and the proposed approach behaves similarly to any non trust aware protocol resulting in high loss. The same tendencies were observed for other malicious nodes penetration values.

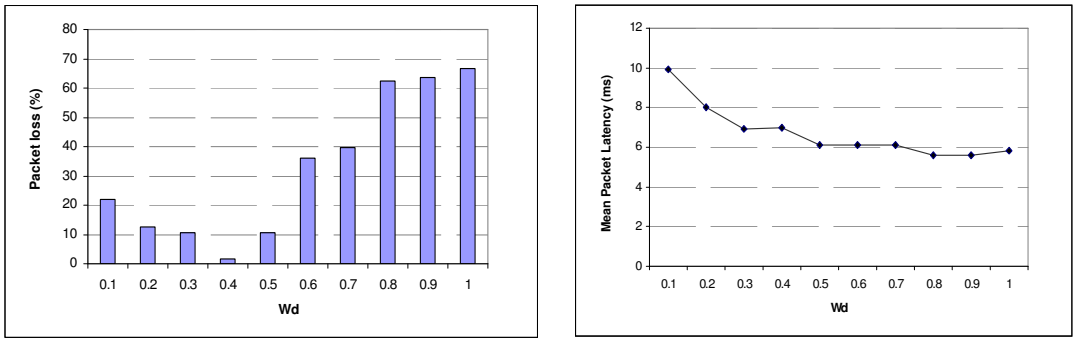

Figure 1: Packet loss and mean packet latency in the presence of 50% malicious (grey-hole) nodes as a function of Wd.

4.2. Balancing the trust metrics

Having selected Wd equal to 0.4, we first validated its efficiency in detecting the attacks and then explored the impact of the weights assigned to the difference trust metric on the performance, since this represents an important design choice.

In a representative scenario which includes three types of attacks, we have set the trust weight metrics equal to 0.2 for forwarding, 0.1 for network acknowledgment, 0.2 for integrity, 0.2 for authentication, 0.1 for reputation response, 0.1 for reputation validation and 0.1 for energy. The scenario was carried out for 1024 fixed-location sensor nodes organised on a symmetric grid and varying numbers of attackers which were placed randomly and issued grey-hole, integrity and authentication attacks with equal probability. The results for the proposed ATSR and for a non trust-aware location based protocol (namely the greedy-perimeter stateless routing, GPSR) in terms of packet loss and mean packet latency are depicted in Figure 2. As expected, the benefits of introducing trust awareness are evident with regard to packet loss (ATSR outperforms GPSR in all cases) with loss lower than 30% observed for 50% malicious nodes in the network. The mean packet latency for those packets that reached the destination is lower for GPSR since it is capable of selecting the shortest route with the lowest number of hops. Instead, ATSR results in higher mean packet latency in all cases, especially when the number of malicious nodes increases, since it finds alternative but "longer" (i.e. involving more intermediate nodes) paths to the destination to avoid malicious nodes. Let us recall here that our main goal is not to present ATSR as a radically new trust model outperforming other research approaches but to show the performance improvements that such an approach brings and then evaluate the corresponding implementation requirements.

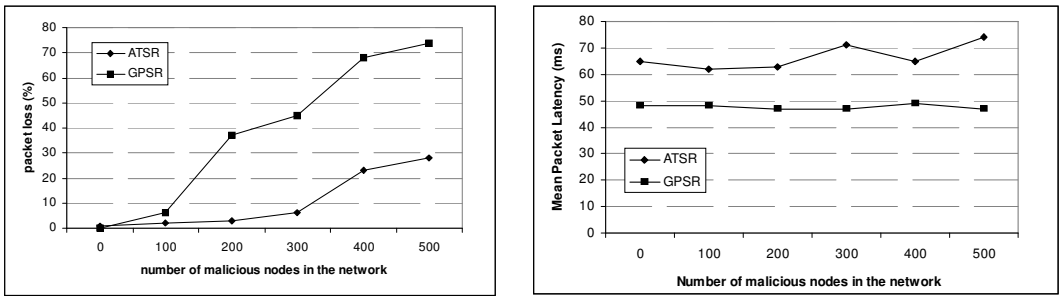

Figure 2: Packet loss and mean packet latency for WSN comprising of 1000 sensor nodes

To assess the impact of the trust metric weight values on performance, we have carried out simulations for a varying number of malicious nodes issuing modification attacks and different values assigned to the integrity metric. The considered WSN consisted of 100 nodes, the location of the malicious nodes was randomly selected, 10 data sessions were established with each of them generating 900 data packets.

The evaluation metric of interest is the number of modified packets since this reveals how fast the malicious nodes are detected and avoided for cooperation. In all the scenarios tested for ATSR, the number of modified packets increases at the beginning of the simulation run (since nodes attempt cooperation before they detect the malicious neighbours) and stops after a number of interactions. Expressing the modified packets as a percentage of the overall transmitted packets is not only meaningless but also misleading since first, this does not reflect the responsiveness of the trust model and second, we would obtain different ratios depending on the run time (the number of packets generated in the simulation run is directly proportional to the simulation run time).

The obtained results are shown in Figure 3, where the numbers of modified packets for different configurations of the ATSR and different penetrations of malicious nodes are included. Comparing the results for 30 malicious nodes in the network, when the integrity metric is set equal to 0.7 (*Wint=0.7*), only 41 interactions are needed to reveal the malicious nodes while for *Wint=0.3*, malicious nodes are detected after 50 interactions. The difference becomes more evident as the number of malicious nodes in the network increases. For 50% malicious nodes,

with *Wint=0.3*, 3350 modified packets have been measured while for *Wint=0.7* this number reduces to 96.

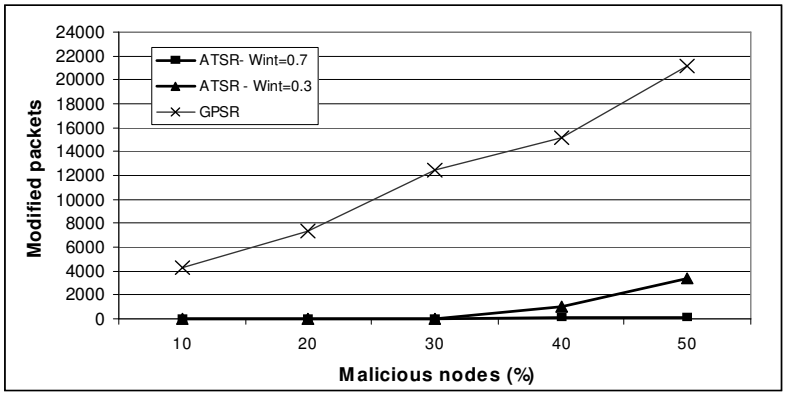

Figure 3: The responsiveness of the trust model (expressed in terms of modified packets) as a function of malicious nodes in the network

For comparison reasons, we have also carried out the same tests for a non trust-aware geographical routing protocol (the GPSR). In this case, the number of modified packets constantly increases throughout the simulation run since there is no tool to detect and avoid nodes issuing modification attacks. Thus, all the packets of the session continue traversing malicious nodes and the modified packets travel to the destination. This implies that the number of measured attacks depends on the simulation time and represents a fixed (over time) percentage of the generated data packets. Having pointed this out, for the simulation parameters previously described and for 10 malicious nodes, 4335 packet modifications took place (10% of the transmitted packets), while for 50 attackers 21,142 packet modifications (51% of the packets) were observed.

Paying special emphasis on the energy metric, since energy is a scarce resource in WSN's and recognizing that energy consumption should be taken into account in all OSI-layers protocol design, we have chosen to include it in the information used when choosing the next hop node. In the ATSR case, this choice has an additional advantage: it leads to path alteration and better load balancing which contributes in defending against traffic analysis attack. To clarify, the forwarding path changes in the duration of a session since the remaining energy of the next hop node that is selected for forwarding drops faster than any other neighbouring node. This way an adversary that monitors the traffic flows trying to identify the nodes that handle the greatest part of the traffic will fail to clearly distinguish among them since the traffic is balanced.

Both the simulation and the experimental results have shown that the performance benefits depend on the weight assigned to the energy metric and can reach a 5% reduction of the consumed energy and the energy metric weight is equal to 0.8. This reduction is important considering that the reputation protocol causes significant increase in the energy consumption due to the introduced overhead.

4.3. Indirect trust

The exchange of indirect trust information among neighbouring nodes has been strongly proposed in the literature and proven (based on simulation results) to be useful, especially in the case of mobile sensor nodes, since nodes arriving in a neighbourhood can obtain trust information from their neighbours before they perform direct interactions. Our goal here is to quantify these benefits and explore the impact of the reputation protocol design choices.

On this purpose, we have run a simulation scenario set for 36 sensor nodes initially placed on a

symmetric grid, 27% of them (randomly placed on the grid) acting as black-hole nodes. Their locations are shown in Figure 4. Node 5 was selected to act as the mobile node crossing a block of malicious nodes and starts transmitting after 100 seconds when it enters in attackers' area.

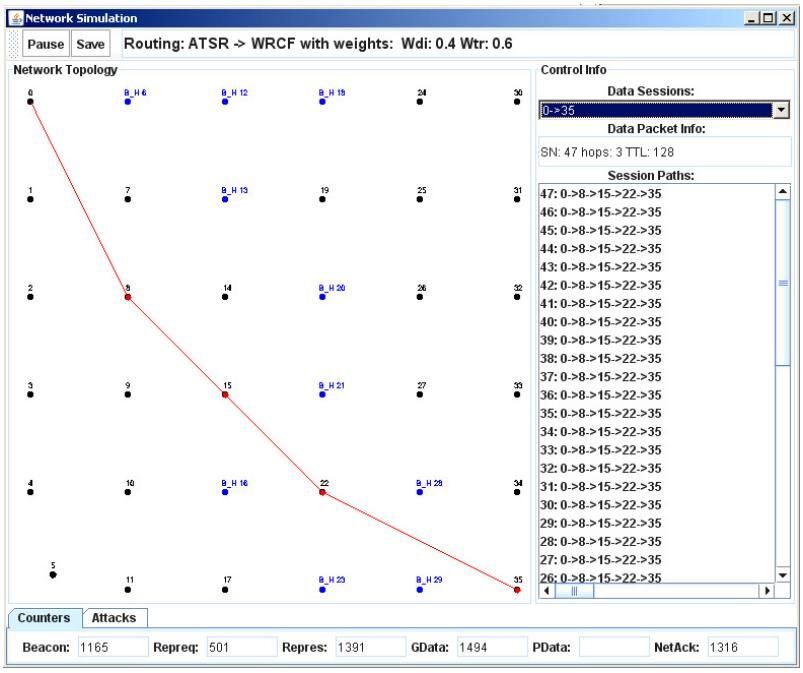

Figure 4: The topology used to evaluate the indirect trust protocol aspects

The results obtained for three different cases, namely when no trust, when only direct trust and when both direct and indirect trust information is taken into account for routing decisions, are reported in Table 1. We also varied the period of reputation request message generation which affects the number of generated messages and the frequency of indirect trust information collection.

The measured packet loss of the monitored session (from Node 5 to 30) is dramatically reduced from 52% to 1% when only direct trust is built and is further reduced to 0.26% when the reputation scheme is activated (for the first three indirect trust scenarios). Executing the reputation exchange protocol, the mobile node issues reputation request messages to its neighbours and thus becomes aware of the malicious nodes before it attempts any cooperation with them. The simulation results showed that the mobile node experienced only 1 attack (just one dropped packet) proving the benefits of the realization of the reputation scheme, when the reputation request generation period is less than 20s.

	Reputation request generation period (s)	Packet loss (%)	Indirect trust messages/data messages
No trust-awareness		52	0
Direct trust		1	0
Both direct and indirect trust	3	0.26	6.68
	7	0.26	2.87
	10	0.26	2
	20	2.1	1

Table 1: Packet loss and introduced overhead comparison for indirect, direct and no trust awareness

However, this performance improvement comes at the cost of high overhead. The implementation of the indirect trust model dictates for the exchange of two new messages: the Reputation Request and the Reputation Response messages. The transmission, reception and processing of these messages, even in the absence of any malicious node, consumes significant energy resources (apart from memory resources). This increase in energy consumption is directly proportional to the number of exchanged messages. In the presented protocol, placing emphasis on the energy consumption, for each reputation request message, just four reputation response messages are generated. Thus, if the reputation response message is generated with frequency equal to the beacon message frequency, then the energy consumed for routing and trust purposes is multiplied by six.

The performance improvement brought by the reputation scheme highly depends on the relation among the frequency of the reputation exchange, the frequency of data message generation and the node speed. When the node moves so fast that each time a new data message is generated, the node is located in a different neighbourhood, if the reputation exchange occurs more frequently than the data transmission, then every time that a data message has to be transmitted, the (moving) source node has already gathered indirect trust information and can choose the next-hop node in a secure manner. Otherwise, the data forwarding decisions are made without any (indirect) trust information taken into account, and data forwarding attempts produce direct trust measurements. In the latter case, the reputation exchange mechanism is almost useless.

In the presented simulation scenarios involving indirect trust, when a reputation request message is issued every 3s (data packets generated every 5s), the measured ratio between the reputation –related messages and data messages is 6.68 (as also shown in Table 1). The same (excellent) performance in terms of packet loss is achieved for reputation request period equal to 7s and 10s, while the introduced overhead is significantly lower. In the last reported case, where reputation requests are issued every 20s, the packet loss increases at 2.1. The performance in this case is worse than with direct trust only, because not only the nodes do not obtain indirect trust information prior to direct interactions, but additionally the exchange of reputation information causes traffic congestion. It is worth pointing out that even with that rare reputation request the number of reputation –related messages is equal to the number of data messages. These conclusions were also validated during the experiments in the AWISSENET test-bed.

5. THE ATSR IMPLEMENTATION ARCHITECTURE

The finalization of the ATSR design triggered the implementation activities in real-life sensor nodes to prove the feasibility of its implementation and evaluate the related cost. The ATSR protocol was implemented in IRIS sensor nodes [32] running TinyOS v2.1 and was integrated with other security-related modules in the AWISSENET test-bed consisting of 100 nodes. Its integration with the Distributed Intrusion Detection System, Secure Service Discovery, Location Identification scheme, and encryption and application module proves the feasibility of implementing such a trust-aware routing with other modules in state-of-the-art sensor nodes and allows for performance comparisons between computer simulations to the real network case.

The test bed consisted of 100 IRIS sensor nodes (equipped with ZigBee interface, 2.404 – 2.481GHz / IEEE 802.15.4, an Atmel ATmega1281 processor, 8K RAM and 128K ROM memory) [32] acting as simple sensors. Each sensor is equipped with a sensor and data acquisition board (MDA100CB) [33] with a precision thermistor, a light sensor and a general prototyping area for mounting custom sensor circuits. Moreover, at least one gateway node is included in the test-bed which provides higher level data processing or interconnectivity functionality for exchanging information with other remote islands via VPN connections. The gateway consists of an Intel Atom-based motherboard connected with an FPGA and an IRIS mote (via MIB520 [34] programmer board USB port).

All network nodes are running TinyOS-2.1 which is an open-source operating system designed for wireless embedded sensor networks. It features a component-based architecture which enables rapid innovation and implementation while minimizing code size as required by the severe memory constraints inherent in sensor networks. TinyOS's component library includes network protocols, distributed services, sensor drivers, and data acquisition tools – all of which can be used as-is or be further refined for a custom application. ATSR was implemented as a new TinyOS event-driven component in NesC [35] [36] considering three basic principles: Low memory, optimal energy consumption and low processing requirements. The TOSSIM [37] framework was used to debug, test and analyze algorithms in a controlled and repeatable environment. The TOSSIM results were validated by the experimental results from the test-bed.

The high-level architecture of the ATSR block is shown in Figure 5**.** Every time an event regarding the direct trust metrics (packet forwarding, precision, network Ack or energy) occurs, the corresponding values stored in the Direct Trust table are updated. Similarly, each time reputation response messages are received, their content is stored to the indirect trust table after the checks outlines in section 4. Both tables include information for each neighbour. The total node's trust value is only calculated when a new message has to be transmitted, even though the trust measurements and reputation information is updated every time an event has occurred or a reputation response message has been received. This event-driven approach was adopted to economise mainly energy but also processing resources. To make the final routing decision, the distance metric which is calculated per neighbour based on the location coordinates announced through the beacon messages, is combined with the total trust value to define the RF value per neighbour.

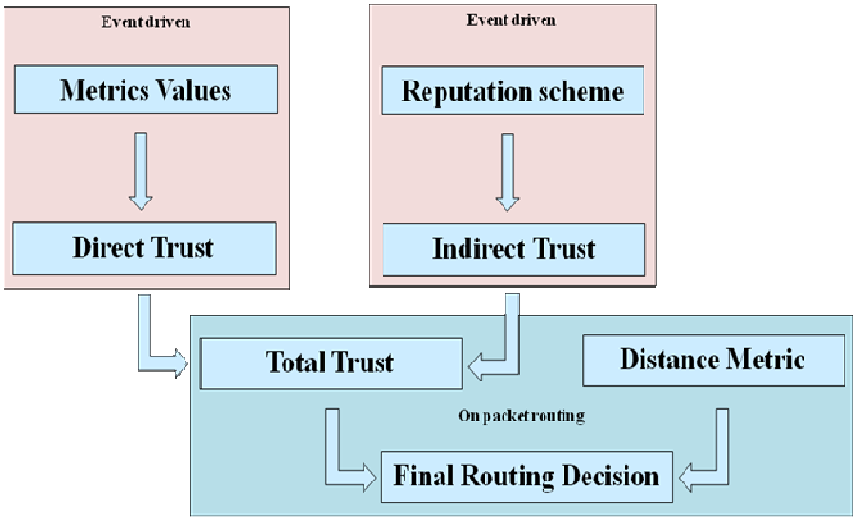

Figure 5: High-level architecture of the ATSR trust model

For debugging and demonstration purposes, we developed a custom software tool, based on the *Listen* library of TinyOS which is capable of showing the remaining energy, node coordinates and ID, as well as the temperature and lighting indications, the types of messages, the routing path (number of hops and node id), the neighbouring nodes and the packet loss indication. An example screenshot from the developed tool and part of the test-bed are shown in Figure 6.

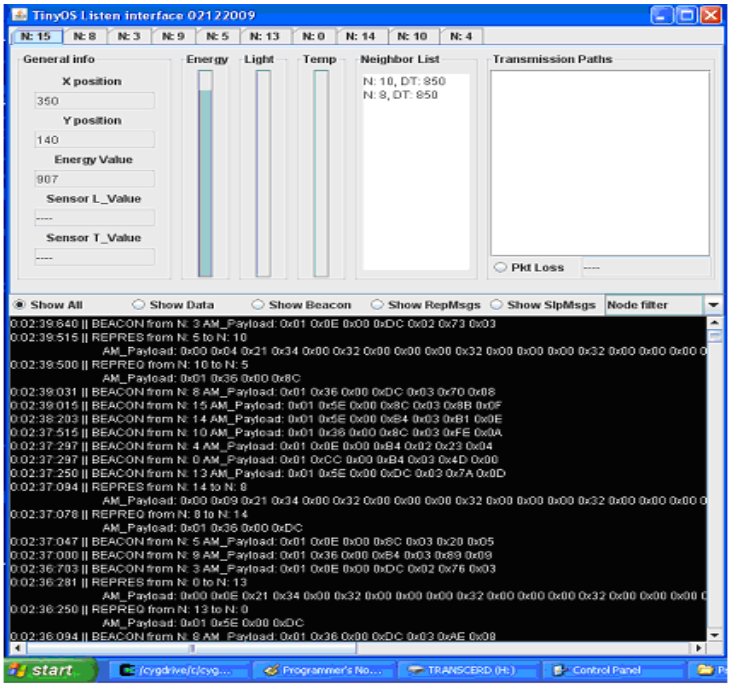

Figure 6: Screenshot from the validation tool

6. IMPLEMENTATION COST

The code implementing the ATSR (including the trust model and the routing protocol) was successfully compiled, consuming in total 35Kbytes of the available 128K ROM and about 3.5Kbytes of the available 8K RAM of the IRIS motes. Focusing on the trust model, the memory requirements per trust metric are included in Table 2 for 8 neighbouring nodes in the routing table and for TOSH_DATA_LENGTH= 120, which is the maximum data length in link layer. The default value in TinyOS 2.1 is 28 bytes and in the proposed implementation each increment by 20 bytes costs 600 bytes in RAM memory.

Trust Metrics	RAM (bytes)	ROM (bytes)	
Forwarding	120	286	Direct trust components
NetAck	110	106	
Integrity	868	48	
Authentication	10	64	
Energy	146	514	
Reputation exchange protocol	374	2,222	Indirect trust components
Reputation Response	77	332	
Reputation Validation	90	180	
Total trust functionality	1,795	3,752	
ATSR	3,500	35,000	

Table 2: Analysis of the implementation cost of the ATSR block

It is worth stressing that due to the adopted location-based routing principle, each node keeps location and trust information only for its one hop neighbours and thus, the implementation cost does not depend on the number of nodes in the network but on the number of nodes in the neighbourhood, i.e. the network density, for a given transmission range. As the number of neighbours increases, it is mainly the RAM requirements that increase. Especially, for each new neighbour in the table, the cost in RAM memory increases by 215 bytes.

7. LESSONS LEARNED

The realisation of a trust management scheme was proved to efficiently detect an important number of routing attacks at affordable resource cost, acting as a first line of defence against adversary nodes. More sophisticated attacks, attempting to mislead the routing protocol's state machine (like stale or wrong routing information, advertisement attack and acknowledgement spoofing) can be detected only by more powerful nodes equipped with intrusion detection system functionality. The presented trust model, even though it does not involve any probabilistic calculations for trust, consumes half of the RAM resources spent for the whole ATSR and 11% of the ROM resources.

Focusing on the implementation of the direct trust management functionality, a first observation is that the required memory resources depend on the number of adopted trust metrics, which drives us to examine each metric individually. Starting from the forwarding metric, this has rather low memory requirements (namely 120B of RAM and 286B of ROM). This metric is the one that enables the detection of black-hole and grey-hole attacks which are easily implemented by adversaries and damage the network operation. The experimental tests (based on the AWISSENET test-bed) have validated the conclusions of the simulation results and have shown that the proposed ATSR succeeds in finding alternative paths to the destination even for high penetration of malicious (black- or grey-hole) nodes. Small deviations between the simulation results and the results from the AWISSENET test bed were noticed (higher packet loss in the test-bed) and were attributed to packet loss due to traffic congestion. The possible collisions (apart from contributing to packet loss) mislead the direct trust measurement process, since the trust management system cannot distinguish between a malicious node intentionally dropping a packet and a packet which has collided with another and has thus not been successfully forwarded. The problem of traffic congestion is aggravated when the reputation protocol is in place. To mitigate this problem, an interface between the MAC and the ATSR blocks would allow for distinguishing between congestion and malicious behaviour. To sum up, the maintenance of the forwarding metric is a low-cost tool towards defending against nodes that deny forwarding, i.e. black-hole and grey-hole attacks.

The implementation of the network acknowledgement metric requires similar resources to the forwarding metric and is a first line of defence, targeting the detection of colluding adversary nodes since it checks the proper reception of the data packet at the base station.

Coming to the integrity metric, this consumes significantly more resources than the forwarding and network acknowledgement metrics since it requires message storage and processing to check the integrity of the forwarded data packets. The integrity check was not extended to the control messages because, as these are periodically issued, the relevant information would be repeated later on and moreover, in the reputation exchange protocol case, checking the reputation messages would significantly increase the node processing time leading to packet loss in the nodes.

Regarding the energy metric, the associated low implementation requirements are fully justified by the achieved performance improvements. It is also worth stressing that while the other trust metrics are implemented just for the case of an adversary node in the network, the improvement in energy consumption are valid for the entire network lifetime irrespective of the presence of any malicious node in the network.

Summing up the conclusions regarding the direct trust model realisation, such a scheme based on behaviour monitoring is a powerful tool which efficiently detects misbehaving nodes. The difference between non trust-aware routing protocol and the proposed trust aware solution is more than evident. As the implementation cost depends on the number and type of the employed metrics and the efficiency in detecting each type of attack depends on the weight value assigned to the relevant metric (which sum up to 1), the trust model designer and implementer should very well consider the number of employed trust metrics and balance the weight values

according to the expectation and/or importance of each attack. Adopting all the trust metric listed in [28] with each of them assigned a low weight value may lead to a trust model with low responsiveness and high implementation cost. Instead, high performance at low cost can be achieved by designing a trust model that takes into account the specific WSN application and the level of motivation of the potential adversary.

The incorporation of the indirect trust functionality requires significantly higher ROM memory resources than the implementation of the direct trust, with the direct trust functionality consuming 1108bytes of RAM and 504bytes of ROM while the indirect trust consumes 541B of RAM and 2734B of ROM respectively. These absolute values compared to the available node resources lead to the conclusion that the implementation of the indirect trust scheme should be well justified by the benefits it brings, i.e. rapid trust build-up in mobile sensor networks. The realization of an indirect trust information exchange scheme offers excellent performance (in terms of packet loss), however, in very specific cases of mobiles nodes and when the reputation request generation frequency is fine-tuned with respect to the node mobility and data generation frequency. It introduces very high overhead with direct consequences on energy consumption and message collisions.

Based on these results, a viable solution is to implement the reputation request exchange frequency as a dynamically changing parameter and vary it according to the node mobility status and the application messages generation frequency. In other words, static nodes do not need to exchange reputation information, thus the relevant overhead can be economized (i.e. reputation request frequency equal to 0). On the contrary, mobile nodes should activate the reputation mechanism (in order to obtain indirect trust information from their neighbours) and set the reputation frequency equal to or greater than the application message exchange frequency (assuming periodic data generation). This way, it is only the mobile nodes that cause the exchange of reputation request and the overall overhead is significantly lower.

Regarding the trust metrics used to protect against the reputation model –related attacks, (i.e. the reputation responsiveness and reputation validation), the required resources are low and thus, their implementation is fully suggested when the realization of a reputation request scheme is considered necessary.

8. CONCLUSIONS

Ad-hoc personal area networks (PAN) and wireless sensor networks impose new challenges on the design of security tools which are more imperative than ever due to their unattended operation in open environments. To defend against routing attacks, the implementation of a trust management system is suggested. We presented a trust-aware routing protocol that can efficiently detect and avoid nodes issuing routing attacks based on a distributed trust management system. The proposed routing solution was successfully implemented and validated in real-life sensor nodes proving its implementation feasibility. The realisation of a trust-aware routing protocol brings clear performance benefits as both the simulation and real –life test-bed results have shown. The involved implementation cost mainly depends on the adoption of a reputation exchange protocol and on the number of behaviour aspects used for the evaluation of each node's trustworthiness. Thus, in the design and implementation of security solutions, the specific application scenario and the degree of attacker's motivation should be well considered.

Acknowledgment: The work presented in this paper was partially supported by the EU-funded FP7 211998 AWISSENET project and ARTEMIS JU 2008-100032 SMART (Secure, Mobile Visual Sensor Networks Architecture) project.

References

[1] Son, B., Her, Y., Kim, J., "A Design and Implementation of Forest-Fires Surveillance System based on Wireless Sensor Networks for South Korea Mountains", IJCSNS International Journal of Computer Science and Network Security, vol.6 No.9B, 124–130, September 2006.

[2] Mainwaring et al, "Wireless Sensor Networks for Habitat Monitoring", International Workshop on Wireless Sensor Networks and Applications (ACM), Sep. 2002,

[3] Chintalapudi, K.; Fu, T.; Paek, J.; Kothari, N.; Rangwala, S.; Caffrey, J.; Govindan, R.; Johnson, E.; Masri, S., "Monitoring civil structures with a wireless sensor network," *Internet Computing, IEEE* , vol.10, no.2, pp. 26-34, March-April 2006

[4] Ian F. Akyildiz, Tommaso Melodia, Kaushik R. Chowdhury, "A survey on wireless multimedia sensor networks", *The International Journal of Computer and Telecommunications Networking*, Vol. 51 , Iss. 4, March 2007, pp. 921-960.

[5] V. C. Giruka, M. Singhal, J. Royalty, S. Varanasi, "Security in wireless sensor networks", *Wirel. Commun. Mob. Comput.* 2008; 8:1–24.

[6] T.Kavitha, D.Sridharan, "Security Vulnerabilities In Wireless Sensor Networks: A Survey" *Journal of Information Assurance and Security*, Vol. 5 (2010) 031-044.

[7] Jaydip Sen, "A Survey on Wireless Sensor Network Security", *International Journal of Communication Networks and Information Security (IJCNIS)* Vol. 1, No. 2, August 2009.

[8] Chris Karlof, David Wagner, "Secure routing in WSNs: attacks and countermeasures", Ad hoc networks Journal, vol. 1, Issue 2-3, Sept. 2003, pp.293-315.

[9] G. Padmavathi, D. Shanmugapriya, "A Survey of Attacks: Security Mechanisms and Challenges in Wireless Sensor Networks", (IJCSIS) *International Journal of Computer Science and Information Security*, Vol. 4, No. 1 & 2, 2009.

[10] Asad Amir Pirzada, Chris McDonald, and Amitava Datta "Performance Comparison of Trust-Based Reactive Routing Protocols", *IEEE Transactions on Mobile Computing*, Vol. 5, No. 6, June 2006.

[11] Y. Sun, Z. Han, K. J. RAY Liu, "Defense of Trust Management Vulnerabilities in Distributed Networks", *IEEE Communications Magazine*, February 2008, pp: 112–119.

[12] Sencun Zhu, Sanjeev Setia, Sushil Jajodia. "LEAP: Efficient Security Mechanisms for Large-Scale Distributed Sensor Networks", Proc. Of 10th ACM conf. on Computer and communications security, 2003.

[13] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar, "SPINS: Security Protocols for Sensor Networks, 7th Annual International Conference on Mobile Computing and Networking (MobiCom 2001), 2001.

[14] Chris Karlof , Naveen Sastry, David Wagner "TinySec: A Link Layer Security Architecture for Wireless Sensor Networks" SenSys'04, November 3–5, 2004, Baltimore, Maryland, USA

[15] HangRok Lee, YongJe Choi, HoWon Kim "Implementation of TinyHash based on Hash Algorithm for Sensor Network". World Academy of Science, Engineering and Technology, Vol. 10, 2005.

[16] Matthias Becker, Sven Schaust and Eugen Wittmann. Performance of Routing Protocols for Real Wireless Sensor Networks. 10th Int. Symposium on Performance Evaluation of Computer and Telecommunication Systems (SPECTS'07), San Diego, USA, 2007.

[17] Matthew J. Probst and Sneha Kumar Kasera, "Statistical Trust Establishment in Wireless Sensor Networks", 13th International Conference on Parallel and Distributed Systems, 2007.

[18] Xiaoqi Li, Lyu, M.R., Jiangchuan Liu, "A Trust Model Based Routing Protocol for Secure Ad hoc Networks", IEEE Proceedings on Aerospace Conference, 2004, vol. 2.

[19] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating Routing Misbehavior in Mobile Ad hoc Networks," in Proceedings of the Sixth Annual International *Conference on Mobile Computing and Networking (MobiCom)*. ACM Press, 2000, pp. 255–265.

[20] Asad Amir Pirzada Amitava Dattaa and Chris McDonald, "Incorporating trust and reputation in the DSR protocol for dependable routing" *Computer Communications* Volume 29, Issue 15, 5 September 2006, Pages 2806-2821

[21] S. Buchegger and J. Boudec, "Performance Analysis of the CONFIDANT Protocol: Cooperation Of Nodes - Fairness In Distributed Ad-hoc NeTworks," in Proceedings of the 3rd *ACM International Symposium on Mobile Ad hoc Networking and Computing* (MobiHoc), ACM Press, 2002, pp. 226–236.

[22] P. Michiardi and R. Molva, "CORE: A Collaborative Reputation Mechanism to Enforce Node Cooperation in Mobile Ad hoc Networks," in Proceedings of the IFIP TC6/TC11 Sixth Joint Working *Conference on Communications and Multimedia Security*, vol. 228. Kluwer Academic Publishers, 2002, pp. 107–121.

[23] Asad Amir Pirzada and Chris McDonald "Trusted Greedy Perimeter Stateless Routing", IEEE, *ICON2007*

[24] Sapon Tanachaiwiwat, Pinalkumar Dave, Rohan Bhindwale, Ahmed Helmy "Location-centric Isolation of Misbehavior and Trust Routing in Energy-constrained Sensor Networks" IEEE *International Conference on Performance, Computing, and Communications*, 2004

[25] Adrian Perrig, Robert Szewczyk, J.D. Tygar, Victorwen And David E. Culler "SPINS: Security Protocols for Sensor Networks" ACM Journal of Wireless Networks, 8:5, September 2002, pp. 521-534

[26] Y.C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," Proc. Eighth Ann. Int'l *Conf. Mobile Computing and Networking* (MobiCom), pp. 12-23.

[27] Garth V. Crosby and Niki Pissinou, "Cluster-based Reputation and Trust for Wireless Sensor Networks" *Consumer Communications and Networking Conference*, 2007. CCNC 2007Las Vegas, NV, USA, Jan. 2007

[28] Theodore Zahariadis, Helen Leligou, Panagiotis Trakadas, Stamatis Voliotis, "Trust management in Wireless sensor Networks", European Transaction on Telecommunications, 2010.

[29] K.Seshadri Ramana, A.A. Chari and N.Kasiviswanth "A Survey on Trust Management for Mobile Ad Hoc Networks IJNSA, April 2010, Vol. 2, No2.

[30] Brad Karp, H. T. Kung, "GPSR: Greedy Perimeter Stateless Routing for WirelessNetworks", MobiCom 2000.

[31] www.j-sim.org

[32] http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/IRIS_Datasheet.pdf

[33] http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MTS_MDA_Datasheet.pdf

[34] http://www.xbow.com/Products/Product_pdf_files/Wireless_pdf/MIB520_Datasheet.pdf

[35] http://nescc.sourceforge.net/papers/nesc-ref.pdf

[36] http://www.tinyos.net/tinyos-2.x/doc/pdf/tinyos-programming.pdf

[37] http://www.eecs.berkeley.edu/~pal/pubs/nido.pdf