Research Article

Combining trust with location information for routing in wireless sensor networks

H. C. Leligou1, P. Trakadas2, S. Maniatis2, P. Karkazis1 and T. Zahariadis1

1 Department of Electrical Engineering, TEI of Chalkis, Psahna, Greece
2 Hellenic Authority for Communications Security and Privacy (ADAE), Athens, Greece

Abstract

As the applications of wireless sensor networks proliferate, the efficiency in supporting large sensor networks and offering security guarantees becomes an important requirement in the design of the relevant networking protocols. Geographical routing has been proven to efficiently cope with large network dimensions while trust management schemes have been shown to assist in defending against routing attacks. Once trust information is available for all network nodes, the routing decisions can take it into account, i.e. routing can be based on both location and trust attributes. In this paper, we investigate different ways to incorporate trust in location-based routing schemes and we propose a novel way of balancing trust and location information. Computer simulations show that the proposed routing rule exhibits excellent performance in terms of delivery ratio, latency time and path optimality. Copyright © 2010 John Wiley & Sons, Ltd.

Keywords
wireless sensor networks; trust management; secure routing protocol; location-based routing; geographical routing; performance evaluation

*Correspondence
H. C. Leligou, Department of Electrical Engineering, TEI of Chalkis, Psahna, Greece.
E-mail: leligou@teihal.gr

1. INTRODUCTION

Wireless sensor networks (WSN) are gaining more and more attention due to the great variety of the applications they offer [1]. Recently, research efforts in this area are focusing on the design of large WSN [2] and on their integration in the future Internet [3], while the realisation of security solutions has not yet reached a mature stage, although it consists a key user requirement for a wide set of applications. The reason is that although a long list of possible security attacks has been defined [4–6], the implementation of security solutions is impeded by the limited node capabilities in terms of processing power, memory space and energy resources, rendering infeasible the application of legacy security solutions.

In the infrastructure-less environment of a WSN, nodes rely on the cooperation among each other for forwarding their packets towards the sink node. This cooperative routing procedure can be exploited by adversaries to damage the network operation. For example, an adversary node may exhibit selfish behaviour and refuse to forward all (or part of) its neighbours’ traffic [6] realising a black-hole (or grey-hole) attack. To detect and avoid malicious nodes, the implementation of a trust establishment system has been proposed in many research articles (e.g. Ref. [7,8]): nodes monitor the behaviour of their neighbours in order to establish trust relationships between each other. The trustworthiness of each node is then taken into account to make higher layer decisions, such as routing and data aggregation [9]. Although a plethora of trust management schemes has been proposed to define the trustworthiness of each neighbour based on direct [10] and indirect [11] evidence gathered from all or part of the neighbours in a probabilistic [12] or not [13] way, relatively little attention has been paid to the way the trust information is incorporated in the routing algorithm. In other words, each routing algorithm defines the next hop (or path) through which the packet will be routed based on some criteria such as location, latency, link quality, hop count, etc. [14]. Trust is an additional criterion, which has to be introduced in the routing algorithm. Routing the packets through the most trusted neighbour is not necessarily the optimum choice, since it usually results in very poor performance in terms of latency and network lifetime.

In this paper, we are focusing on the way that trust information is combined with location-based routing protocols.
Our choice was driven by the fact that they are inherently scalable and support node mobility. These two features are mandatory for the support of large WSNs and stem from the fact that routing is decided based on local information only. Each node periodically broadcasts a beacon message announcing its location, and thus allowing its neighbours to be aware of its presence. When a node has a packet to route, the (one-hop) neighbour that is closest to the destination is selected. This procedure is repeated until the packet reaches the destination. Adopting this approach, each node keeps routing information only for nodes within its range. The most widely cited geographical routing protocol is the greedy perimeter stateless routing (GPRS) algorithm for WSN initially presented by Karp and Kung [15]. Pursuing energy consumption improvements the ‘geographical and energy aware routing’ (GEAR) was proposed in Ref. [16] while other energy-aware location-based routing protocols have also been presented in Ref. [17,18]. Significant research effort has been spent on the impact of location errors [19,20] and the design of techniques that obviate the need for accurate physical position coordinates [21–23]. In all these works, the routing is based on (physical or virtual) location information.

Furthermore, in this paper, we investigate different ways to incorporate trust information in geographical routing algorithms for WSNs and we present a novel routing solution which uses a weighted routing cost function to perform trust- and location-aware routing. Computer simulations are used to evaluate the performance in terms of delivery ratio and message latency in the presence of malicious nodes in the network. The proposed routing rule is shown to offer excellent performance even for 50% malicious nodes in the network, while at the same time its implementation complexity is limited and affordable in the severely constrained sensor environment. In the rest of the paper, we first briefly report the related work on trust-aware routing found in the literature, while in Section 3 we describe the routing rules under investigation and detail the ambient trust sensor routing (ATSR) solution which capitalises on the weighted routing cost function. The developed simulation model is detailed in Section 4. The performance results are included in Section 5 where first each routing rule is individually explored and then they are compared. Final conclusions are drawn in Section 6.

2. RELATED WORK ON TRUST-AWARE ROUTING

To enhance the security of the routing procedure in a WSN, the researchers have designed and evaluated a wide variety of trust management systems [24] which target the evaluation of the trustworthiness of each node and the detection of the malicious ones. The rules of the routing protocols are then modified to take into account the trustworthiness of each neighbour. For protocols that enforce source routing (i.e. the source defines the exact path to the destination that every packet has to follow), the trustworthiness of all participating nodes has to be disseminated throughout the network mandating the realisation of a trust information exchange scheme. Such a protocol, namely the dynamic source routing (DSR) is modified in Ref. [25] so that the path with the highest trust is selected among those leading to the desired destination, instead of selecting the shortest path, to enhance security. In the same way, similarly in AODV (Ad hoc on-demand distance vector routing) protocol, the highest-trust path towards the destination is selected instead of the least-hop path.

For routing protocols that operate on a hop-by-hop manner, direct trust measurement is adequate, i.e. each node evaluates the trustworthiness of all its one-hop neighbours which renders the overall solution more scalable, consuming less network and node resource since the overhead required for the dissemination of the trust information is economised. Focusing on geographical routing, this presents the intricacy that each node decides only the next hop and given that no path is established, the next hop should be in the correct direction for the packet to reach its destination. In other words, in DSR, the source node has to select among a set of alternative paths that assuredly lead to the destination and is thus quite easy to replace the ‘shortest’ criterion with the ‘most trusted’. In geographical routing, each node makes a local decision and there is no guarantee that the packet will finally reach its destination if the location criterion is overlooked for the trust.

To enhance the security in geographical routing two parallel directions are followed: first, a trust model is implemented to assess the cooperating willingness of each neighbour and detect black-hole, grey-hole and integrity attacks [26]. Second, to defend against Sybil attack (i.e. discover a node which tries to deceive its neighbours forging multiple non-existent identities in an attempt to fool other nodes with the illusion that these identities are cooperating), the verification of the location information becomes imperative, since it consists the basis of the routing procedure [27]. Interesting location verification techniques are proposed in Ref. [27] where additionally, multipath routing is suggested, to increase the probability of reaching the destination, sacrificing node and network resources for the transmission of multiple copies of each packet. In this work, Kang et al. present a trust-aware location-based routing protocol following which the nodes located closer to the destination than the source node form the forwarding set (FS) and the next hop is selected from the FS with most trusted neighbours selected with higher probability. This approach ensures that the packets will travel through a trusted, possibly longer path, which may result in higher latency and overall energy consumption in the network. If we opt for unicast forwarding, following this routing rule, the highest probability is to select the most trusted node in the FS. The most trusted node in the FS is also selected in Ref. [26] where the trust is defined as the ratio between the differences between successfully completed and failed coopetations versus the total number of attempted cooperations.

DOI: 10.1002/wcm
Systematically selecting the most trusted nodes results in the ‘effect of node out of energy’ (i.e. the energy of the most trusted nodes is exhausted prior to any non-trusted node), while leaving in the FS all nodes closer to the destination than the source node, it is very possible that a node close to the source node may be selected. The result is that the packets do not follow optimal paths and travel over a higher number of hops. To face these issues and extend the network lifetime, the authors of Ref. [28] narrow the FS taking into account the coverage area of each node. However, this approach consumes significant node resources, since it requires the derivation of the coverage area of each neighbour, based on beacon messages and rather complex (for a sensor node) estimations and iterations. Then, the probability of selecting a neighbour from this set is based on the relative trust value.

A rule derived from Ref. [13] and also reported and evaluated in Ref. [28] is to route packets through nodes with trustworthiness higher than a predefined value. These nodes are assumed to form a trusted neighbour set (TNS). This choice introduces the need for selecting an application-dependent trust threshold which enables security differentiation. This may turn out to be a difficult task especially in the absence of any ‘standardised’ way to define trust. The advantage of this approach is that optimal routing is performed when many honest neighbours (with trust value exceeding the threshold) exist. On the other hand, when no such node exists, the connection is blocked, resulting in limited connectivity. To alleviate the blocking drawback, an alternative approach could be to define the TNS based on relative trust criteria (e.g. include the k most trusted nodes in the TNS), and then select the node that is closest to the destination.

It is evident that the way trust is combined with location information to produce the routing decision affects the provided security and the path optimality. Underestimating the significance of trust may result in high packet loss ratio as the percentage of malicious nodes in the WSN increases. On the other hand, overlooking the distance criterion results in long paths, which, apart from introducing higher latencies, lead to significant energy waste in this resource-constrained environment. Balancing trust with location turns out to be an interesting and challenging task which is at the focus of our investigation.

3. DESIGNING A TRUST-AWARE ROUTING RULE

In order to isolate the interplay between location information and trust, we outline four trust-aware routing rules based on which the next hop neighbour is selected (i.e. unicast forwarding is adopted to economise energy resources), following a limited number of steps to keep the required processing load as low as possible. All these rules represent already deployable solutions requiring minimal processing effort for the sensor node and can efficiently avoid malicious nodes, as will be evaluated thoroughly in the next section.

3.1. Rule 1: TF-rule

Placing emphasis on security, the nodes existing in the neighbour list are first ranked based on their trustworthiness. The most trusted nodes form the TNS with population k and then the next hop node is decided based on location criteria, i.e. the node in the TNS that is closest to the destination is selected. This rule will be called hereafter as TF-rule since trust information is inspected first and can be formulated as follows:

To define the next hop node:

1. Define the TNS, which includes the k most trusted neighbours, i.e. let NS = \{n_0, n_1, \ldots, n_m\} be the full set of one-hop neighbours, ordered by decreasing trust value and TNS = \{n_0, n_1, \ldots, n_k\}.
2. From nodes in TNS, choose the node located closest to the destination, i.e. choose n_j so that \(d_{n_j,D} = \min(d_{n_i,D})\) for all \(n_i\) in the TNS, where \(d_{n_i,D}\) the distance between node \(n_i\) and the destination of the packet \(D\).

The TNS population parameter (k) offers the flexibility of balancing the importance of trust to location information when deciding the next hop. Assigning low values to k parameter, trust becomes more important than location, while greater values of k, relax the trust criterion and allow for more distance-aware decisions. As will be explored in Section 5, when k becomes too small compared to the population of the neighbour list, it may happen that all nodes in the TNS are located in the opposite direction than the one the message should follow and thus path optimality is compromised.

3.2. Rule 2: SDDF rule

Reversing the priorities, the nodes in the neighbour list are ranked based on their distance to the destination and those attributed the shortest distance form the FS with population k. The next hop is selected as the most trusted node in the FS. This is a solution derived from Ref. [27] for unicast forwarding and selecting the node from the FS in a deterministic way. This approach economises energy and provides higher security since the most trusted node in the FS is always selected. The rule we explore in this paper is called hereafter SDDF rule, since the nodes of shorter distance to the destination are inspected first and can be formulated as follows:

To decide the next hop node:

1. Define the FS, which includes the k nodes closest to the destination, i.e. let NS = \{n_0, n_1, \ldots, n_m\} be the full set of one-hop neighbours ordered by increasing distance to the destination value and FS = \{n_0, n_1, \ldots, n_k\}.
2. From nodes in FS, select the most trusted node, i.e. choose \(n_j\) so that \(TT_{n_j} = \min TT_{n_i}\) for all \(n_i\) in...
the FS, where $T^A_{n\to i}$ the total trust that the source node $A$ has assigned to node $n_i$.

Again, as in the case of the first rule, the population parameter $k$ of the FS affects the performance of the routing rule. As $k$ decreases, near optimal paths are followed with less emphasis on security. An important advantage is that, when the next hop node is selected based on its relative trust among the nodes in the FS, it is ensured that at least one neighbour will be selected (high connectivity) even when all nodes have low trust values.

### 3.3. Rule 3: Thr-rule

Routing packets through neighbouring nodes that exceed a specific trust value threshold is the only way to provide (absolute) security guarantees and also allow for better load balancing and relative path optimality. A third option is thus to define the TNS using an application-specific trust threshold, in order to make sure that all messages will travel through nodes exceeding this trust threshold and no message will ever traverse a ‘doubtful’ node. The routing rule is marked as Thr-rule (threshold based rule) and is formulated as follows:

To decide the next hop node:

1. Define the TNS, which includes all nodes with trust higher than the predefined threshold value, i.e. Let
   
   $\text{NS} = \{n_0, n_1, \ldots, n_m\}$ be the full set of one-hop neighbours ordered by decreasing trust value and $\text{TNS} = \{n_0, n_1, \ldots, n_j\}$ with $T_{n_i} >$ threshold for $0 \leq n_i \leq n_j$.

2. From nodes in TNS, choose the node located closest to the destination, i.e. choose $n_j$ so that $d_{n_i,D} = \min(d_{n_i,D})$ for all $n_i$ in the TNS, where $d_{n_i,D}$ the distance between node $n_i$ and the destination of the packet $D$.

The achieved performance depends on the predefined threshold. If this is set too high, a node can be easily isolated if all its neighbours have a trust value lower than the threshold. This is not unlikely to happen, e.g. due to packet loss caused by accidental situation such as congestion, or due to the existence of a block of attackers in the node’s neighbourhood. All other rules would have attempted forwarding, even at the risk of experiencing an attack. To alleviate this disadvantage, the Thr-rule can be combined with choosing a random next hop from the neighbour list, when TNS is empty.

### 3.4. Rule 4: ATSR-rule

To balance the importance of trust versus distance in a flexible way, in our ATSR solution, we use a novel routing cost function, which assumes the takes into account (normalised) trust and distance metrics. The routing rule we propose is as follows:

To decide the next hop node:

Select the node that maximises the following routing cost function:

$$RF^A_{B} = W_t \times T^A_{n\to i} + W_d \times d^A_{B}$$

where $W_t$ and $W_d$ are the weights that express the importance of trust $T^A_{n\to i}$ that node $A$ has built for node $B$ and distance respectively, with $W_t + W_d = 1$ while

$$d^A_{B} = \frac{d_{\text{min}}}{d_0}$$

is expressed as the ratio of $d_{\text{min}}$ (the Euclidean distance of the closest to the base station neighbour of $A$, to the base station) over $d_0$ (the Euclidean distance of neighbour $B$ to the base station). Following Equation (2), the shortest distance to the destination maximises the $d^A_{B}$ value (actually it sets it equal to 1). Based on Equation (1), a routing value for each neighbour is calculated and the node that corresponds to the maximum value is selected for forwarding the packet as it represents a good candidate, satisfying both trust and geographic criteria. It is worth stressing that both the total trust value and distance metric $d^A_{B}$ range from 0 to 1, expressing the relevant distance and trust cost.

The rationale behind our choice is that the distance and the trustworthiness of each node have to be balanced. Each source node should select for forwarding a node that is close to the destination and trusted at the same time. When two or more nodes present almost equal trustworthiness, it is the distance criterion that guides the next hop node decision. When two neighbours are located in almost equal distances from the destination, it is the trust that guides the decision. However, when significant differences in the trustworthiness exist, the node that presents the optimum balance between trust and distance to the destination will be chosen. In this case, the closest node to the destination can be overlooked for a more trusted node, even if it is located in a less optimal routing path. In this context, the weight factors can play an important role. As $W_d$ increases, the closest to the destination node will be overlooked only for a node with significantly higher trustworthiness. For example, for $W_d = 0.8$, the closest to the destination neighbour $C$ with trust value equal to 0.8 will be disregarded for node $B$ that is located 5% further from destination than node $C$ only if its trust value exceeds 0.92 (i.e. is 15% greater). For lower $W_d$ values, e.g. for $W_d = 0.4$, node $C$ will be disregarded for node $B$ if its trust value exceeds 0.85 (i.e. is 6.2% greater). This way the parameters $W_t$ offer the flexibility of shifting emphasis from path optimality to trustworthiness in different application areas.

### 4. THE SIMULATION MODEL

To evaluate the routing rules defined above, we have modelled them using the JSim simulation platform [29].
trust model we adopted follows fully distributed trust architecture, i.e. the trust management functionality is distributed over the network nodes. Each node is responsible for computing its own trust value per relation in the network, collecting evidence (events) from direct relations, i.e. interactions with its neighbours. The direct trust value for each neighbouring node is determined based on a limited set of event types. The concept is to create on each sensor a trust repository (trust table), which will maintain and handle trust information about each neighbouring node and per event type (also called trust metric). The total trust value for each neighbour is calculated based on the trust table values regarding the trust metrics. (No indirect trust information exchange is assumed since our target is not to explore the trust model and its capabilities but the way trust is incorporated in the routing algorithm.) In the scenarios presented in this paper, the set of the following five trust metrics was assumed: (Please note that authentication and confidentiality represents different metrics which are dealt with in a similar manner).

1. **Packet forwarding**: To detect and avoid black-hole and grey-hole security attacks [6, 26], each node is evaluated regarding its willingness and sincerity in the routing procedure cooperation. This is checked through overhearing; i.e. each node, immediately after transmitting a message, enters the promiscuous mode in order to check whether the selected neighbour actually forwarded the message or not.

2. **Network layer ACK**: To detect and defend against colluding adversaries attacks, each node checks whether it receives the network layer ACK from the Base Station for each transmitted packet.

3. **Integrity**: To detect nodes performing modification attacks, each time a node transmits a packet, it overhears not only whether the selected neighbour forwarded it but also whether it has performed any unexpected modification to its payload.

4. **Authentication-confidentiality**: A node can collect trust information about neighbouring nodes during interactions regarding the proper use of the security measures applied. For example, a node might use a mechanism to authenticate the message of a neighbouring node or the base station. Furthermore, integrity measures (such as the use of SHA-1 algorithm) and confidentiality measures (symmetric, public key or elliptic curve cryptography) can be applied for the communication between neighbouring nodes.

For each event type $i$ defined above, node $A$ calculates a trust value $T_{A,B}^i$ regarding node $B$ which is equal to the successfully completed interactions versus the total amount of attempted interactions. To calculate the total direct trust value, the trust values are summed up in a weighted manner:

$$TT_{A,B}^i = \sum_{i=1}^{5} W_i \times T_{A,B}^i$$

The weights $W_i$ are equal to 0.5, 0.3, 0.2, 0, and 0, respectively, for the results presented in this paper unless differently stated. The initial trust value for all neighbours has been set equal to 1 (i.e., all nodes are considered to be trusted a priori) as also adopted in Ref. [12] while the transport protocol used operates in a similar way as UDP, with no retransmission at the application layer.

The simulation topology consists of 100 nodes placed on a $10 \times 10$ grid and four application-layer sessions were active unless otherwise stated (e.g. in Section 5.5.). The simulations were carried out for different number of malicious nodes in the network performing grey-hole attacks, uniformly distributed in the network. The topology for 20, 30, 40 and 50 malicious nodes in the network was kept the same for all corresponding runs applying different routing rules. Simulations using more attack types are reported only in Section 5.5.

## 5. PERFORMANCE EVALUATION

The target of the trust-aware routing rules is to find the optimal and trusted path towards the destination. For this reason, the performance is evaluated for the case where malicious nodes exist in the network and is measured in terms of packet delivery ratio, which shows whether the algorithm has detected the malicious nodes and has changed the routing decision accordingly to avoid them, in terms of average latency, showing how optimal the selected path is, which also affects the energy consumed for the end-to-end forwarding of each packet, and the number of performed attacks, which reflects how fast the trust-aware routing rules detect the malicious nodes and how much energy was consumed on failed cooperation attempts (directly affecting the network lifetime).

### 5.1. Performance evaluation of the TF routing rule

Placing emphasis on trust, the TF routing rule chooses for forwarding the node that is closest to the destination among the $k$ most trusted neighbours. The results in terms of delivery ratio (i.e. the ratio between the packet that successfully reached their destination over the packets that were transmitted) and average latency for different values of $k$ are included in Figure 1 where the results for $k$ equal to 3 are indicated as TF 3, for $k$ equal to 6 as TF 6, etc. It is clear that when the percentage of malicious nodes is relatively low, higher $k$ values result in higher delivery ratio since in this case, there are many honest nodes in the neighbourhood and thus, by increasing $k$, each forwarding node has a wider set of candidate nodes to forward the packet and consequently the distance criterion is successfully applied. For $k = 3$, the performance is poor, because most nodes have similar trust values. Thus, restricting the candidate set to the three most trusted, these may be located in any direction and not necessarily towards the destination. This
notion is also supported by the measured average latency values shown in the right hand part of Figure 1. For \( k = 3 \), the average latency has the greatest value in all four cases of malicious nodes existing in the network. Moreover, the difference in the observed latency for different \( k \) values is larger for low penetration of malicious nodes. This situation leads to long routes to the destination, introducing higher risks due to the additional hops, as well as to overall network congestion. Network congestion further causes reduced capability to successfully perform overhearing, which is the basis of the trust model. Therefore, achieving an optimal path is not only a performance metric, but also an important requirement for the correct operation of the trust model.

As the number of malicious nodes in the network increases, if \( k \) becomes close to the neighbour list population (which is approximately 12–20 in our simulations depending on the position of the node in the grid), then no trust-awareness is actually applied, i.e. all nodes are considered trusted and only the distance criterion is applied, leading to a greedy forwarding geographic equivalent routing (e.g. GPSR). This is why for higher \( k \) values (e.g. \( k = 16 \)) the delivery ratio drops faster than for \( k = 6 \) or 10 and becomes lower than the one measured for \( k = 3 \). For this value (\( k = 3 \)), highly trust nodes are only considered for forwarding which leads to higher delivery ratios when 40 or 50\% of nodes are malicious nodes.

Concluding, based on the delivery ratio results, the best performance is achieved for different \( k \) values depending on the percentage of malicious nodes in the network, which drives us to consider the case of reconfiguring this parameter at run time. If the \( k \) value has to remain fixed, this should be set equal to 6 or 10, i.e. almost half of the neighbour list population, since further restricting the TNS does not allow for distance-wise routing decisions, while safeguarding the routing security. For the messages that manage to reach their destination, shorter (near optimal) paths are followed as \( k \) increases as shown in the average latency results. As the ideal \( k \) value depends on the number of nodes in the neighbour list which may fluctuate, if sensors are moving around, this solution is more suitable for stationary WSNs or WSNs with almost fixed density.

### 5.2. Performance evaluation of the SDDF routing rule

The SDDF routing rule defines first the FS based on location criteria, i.e. selects the \( k \) nodes closest to the destination and then the most trusted among them is chosen for forwarding. In this case, it is the population of the first set that affects the performance of the trust-aware routing protocol. The results in terms of delivery ratio and average latency for different \( k \) values are included in Figure 2.
A general remark is that as the number of malicious nodes increases, the delivery ratio drops. Focusing on the performance for different values of the parameter $k$, an obvious result is that when $k$ is chosen to be equal to 10 (curve marked as SDDF 10), the delivery ratio is very low for all percentages of grey-hole nodes. This poor performance is due to the fact that when $k$ increases (becomes close to the number of nodes in the neighbour list), the distance criterion is almost neglected since almost all the nodes in the neighbour list are included in the FS and finally routing is decided based on trust (rather than on distance). The source node selects its most trusted neighbour for forwarding (without bothering about the followed direction) and thus the packet travels a random route through highly trusted neighbours never reaching the destination. (An important part of the traffic is dropped due to TimeToLive expiration.) This is also justified by the latency results (shown at the right hand part of Figure 2) which show that for high $k$ equal to 10 even for low penetrations of malicious nodes, the experienced latency is by far higher than for other $k$ values.

Similar poor performance is observed in the TF rule when $k$ is small relatively to the population of the neighbour list, especially in case of low percentage of malicious nodes. Therefore, we can conclude that performing routing based only on location or only on trust information, provides poor performance. In SDDF, lower $k$ values perform better. In more detail, for $k$ equal to 3, higher delivery ratio than for $k = 6$ or 8 is achieved when less than 30 malicious nodes exist, but this is reversed when the number of attackers increases. Setting $k$ equal to 6 or 8 provides more balanced performance under any percentage of malicious nodes.

The results regarding the average latency for the same simulation runs (included in the right hand part of Figure 2) show that the measured latency remains below 30 ms for all $k$-values tested except $k = 10$. As already justified, for this value, the distance criterion is overlooked and thus packets travel over longer paths to their destination, which introduces higher risks due to the additional hops and leads to overall network congestion, as already discussed in the case of the TF-rule.

Taking into account the delivery ratio and the experienced latency, the overall best performance is achieved for $k = 6–8$. Observing that the best performance measured for different number of malicious nodes corresponds to different $k$ values (exactly as happened for the TF-rule) renders the investigation of reconfigurable node architecture according to the measured trust more prominent.

### 5.3. Performance evaluation of the threshold-based routing rule

Following the Thr-rule, routing is performed over nodes exceeding a predefined trust threshold. In this case, the value of this threshold affects the performance in terms of delivery ratio and average latency in the presence of malicious nodes in the network. The delivery ratio and average latency results for different threshold values and percentage of grey-hole nodes are shown in Figure 3.

Setting this threshold at low values (such as 0.5, which means that packets are routed via nodes that forward at least half of the traffic they receive) results in poor performance since grey-hole nodes are considered as candidates for forwarding, exactly as if they were trusted. Increasing this threshold, higher performance in terms of delivery ratio is achieved. As the trust threshold increases, the delivery ratio drops more sharply when the malicious nodes proliferate. However, it should be stressed that selecting a trust threshold higher than 0.9 (e.g. 0.95) will not bring any significant performance improvement, since a node trust-worthiness may drop to 0.94 due to failed packet forwarding caused by congestion. At this high threshold values, the performance depends also on the robustness of the trust model. The measured latency, shown in Figure 3, remains below 16 ms for trust threshold values equal to 0.5, 0.9 and 0.95. The low latency values reflect the path optimality.

Threshold values equal to 0.80 or 0.90 result in very good performance both in terms of average latency and delivery ratio with 0.9 presenting higher delivery ratios for low grey-hole nodes penetration while sharply degrading as the number of grey-hole nodes increases. The threshold essentially adjusts the population of the FS and narrows it as malicious nodes penetrate in the network. It thus acts as a regulation factor. However, the need to define an application-specific value remains an open issue. For example, for military or financial applications selecting a
When the number of malicious nodes increases, the delivery ratio decreases fast, since trust is sacrificed to distance criteria. (When thresholds are assigned very low values, e.g. 0.2, the delivery ratio is lower and since trust becomes the main criterion, the packets travel long paths (as the latency results prove) to cross a high number of nodes, which introduces risks and network congestion, especially at high percentages of malicious nodes.

For high thresholds, as emphasis is shifted to the distance criterion, the latency decreases, since the packets that manage to reach the destination follow a near-optimal path. (Unfortunately, this comes at the cost of higher packet loss as already explained.) When thresholds decrease towards 0.2, the delivery ratio decreases while the delay increases as well (especially when 50% malicious nodes exist), because, paying less attention to distance criteria, the data packets travel longer paths to the destination through highly trusted nodes sometimes failing to reach their destination, due to the reasons discussed before in all previous cases.

The best performance, both in terms of delivery ratio and latency is observed for $W_d = 0.6$. This weight value represents a good balance between trust and geographic metric. Extreme values either in favor of trust or distance lead to high latency and high packet loss, respectively. For this routing approach, weight values that lead to optimised performance both in terms of delivery ratio and average latency exist, which was not the case for the TF and SDDF rule. It is also worth stressing that this approach represents a deployable solution as it requires only the calculation of a weighted function.

**5.5. Comparison**

To compare the four routing rules in a fair manner, first we compare the best-obtained results for each of them for the already presented scenario settings (100 nodes and grey-hole attackers). Then, for the configurations that lead to the best results for each rule, we ran scenarios including hundreds of nodes and different conditions to check whether the initially obtained comparison results still hold. These scenarios include higher number of nodes (625 nodes), two different types of attacks (grey hole and black hole) while for the ATSR we run scenarios assuming mobility and various types of attacks.

![Fig. 4. Delivery ratio and average latency for different distance and trust weight values (ATSR routing rule).](image-url)
neighbour list size. The TF rule clearly results to poor performance and can only be considered as a good alternative solution to the threshold based routing, when high security requirements are dictated. The threshold-based rule also exhibits a smooth degradation but is outperformed in all cases by the ATSR function. Comparing the rest modelled routing rules, SDDF performs better than the TF rule while in general the relative advantages observed for a number of malicious nodes for a certain rule are reversed for other scenarios.

Concentrating on the average latency (reflecting also the path optimality), the ATSR rule again leads to the lowest average latency value for all the tested cases, closely followed by the threshold based rule. As regards the TF and SDDF rule, the latency strongly depends on the population $k$ of the TNS and FS sets, respectively, driving the decisions for the best value to be adopted. It is worth stressing that this value also depends on the number of nodes in the neighbour list, and thus a final decision on the best value for $k$ is not a straightforward process. Similar conclusions are drawn based on the number of the performed attacks, which drives us to conclude that the ATSR routing rule succeeds in finding trusted near optimal paths even when the percentage of malicious nodes is high, balancing the distance and trust criteria. The threshold based routing rule is the second good candidate, while the case where the next hop neighbour is chosen among the most trusted applying secondly the distance criterion is leading to the poorest performance.

The number of the performed attacks (i.e. dropped packets) reflects the responsiveness of the routing algorithm. In other words, it shows after how many experienced attacks each node has detected its malicious neighbours and started avoiding them for forwarding. Taking into account that the transmission and reception of messages consumes significant part of the node power, it is highly desired to keep this performance parameter as low as possible. The results for this performance aspect are included in Figure 6, for the tuned versions ($k$ and threshold values leading to the best performance) of the routing rules.

A first clear conclusion from this figure is that ATSR and the threshold-based approach lead to the lowest number of experienced attacks. ATSR is outperformed only for 50% malicious nodes in the network. (The number of attacks observed for $W_d = 0.6$ is an order of magnitude lower than the attacks measured for $W_d = 0.8$ not included in the figure.) Comparing TF and SDDF rules, TF leads to more attacks than SDDF. Adopting the TF rule, the higher number of attacks is experienced for $k = 10$ which is consistent with the fact that for $k = 10$, lower delivery ratio is observed than for $k = 6$ for all tested penetration of malicious nodes, except for 50%. In this case, the performed attacks for $k = 6$ is higher and the delivery ratio (shown in Figure 1) is lower. Similarly, for SDDF rule, the number of attacks increases as the number of malicious nodes increase, since more cooperations are attempted before all the malicious nodes are detected. Lower $k$-values lead to lower number of experienced attacks.

7. COMPARISON FOR LARGE WSN AND DIFFERENT ATTACK TYPES

To check whether the conclusions reached above still hold for other network topologies and types of attacks, we performed simulation runs for a network consisting of 625 nodes (placed on a $25 \times 25$ grid), for different grey-hole and black-hole attackers penetrations uniformly distributed in the network.

The results in terms of normalised delivery ratio are tabulated in Table I. Comparing the delivery ratio values observed for 30% grey-hole nodes in a network of 625 nodes with the values for a network of 100 nodes (included in Figure 5), it is obvious that the delivery ratio is lower for the large network. This mainly stems from the fact that the
connections on average span more hops than in the small network (of 100 nodes), increasing the possibility of loss in an adversary environment. However, the relative performance remains the same, i.e. ATSR performs better than the other routing rules for the large network exactly as observed for the network of 100 nodes.

Similar conclusions are reached comparing the case where 50% grey-hole attackers exist in the network of 625 nodes and 100 nodes, respectively. The delivery ratio drops for the large network for all the tested routing rules while the ATSR still performs better. Also, exactly as expected, for higher number of attackers (50%) the delivery ratio is lower than for 30% of malicious nodes for the same network topology of 625 nodes.

To check the behaviour of the four routing rules, we changed the type of attack from grey hole to black hole. The delivery ratio observed in this case (included in the last row of Table I) is lower than the delivery ratio observed for 30% grey-hole attackers for all routing rules, since black-hole attackers totally block traffic, while grey-hole ones forward randomly some packets.

The important observation for the comparison of the routing rules is that the relative performance remains the same in all tested scenarios. This is due to the fact that in all cases we have assumed the same trust model which means that once the trust is evaluated, the relative performance depends on how this is combined with the pure routing factors (in our case, distance). For this reason, the experienced performance of the four routing rules for different conditions, e.g. assuming mobile nodes or different attacks such as modification or badmouthing attacks, mainly depends on the trust model. For example, as regards the support of mobility, the performance for mobile WSNs is better when the trust model includes an indirect trust exchange protocol which allows nodes to share their opinions about their neighbours [24]. To assess such behaviour differentiations, we have evaluated ATSR under two simulation scenarios pairs: one with only grey-hole attackers where we varied the weight values for each trust metric and another with a mix of nodes that perform grey-hole, modification, authentication and encryption attacks. In each scenario pair, ATSR 1 indicates $W_1 = 0.3, W_2 = 0.1, W_3 = 0.2, W_4 = 0.2, W_5 = 0.2$ and ATSR 2 corresponds to $W_1 = 0.25, W_2 = 0, W_3 = 0.25, W_4 = 0.25, W_5 = 0.25$ where $W_i$ the weights in Equation (3). The results in terms of delivery ratio are included in Figure 7 and clearly show that the difference in the delivery ratio becomes non-negligible only for malicious nodes penetration above 30%. When malicious nodes exceed 30%, the delivery ratio depends on the weight assigned to the trust metric based on which the relevant attack is detected. Thus, for example, in the case of grey-hole attackers, ATSR 1 performs better since the first two trust metrics which mainly target the detection of grey-hole attackers are assigned higher weights than in ATSR 2.

Focusing on the case where four different types of attacks are simulated, ATSR 1 and 2 perform almost similarly since the relevant trust metrics have equal weights ($W_1, W_2$ and $W_3$). Further elaboration on the relation between attack detection and the weights assigned to the trust metrics can be found in Ref. [30].

To this end, the conclusions that we reached based on extensive simulation results are summarised in Table II, where the four routing rules are assessed with regards to the achieved connectivity (which is quantified by the delivery ratio), the security and the path optimality which is directly translated in packet latency. The ATSR provides the higher delivery ratio in all tested cases (for $W_d = 0.6$) and provides connectivity even for high penetration of malicious nodes. However, it does not guarantee that the packets will never traverse a less-trusted node. This guarantee is only provided by the threshold-based rule which offers high delivery ratio as soon as nodes with trust exceeding the predefined threshold exist. Otherwise, a session can be blocked even if just one forwarding node has no trusted neighbours (i.e. neighbours with trust exceeding the threshold). The SDDF and TF provide lower delivery ratio than the other two rules, with SDDF offering higher path optimality while TF offering higher security guarantees. The drawback of SDDF and TF rules is that for different attacker penetrations, different values of $k$ parameter lead to the best performance. However, this cannot be easily changed (reconfigured) depending on the malicious nodes penetration at run time.

### Table I. The (normalised) delivery ratio measured for the four different routing rules for a network consisting of 625 nodes.

<table>
<thead>
<tr>
<th>Penetration of malicious nodes</th>
<th>ATSR</th>
<th>TF 6</th>
<th>SDDF 6</th>
<th>Thr 0.8</th>
</tr>
</thead>
<tbody>
<tr>
<td>30% grey hole</td>
<td>0.85</td>
<td>0.63</td>
<td>0.65</td>
<td>0.80</td>
</tr>
<tr>
<td>50% grey hole</td>
<td>0.78</td>
<td>0.52</td>
<td>0.57</td>
<td>0.64</td>
</tr>
<tr>
<td>30% black hole</td>
<td>0.80</td>
<td>0.56</td>
<td>0.55</td>
<td>0.78</td>
</tr>
</tbody>
</table>

### Table II. Qualitative comparison of the four routing rules.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Connectivity/delivery ratio</th>
<th>Security guarantees</th>
<th>Path optimality</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATSR</td>
<td>Guaranteed/high</td>
<td>Relative</td>
<td>Medium</td>
</tr>
<tr>
<td>Thr-based</td>
<td>Can be blocked/high</td>
<td>Strict</td>
<td>Medium</td>
</tr>
<tr>
<td>SDDF</td>
<td>Medium</td>
<td>Relative</td>
<td>High</td>
</tr>
<tr>
<td>TF</td>
<td>Medium</td>
<td>High</td>
<td>Low</td>
</tr>
</tbody>
</table>

**Fig. 7.** Performance results for different attack types and ATSR configuration.
8. CONCLUSIONS

In the unmanaged environment of WSNs, the list of security attacks addressing the routing procedure is very long. The establishment of trust relationships among nodes, exactly as in human societies, is a useful and effective tool. Once the trustworthiness of each node is defined, it has to be taken into account during routing decision making. We have defined four different ways to incorporate trust knowledge in location-based routing algorithms and we have evaluated them, in terms of delivery ratio in the existence of malicious nodes, packet latency and near optimal path length, using computer simulations under several conditions. Adopting trust (distance) criterion first to narrow the neighbour set and afterwards apply the distance (trust, respectively) criterion leads to two routing rules which exhibit performance highly depending on the narrowing factor and the penetration of malicious nodes. As this cannot be predicted, flexibly balancing trust with distance or setting a strict trust threshold appear more attractive. The threshold based rule offers strict security guarantees while under certain circumstances can lead to connection blocking. The best performance in all cases is achieved by the proposed ATSRS solution which adopts a weighted routing cost function to balance trust with distance. Although a configuration of this rule leading to better results in all case has been identified, we should keep in mind that it does not provide strict security guarantees. Summing up, the ATSRS routing rule succeeds in finding trusted near optimal paths even when the percentage of malicious nodes is high, balancing the distance and trust criteria while the threshold based routing rule is the second good candidate.

The investigation of different ways to incorporate trust in link-state routing protocols remains for further investigation while real life implementation work has already started to prove the feasibility of the approach.

ACKNOWLEDGEMENTS

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

REFERENCES


Trust-aware location-based routing in WSNs

H. C. Leligou et al.


AUTHORS’ BIOGRAPHIES

Helen C. Leligou received the Dipl.-Ing. and Ph.D. degrees, both in Electrical and Computer Engineering, from the National Technical University of Athens (NTUA), Athens, Greece, in 1995 and 2002, respectively. Her research interests lie in the area of protocol design for communication systems, access control mechanisms in broadband networks including HFC, PON, WDM, and core networks. Currently she is working on security protocols for wireless sensor networks. Her research results have been published in more than 80 scientific journals and conferences. She has participated in several EU-funded ACTS, IST and ICT research projects in the above areas. Since 2007 she is a lecturer at Technological Educational Institute of Chalkida.

Panagiotis Trakadas received his Dipl.-Ing. Degree in Electrical & Computer Engineering and his Ph.D. in the area of Electromagnetic Compatibility from the National Technical University of Athens (NTUA). Currently, he is working for the Hellenic Authority for the Information and Communication Security and Privacy as a security auditor. In the past, he has been with the Hellenic Aerospace Industry (HAI) as a Senior Engineer in military telecom networks, Technological Educational Institute of Athens as an Assistant Professor, and the National Technical University of Athens as a Senior Researcher. Dr. Trakadas has been actively involved in many EU-FP7 research projects and has published more than 45 papers in magazines, journals, books and conferences. He has been a reviewer in IEEE Transactions on Communications and IEEE Transactions on Electromagnetic Compatibility.

Sotiris Maniatis received a Diploma degree in Computer Engineering and Informatics from the University of Patras, Greece, in 1996, an MSc degree in Information Systems Engineering from UMIST, United Kingdom, in 1997, and a PhD degree from the School of Electrical and Computer Engineering of the National Technical University of Athens (NTUA) in 2002. Since 2005, he has been working as an ICT expert with the Hellenic Authority for Communication Security and Privacy. His primary research interests are secure communications, mobile computing, QoS and resource management, and next generation networks. He has received an academic excellence award from the National Scholarship Foundation and is a member of the Technical Chamber of Greece.
Panagiotis Karkazis received his degree in Electrical Engineering from Technological Educational Institute of Piraeus, Athens, Greece in 1999 and Msc in Data Communications and Networking from Kingston University, London, UK in 2005. Since then he is working as Software Engineer and he has experience in development of communication protocols for industrial and WSN applications. Also he has participated in EU-funded ICT research projects in trust and security issues for WSNs and the results has been published in scientific journals and conferences. Since 2007 he is teaching in Technological Educational Institute of Piraeus.

Theodore Zahariadis received his Ph.D. degree in electrical and computer engineering from the National Technical University of Athens, Greece, and his Dipl.-Ing degree in Computer Engineering from the University of Patras, Greece. Currently, he is the technical coordinator of the EC FP7 ICT projects: COAST, BeyWatch and VITRO. He is also heavily involved in the EU Future Internet Assembly (FIA) activities and leads the EC Future Internet Architecture (FIArch) group and the EC Future Media Internet Architecture Think Tank (FMIA-TT). Moreover, he coordinates as “caretaker” the FIA Future Content Network (FCN) subgroup. Since 1994, he has participated in many ACTS, ESPRIT, IST and ICT projects as Senior Researcher, Technical Manager or Project Manager. Dr. Zahariadis has published more than 80 papers in magazines, journals and conferences and he is the author of two books and many book chapters. He has been a reviewer and principal guest editor in many IEEE and ACM journals and magazines and he is a Technical Editor of the IEEE Wireless Communications Magazine.