Implementing a Trust-Aware Routing Protocol in Wireless Sensor Nodes

Theodore Zahariadis¹, Panagiotis Trakadas², Helen Leligou¹, Panagiotis Karkazis¹
¹ Technological Educational Institute of Chalkis, Greece
² Hellenic Authority for Communications Security and Privacy (ADAE), Greece

Abstract

Wireless Sensor Networks are gaining popularity due to the fact that they offer low-cost solutions for a variety of application domains. However, these networks are highly susceptible to attacks, due to both the open and distributed nature of the network, as well as the limited resources of the nodes, which dictate the implementation of sophisticated security frameworks. Trust-aware routing is an important direction in designing secure routing protocols for WSN. The common approach to provide trust-aware routing is to implement an efficient mechanism to evaluate the trustworthiness of neighboring nodes, based on a proper set of trust metrics. In this paper, we discuss several challenges addressed during the implementation of a trust-aware management scheme on commercial sensor nodes, as well as problems faced during the deployment of the test-bed network. As it will be discussed, the limited memory, computational power, energy resources and radio bandwidth of sensor nodes deeply impact the implementation strategy, while additionally, the realities of radio propagation, such as lossy and asymmetric links, require careful evaluation of the routing selection metrics.

Key Words: wireless sensor network, trust management, security.

1. Introduction

Wireless Sensor Networks (WSN) offer efficient, low-cost solutions for a great variety of application domains including military fields, healthcare, homeland security, industry control, intelligent green aircrafts and traffic control in smart roads [1]. Although networking and security technologies are in a mature stage, the limited sensor node resources in terms of memory space, processing power and energy availability, constrain the complexity of the security mechanisms that can be implemented, dictating the need for new protocol approaches design. Due to their distributed nature, WSNs are vulnerable to various attacks [2], including attacks targeting on the disruption of the routing procedure [3]-[5] which is accomplished in a cooperative, multi-hop fashion.

While the traditional (or the so called “hard”) security measures (e.g. encryption, authentication) are quite efficient in mitigating some types of attacks, there are some specific types of attacks that can be better handled by using a reputation and trust-based management scheme (as an example, we mention the selfish behaviour of a node). In other words, security and trust are tightly coupled and cannot be separated from one another. As mentioned in [6]: “Cryptography is a means to implement security, but it is highly dependent on trusted key exchange. Similarly, trusted key exchange cannot take place without requisite security services in place”.

The notion of trust, in the realm of network security, can be translated as a set of relations among entities (nodes) that participate in a protocol. These relations are based on the evidence generated by the previous interactions of entities within the protocol. In the literature, trust is interpreted as reputation, trusting opinion, probability, etc. Consequently, the evaluation of trust is based on several metrics, using different approaches, and thus it is very hard or even impossible to compare these trust-management approaches that lead to trust establishment between the nodes.

So, we clarify that in this text, reputation is defined as the opinion of one entity about another, while trust is the expectation of one entity about the actions of another.

Several trust management solutions have been proposed in the literature in the past. Most of these protocols have been studied only under simulation environment, assuming simplistic radio models, symmetric links, unlimited memory space and bandwidth, etc. In contrast, the WSN community demands solutions that work on commercial, resource-constrained and heterogeneous hardware platforms. In this paper, we present design and implementation issues of a trust-aware routing protocol, the differences between simulation platforms and test-bed environment, as well as results from test-bed environment, putting emphasis on the implementation issues.
2. Trust-aware routing protocol description

In this section, we present, in brief, a novel distributed trust-aware routing protocol, suitable for the demanding and highly unreliable WSN environment, which incorporates both direct and indirect trust evidence (Figure 1). This trust-aware routing protocol brings two important innovations: first, it defends against a wide set of attacks by monitoring multiple behaviour aspects (trust metrics) as seen in Table 1, and second, it incorporates energy-awareness which allows for better load balancing and higher resilience against attacks. Moreover, a routing cost function incorporating trust, energy and location information is derived to guide routing decisions.

The proposed trust-aware routing protocol requires three control messages, namely BEACON, REPREQ and REPRES, where the last two are needed to support reputation scheme (exchange of indirect trust information between neighboring nodes).

As regards the quantification of trust, for each monitored behavior listed in Table 1, a node \( i \) calculates a trust value regarding a neighboring node \( j \) for a trust metric \( m \) by dividing the number of successfully completed interactions to the total number of attempted interactions:

\[
T_{m}^{i,j} = \frac{S_{m}^{i,j}}{S_{m}^{i,j} + F_{m}^{i,j}}
\]

Afterwards, the trust values calculated for the monitored behaviors are combined in a weighted sum to produce the direct trust value:

\[
DT_{C}^{i,j} = \sum_{m} W_{m} * T_{m}^{i,j}
\]

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.

Coming to the reputation (indirect trust) exchange protocol which is used to accelerate the trust information build-up procedure, in the presented trust-aware routing protocol each node \( i \) periodically requests reputation information (REPREQ) from one randomly selected neighbour \( j \) per quadrant.

### Table 1: List of trust metrics

<table>
<thead>
<tr>
<th>Nr.</th>
<th>Trust metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Forwarding</td>
</tr>
<tr>
<td>2</td>
<td>Network-ACK</td>
</tr>
<tr>
<td>3</td>
<td>Data integrity</td>
</tr>
<tr>
<td>4</td>
<td>Node authentication</td>
</tr>
<tr>
<td>5</td>
<td>Data confidentiality</td>
</tr>
<tr>
<td>6</td>
<td>Reputation Responses</td>
</tr>
<tr>
<td>7</td>
<td>Reputation Validation</td>
</tr>
<tr>
<td>8</td>
<td>Remaining Energy</td>
</tr>
</tbody>
</table>

The number of received \( k \) reputation responses (REPRES) is summed up in a weighted manner with the weight representing the relevant trustworthiness \( (Np) \) of the node that provided it:

\[
IT_{C}^{i,j} = \frac{\sum_{p} (DT_{C}^{i,Np} * DT_{Np}^{j,i})}{\sum_{p} DT_{C}^{i,Np}}
\]

In this concept, every node can build a trust relation with its neighbours, based on actions (events) performed by other nodes in the neighbourhood. This is useful especially in cases where a node arrives in a new neighbourhood which is usually the case for mobile nodes.

Finally, the Total Trust value of a node \( i \) for a neighbour \( j \) is produced by combining direct and indirect trust values in the following formula:

\[
TT_{C}^{i,j} = C_{i}^{i,j} * DT_{C}^{i,j} + (1 - C_{i}^{i,j}) * IT_{C}^{i,j}
\]

where \( C_{i}^{i,j} \) 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 closer to the destination and is combined with the total trust value.
in the Routing Function:

\[ RF^{ij} = W_d^{ij} \cdot T_d^{ij} + W_t^{ij} \cdot TT^{ij} \]  

(5)

Where \( W_d \) and \( W_t \) represent the significance of distance and trust criterion, respectively, with \( W_d + W_t = 1 \).

Once the presented trust-aware routing protocol was designed, the efficiency in detecting and avoiding malicious nodes has been investigated for different types of attacks, different penetration of malicious nodes, different network density and different weighting factor values in JSIM simulation platform. In Figure 2, we present the packet loss ratio for weighting factor values shown in Table 2 and \( W_d = W_t = 0.5 \) for different percentage of malicious nodes (the network consists of 100 nodes). Although the results depend on the weight values, the general conclusion is that even when 50% of network nodes behave maliciously, the packet loss ratio is kept below 40%.

\[ RF^{ij} = W_d^{ij} \cdot T_d^{ij} + W_t^{ij} \cdot TT^{ij} \]  

(5)

### Table 2: The weighting factors values.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Weight value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Forwarding</td>
<td>0.3</td>
</tr>
<tr>
<td>Network acknowledgement</td>
<td>0.2</td>
</tr>
<tr>
<td>Integrity</td>
<td>0.2</td>
</tr>
<tr>
<td>Authentication</td>
<td>0.1</td>
</tr>
<tr>
<td>Confidentiality</td>
<td>0.1</td>
</tr>
<tr>
<td>Reputation response</td>
<td>0.1</td>
</tr>
<tr>
<td>Reputation validation</td>
<td>0.0</td>
</tr>
<tr>
<td>Energy</td>
<td>0.0</td>
</tr>
</tbody>
</table>

Also, in Figure 3, we present results from simulating performance of the trust-aware management scheme for different values of forwarding and network acknowledgement weighting factors.

Since one of the most restrictive factors for the deployment of WSNs is their limited energy, we have also paid special attention to prevent nodes being selected for forwarding systematically which leads to their energy exhaustion. This was achieved by taking into account the remaining energy of each neighbor when calculating its trustworthiness and then routing is decided based on both distance and trust-energy criteria. The simulation results show that the proposed scheme leads to load balancing effects which additionally makes difficult traffic analysis attacks.

As mentioned earlier, the proposed protocol includes a reputation exchange mechanism which allows newly activated nodes (or nodes that recently arrived in a neighborhood) to obtain trust information from their neighbors. However, as this introduces the need for message exchange, it increases the energy consumption. The performance benefits stemming from this indirect trust scheme were evaluated in order to allow for a careful trade-off between security and energy consumption.

To this end, the proposed protocol solution has been tested on efficiently detecting black-hole, grey-hole, integrity, authentication, bad-mouthing and selective forwarding attacks. Extensive JSIM simulation results can be found in [7]-[9].

Finally, the trust-aware routing protocol has been implemented in IRIS and MicaZ motes [10] (the code was written in TinyOS version 2.1).

### 3. Simulation platforms versus test-bed environment

In general, it has been verified that there are differences when evaluating the functional performance of the protocol in the simulation and the test-bed environment. The main reasons for the differences in the two different environments are:

- The energy consumption model either in JSIM or TOSSIM differs in the way that real sensors consume energy when transmitting data packets and control messages. In details, the energy model developed in the simulation platform is functioning differently from the real mote energy model by two factors: First, in TOSSIM and JSIM, all nodes start their operation from maximum battery energy, while motes have their unique (and of course different) battery energy. Second, the TOSSIM and JSIM energy consumption model acts quite differently from motes.
- In simulation, especially in the case of the JSIM package, the memory size and the execution
(run-time) environment does not reflect in a realistic way the restrictions of the real sensor nodes running the single-threaded TinyOS. Also, collisions that happen in the real network might affect differently the network performance than estimated in simulation, especially if the transmission rate of the periodic messages is high. Finally, the radiation pattern of the sensor nodes is different than that assumed in the simulation environments.

- The connectivity graphs of real test-bed deployment and simulation platforms, even for the same topology, are different. The reason is that transmission signal strength fluctuates in test-bed scenario (asymmetric or unstable links) while in TOSSIM or JSIM the transmission loss is constantly defined. Consequently, this has a direct impact on the neighbor list of the nodes and thus at the routing decision process.

- Due to the fact that all motes are using the same channel for transmission and reception, sometimes collisions occur. The number of collisions as well as their random existence differs from simulation to real test-bed cases. This has an impact on sniffing functionality of the code (forwarding metric and reputation response validation) and consequently to the routing decision process.

4. Implementation/Deployment problems

In this section, we classify the problems faced during deployment and code validation phase, into three categories: node problems that involve only a single node, link problems that involve two neighboring nodes and the wireless link between them, and path problems that involve three or more nodes and a multi-hop path formed by them. The test-bed (Figure 4) consisted of 20 IRIS motes in an indoor environment. The basestation was connected to a pc running linux. Also, a sniffer was present, providing a visual presentation of the network and the exchanged messages, as depicted in Figure 5.

Figure 4: A typical example of implementation evaluation testing against various attack types.

4.1. Node Problems

The implementation code should be bit-efficient in order to cope with the memory restrictions of the sensor nodes, especially in cases where many different functional components have to be implemented on a single mote. Thus, we tried to compromise the benefits stemming from implementing different characteristics of our protocol, depending on their RAM allocation. As a general conclusion, it can be stated that a RAM space of 4kB (e.g. in MicaZ) is roughly inappropriate, while a RAM space of 8kB (provided, for example, by IRIS motes) is adequate for implementing the predefined functions of the trust-aware routing protocol under study. In Table 3, a rough estimation of RAM and ROM space allocation per trust metric is shown (MAX_NEIGHBORS=10, RX_QUEUE=8, TX_QUEUE=8). It is mentioned that the limited number of buffer queue for transmitted and received messages was proved insufficient in specific cases during operation, causing loss of packets.

On the contrary, the ROM space (128kB) was proved adequate for the purposes of our trust-aware routing protocol.

Apart from the memory restriction, another general remark regards the battery lifetime of the nodes. IRIS and MicaZ motes operate on a pair of batteries that approximately supply 2200 mAh at 3V. A common node problem is node death due to energy depletion caused by “normal” battery discharge, mainly due to the increased amount of network traffic (frequently exchanged messages, constant routing paths to gateway, computational burden, etc). Low batteries often do not result in a “fail-stop” behavior of a sensor node. Rather, nodes show random behavior below a certain low battery level, where wrong sensor readings might be observed.
Table 3: RAM/ROM allocation per trust metric

<table>
<thead>
<tr>
<th>Trust Metrics</th>
<th>RAM (bytes)</th>
<th>ROM (bytes)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Integrity</td>
<td>868</td>
<td>48</td>
</tr>
<tr>
<td>Authentication</td>
<td>10</td>
<td>64</td>
</tr>
<tr>
<td>Reputation Response</td>
<td>77</td>
<td>332</td>
</tr>
<tr>
<td>Reputation Validation</td>
<td>90</td>
<td>180</td>
</tr>
<tr>
<td>Indirect Trust</td>
<td>374</td>
<td>2222</td>
</tr>
<tr>
<td>Network Ack</td>
<td>110</td>
<td>106</td>
</tr>
<tr>
<td>Forwarding</td>
<td>120</td>
<td>286</td>
</tr>
<tr>
<td>Remaining Energy</td>
<td>146</td>
<td>514</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>1795</strong></td>
<td><strong>3752</strong></td>
</tr>
</tbody>
</table>

Thus, special attention was paid on an energy-efficient technique in order to maximize the node life-time, through the trust-aware routing protocol specification (energy metric).

4.2. Link Problems

Test-bed experiments demonstrated a very high variability of link quality across time and space resulting in temporary link failures and asymmetric links that consequently affected packet loss and latency. In our trials, a percentage between 10-20% of all links have been observed to be asymmetric, with lower transmission power and longer node distance resulting in more asymmetric links.

Network congestion due to traffic bursts was another source of message loss. In our trials, a median message loss of 3-5% was observed due to this reason. Excessive levels of traffic bursts have been caused by accidental synchronization of transmissions by multiple sensors, especially due to the frequent transmission of control messages. In order to minimize this effect, several tests have been performed where the frequency of transmitted control messages was optimized.

Finally, during the tests, interference from other radio devices in the 2.4GHz frequency band (IEEE 802.11 and Bluetooth devices) has been observed. Thus, special care was paid on the selection of the proper channel for sensor communication, in order to improve the communication quality and liability (range, throughput and number of transmission/reception errors).

4.3. Path Problems

WSN applications rely on the ability to relay information across multiple nodes along a multi-hop path. In particular, our test-bed includes one sink node that disseminates queries or other tasking information to sensor nodes and sensor nodes deliver results back to the sink. Thus, it is important that a path exists from the sink to each sensor node, and vice versa. Two common problems in such applications are hence bad path to sink and bad path to node. In our test-bed, malicious nodes have been inserted in various scenarios, in order to evaluate the response of the trust management component functionality and measure the impact on packet loss and latency of the network. The results on our test-bed environment revealed that packet loss due to a bad or malicious route did not exceed 5%, while path stretching was only increased by 15-20% for most cases.

Finally, since routing loops are a common path problem, a Time-To-Live (TTL) counter has been implemented. It was proved that the use of a TTL counter was advantageous (message congestion was lower, energy depletion was avoided, etc), although it resulted in a slightly increased packet loss percentage.

5. Lessons learned from implementing the trust-aware routing protocol

As a general conclusion, we can state that the resources of a sensor node are adequate for building a powerful, attack-tolerant and energy-efficient trust management scheme, consisting of several implemented trust metrics.

However, the efficiency of each one of the metrics differs. After many tests, it was proved that the forwarding metric, even followed by a reduced weighting factor, was more efficient on the identification of a malicious node compared to the network ack metric. Apart from the forwarding metric, the remaining energy metric was also quite efficient as verified through test-bed experiments.

Regarding the implementation of the indirect trust mechanism, the main conclusion is that it, indeed, helps mobile nodes to find a trusted path to the destination in less time compared to relying only on the direct trust calculation (resulting in a slightly increased number of successfully delivered packets to destination), but the main drawback of the solution is the increase in energy consumption which is caused by the exchange and processing of the reputation and request messages. The trade-off between performance gain and energy consumption is a very important factor that should be considered before enabling the indirect trust mechanism, taking into account the requirements of a specific application.

Furthermore, it was obvious that the selection of the routing protocol to be applied is application-specific: in dense network deployments, the choice of a geographic routing protocol was appropriate, since the sensor nodes could only store information about neighboring nodes.

Apart from the efficiency of the trust metrics, we also investigated the impact of the weighting factors, coming to the conclusion that neither a greedy geographic protocol, nor a pure trust-aware management mechanism is efficient enough to solely identify malicious nodes in the network. The trust
management scheme must take into account the advantages stemming from the routing protocol and carefully apply trust metrics on it.

6. Research directions – Future work

In our future work, we will focus on the following three subjects:

- The utilization of all available channels provided by motes could definitely lead to a reduction of network congestions and improve the packet loss percentage. However, an innovative strategy on MAC layer is needed, taking into account the energy consumed by frequently tuning the RF module to different channels, as well as time synchronization problems.
- Also, a well-suited strategy for reducing the impact of asymmetric links must be developed. This problem can be solved by link layer handshaking (which is very expensive) or by using a lower transmission power when sending out messages to ensure that motes choose neighbors that are within a smaller radius, although this might lead to network separation in sparse deployments.
- Given the variation of WSN application domains and routing solutions, a trust-aware scheme should be scalable and modular to fit in many cases.

7. Acknowledgement

The work presented in this paper was partially supported by the EU-funded FP7 VITRO project.

8. References

[10] [www.xbow.com/](www.xbow.com/)