Towards the Future Internet
G. Tselentis et al. (Eds.)
IOS Press, 2009

doi:10.3233/978-1-60750-007-0-47

Securing wireless sensor networks towards a trusted "Internet of Things"

THEODORE ZAHARIADIS[1*], PANAGIOTIS TRAKADAS[1], HELEN LELIGOU[1], KOSTAS PAPADOPOYLOS[1], EVANGELOS LADIS[2], CHRISTOS TSELIKIS[2], CHARALAMPOS VANGELATOS[2], LIONEL BESSON[3], JUKKA MANNER[4], MICHALIS LOUPIS[5], FEDERICO ALVAREZ[6], YANNIS PAPAEFSTATHIOU[7]

1 Technological Educational Institute of Chalkida
2 Hellenic Aerospace Industry S.A.
3 Thales Communications France
4 University of Helsinki
5 Northern Venture Ltd
6 University Politechnico de Madrid
7 Telecommunication Systems Institute

Abstract: Wireless Sensor Networks (WSN) are quickly gaining popularity due to the fact that they are potentially low-cost solutions, which can be used in a variety of application areas. However, they are also highly susceptible to attacks, due to both the open and distributed nature of the network and the limited resources of the nodes. In this paper, we propose a modular, scalable, secure and trusted networking protocol stack, able to offer self-configuration and secure roaming of data and services over multiple administrative domains and across insecure infrastructures of heterogeneous WSNs. The focus is on trusted route selection, secure service discovery, and intrusion detection, while critical parts of the security functionality may be implemented in low-cost reconfigurable hardware modules, as a defense measurement against side channel attacks.

Keywords—security, wireless sensor networks, trust model, service discovery

1. Introduction

The past few years, Wireless Sensor Networks (WSN) (also called Web of Sensors) have demonstrated great potential, as integral components of solutions, applicable in a variety of domains, such as environmental observation, surveillance, military monitoring, smart (home) environments and ambient assisting living. The evolving "Internet of Thing" raises even more the already high hopes of WSN and has attracted the interest of the research community worldwide.

Before their wide deployment however, WSNs have to solve some significant problems, including energy management and security [1] [2]. In fact, the initial driving impetus for the development of sensor networks has been military applications, where

[*] Contact: Th. Zahariadis, TEI of Chalkida, Psachna, GR34400, Greece, zahariad@teihal.gr

security requirements are at their highest. Strong security requirements for such applications are often combined with a hostile and physically unprotected environment. For commercial applications of WSNs, the issue of privacy protection is as important as secure and reliable functioning of a network.

The work presented in this article is carried out in the framework of the AWISSENET (Ad-hoc PAN and WIreless Sensor SEcure NETwork) project, which is partially funded by the European Commission Information and Communication Technologies Program. The project targets the design, implementation and validation of a scalable, secure and context-aware networking protocol stack, able to offer self-configuration and secure roaming of data and services over multiple administrative domains and across insecure infrastructures of heterogeneous ad-hoc & wireless tiny sensory networks. The rest of the paper is organized as follows: we first discuss the intricacies of the problem. In section 3 we present our approach to enhance security in routing, service discovery and intrusion detection, while in section 4 we present the test bed. Finally, in section 5, conclusions are drawn.

2. The intricacies of securing wireless sensor networks

Security in WSN denotes protection of information and resources from attacks and misbehaviors, while maintaining an acceptable level of operation even in the case of adverse conditions. The security requirements list is too long, but unfortunately the same applies for the security attack list [2]. Several countermeasures have been proposed addressing specific types of attack; however, only a few proposals try to address multiple attacks, the main reason being the limited resources of sensor nodes.

WSNs share similarities and differences with ad-hoc wireless networks. The main similarity is the multi-hop communication nature, while the main differences are the usually much larger number of nodes and the node constraints in computational, transmission, energy and memory/storage resources. Moreover, WSN are often deployed in open, potentially harsh environments, where they are left unattended for a long period of time after their deployment, allowing physical attacks, such as node capture and tampering [3]. So, the possible attacks range from the physical layer up to the application layer, where aggregation and in-network processing often require trust relationships, between sensor nodes that are not typically assumed in ad-hoc networks. The impedimenta in securing WSN can be classified in two main categories: those introduced by the restricted node architecture and those stemming from the wireless media and the specific sensor network characteristics, as shown in Table 1.

Table 1: Restrictions stemming from the node and the network characteristics

Node restrictions	Network - channel restrictions
Low Data rates and small packet size impede the exchange of extra information needed to implement security schemes.	Unreliable communication due to the unreliable of low capacity wireless link with transmission collisions.
Limited Processing Power: 8-bit or 16-bit processor architecture, clock up to 8MHz.	The lack of central infrastructure and the unattended operation of WSN obstructs the implementation of well-established security techniques (e.g. PKI).

Restricted Energy Resources: Battery powered sensors (in many cases not even rechargeable)	The WSN topology is characterized by a high scale in terms of the number of the participating nodes and in terms of the network density. Also, topology changes occur due to random battery outages and link failures.

Therefore, in WSN the increased vulnerabilities mandate the design and implementation of a secure network protocol stack taking into account the severe limitations which are inherent in the restricted WSN environment.

3. WSN Security Approach

To efficiently address the security problem in WSNs, we propose a modular, secure sensor node "toolbox", by addressing three key research topics: a) discovery, evaluation and selection of trusted routes, b) secure service discovery, and c) intrusion detection, intruder identification and network recovery. Special emphasis is placed on reducing the footprint, the power consumption and the operating system requirements of the toolbox, to render it adaptable to a large variety of mobile/nomadic devices and tiny sensor nodes, and highly secure against side-attacks.

3.1. Discovery, evaluation and selection of trusted routes

Most sensor nodes have limited communication capabilities and rather short transmission range. Thus, in most cases, they communicate using multi-hop forwarding schemes: they have to forward the packets from node to node, until they reach their final destination. Moreover, WSN have a time-varying networking topology: either because the sensors are randomly deployed or they are moving, or because they are battery powered and each sensor's lifetime may vary based on the networking activity, dramatically changing the WSN network topology anytime. A wide variety of routing algorithms has been proposed in the literature, efficiently dealing with both the multi-hop forwarding communication and the dynamic topological changes of the WSN.

From a security point of view however, these characteristics turn WSN into an extremely vulnerable environment. A significant number of security attacks target the routing procedure, where malicious nodes either deny to forward their neighbors' traffic or on purpose advertise fake routes to attract traffic (to forward it to a colluding adversary or just drop it) or declare a fake identity or even modify the traveling messages, both carrying user data and routing protocol information [5]. Hence, the communication security depends heavily on the proper choice of the path used to reach the destination; thus it is important for a node to know the reliability of a route. To achieve trusted routing, it is necessary to design and implement a trust management system to compute the trustworthiness of the participating nodes and detect a node that is misbehaving, either faulty or maliciously. This information can then be exploited to specify a protocol for secure path selection.

We propose a secure routing mechanism combining a geographical routing protocol with a decentralized trust management scheme which can incorporate traditional security measures (e.g. encryption) to safeguard data integrity, confidentiality and node authentication in order to mitigate routing attacks identified in WSN deployments. The adoption of geographical routing prevents a number of routing attacks dealing with advertisement of attractive paths, since in geographical routing the

nodes only announce (broadcast) their position to their one hop neighbour. This intricacy is also the key for scalability and efficient mobility support. As regards the proposed trust scheme, each node is responsible for computing its own trust value for each neighboring sensor node in the network, either collecting events from direct relations with this node (first-hand information), or by collecting indirect trust values from its one hop neighbours (second-hand information). In this concept, every node can build a trust relation with its neighbors, based on the collection of actions (events) performed by other nodes in the neighborhood.

The types of events that we propose each node should monitor are:

Packet forwarding*:* This metric is based on overhearing that a packet has been forwarded (event type E1) and the "packet precision- integrity" (event E3) is checked.

Network layer ACK. Each node will monitor whether its message has reached a higher layer node in the proposed architecture by counting the network layer acks received (E2). This is a powerful tool especially when combined with cryptography.

Authentication – Confidentiality – Integrity. A node can collect trust information about neighbouring nodes during interactions regarding the proper use of the security measures applied. This behaviour as well as the result of the authentication process is coded as packet Precision-Integrity (event type E3), Authentication (event type E4) and Cryptography-Confidentiality (event type E5).

Reputation Scheme. Another way of evaluating the behaviour of a neighbour is by observing its behaviour: If it replies to reputation request messages, it is rated high due to its willingness to participate in the procedure (event type E6). If node C has proposed node B but interaction between A and B is unsuccessful, then A will decrease the direct trust value of node C, since its reputation value about node B has been proven false (event type E7 value).

Log History. This type keeps the success or failure of the last n events (where n=16 or n=32). This metric aims in protection against on-off attacks, where malicious nodes try to confuse their neighbours by partially forwarding their messages. It also allows for the detection of the beginning of a malicious behaviour, since when the trust value drop, it can easily be checked whether this is due to past or recent behaviour.

Other Events. There is a large set of network events, ranging from hardware-related situations to application layer behaviours that can be used as inputs for the trust management system. For example, for geographic routing protocols, some metrics like the distance of each node to the sink node (E11) may be used.

Table 2: Direct Trust Table structure

Trust metric	Maintained Information	
Forwarding (E1)	# of Success	# of Failures
Network-ACK (E2)	# of Success	# of Failures
Packet precision- Integrity (E3)	# of Success	# of Failures
Authentication (E4)	# of Success	# of Failures
Cryptography-Confidentiality (E5)	# of Success	# of Failures
Reputation RES (E6)	# of Response	# of request
Reputation Validation (E7)	Value	
Remaining Energy (E8)	Value	
Network ACK History Log (E9)	1 0 1 1 0 1 0 0 1 1 0 1 0 1 1 1	
Number of Interactions (E10)	Value	
Distance to the sink node (E11)	Value	

The structure of the Trust Table that stores the trust values is shown in Table 2. Each node with k neighbouring nodes will store k Trust Tables. Thus, the table size should be as small as possible, while keeping the most important information. In order to take the final forwarding decision, the trust values will be combined with factors like the distance to base station, number of hops to base station and node confidence. This is outside the scope of the current paper. In a heterogeneous sensor environment a subset of the above described events can be monitored and used to evaluate a node's trustworthiness based on different sensor node types and capabilities.

3.1.1. Direct Trust Evaluation

For each one of the first 6 events of Table 1, node's A Trust regarding node B, i.e. $T_i^{A,B}$, can be calculated:

$$T_i^{A,B} = \frac{a_i S_i^{A,B} - b_i F_i^{A,B}}{a_i S_i^{A,B} + b_i F_i^{A,B}} \quad (1)$$

Where: $Si^{A,B}$ and $F_i^{A,B}$ are the successful and failed type i events that A has measured for B, a_i and b_i represent the weight/significance of a success vs. failure and their values will be evaluated using computer simulations.

For the History Log (E9), we propose a simple pattern matching technique which will help towards either calculating the trust value or categorizing the neighbouring nodes activity. The number of interactions (shown as E10) is a measure of confidence. A high confidence value means that the target has passed a large number of tests that the issuer has set, or that the issuer has interacted with the target for a long time, and the node is sure that the value of the neighbouring node is more certain. The algorithm of trust evaluation is more sensitive in the beginning of the interactions period (since confidence value is small, one fault should have a large impact in trust value), while as confidence value increases, the impact (either on positive or negative events) is smoother. Thus, we define a confidence factor, like in the next equation:

$$C^{A,B} = 1 - \frac{1}{noi + a_{10}} \quad (2)$$

Where n_{oi} indicates the number of interactions with B and a_{10} is a factor whose value will be evaluated during simulation testing. This confidence factor can be proved useful, especially during the beginning of network operation.

In case of geographic routing algorithms, a proper metric is the distance of the neighboring node to the sink (E11). The closer a node to the sink, the greater the value added to the final direct trust of the node.

Finally, node's A Direct Trust value for its neighboring node B, i.e. $DT^{A,B}$ with k event types (in our case k =10) can be calculated according to the following equation:

$$DT^{A,B} = C^{A,B}\left(\sum_{i=1}^{k} W_i * T_i^{A,B}\right) \quad (3)$$

Where: W_i is the weighting factor for each one of the k event types, $Ti^{A,B}$ is node's A trust value of event i regarding node B. The use of the weighting factors is a very important feature of the adopted trust model. By using these weighting factors, during the simulation and validation process, we'll be able to categorize the severity of each one of the events that will have a different impact on the direct trust value.

3.1.2. Indirect trust/Reputation evaluation

There are several cases where a node (e.g. node A) needs the trust opinion of its neighbouring nodes (e.g. node C, D, E) regarding a specific node (node B). Examples of such cases may be the discovery of a new node appeared during a HELLO message or when direct trust value of node B is neutral (its value is neither large nor small). In the proposed trust model, a node A may find the indirect trust/reputation value of a node B i.e. the $IT^{A,B}$ by combining the direct trust values (reputation values) of its neighboring nodes, as shown in the following equation:

$$IT^{A,B} = \sum_{j=1}^{n} W(DT^{A,N_j}) DT^{N_j,B} \quad (4)$$

Where, n is the number of neighbouring nodes to A, N_j are the neighbouring nodes to A, $DT^{Nj,B}$ is node's N_j reputation value of node B, $W(DT^{A,Nj})$ is a weighting factor reflecting node's A direct trust value of node N_j

As in the previous section, we use different weighting factors for each node regarding the events described above. For example, if node's C direct trust value (evaluated by node A) is large and also node C is frequently sending responses to node's A requests, then its weighting factor is large. The reputation value $DT^{Nj,B}$ that the neighbouring nodes propagate to the interested node are kept to the Reputation – Indirect Trust Table, thus the interested node can check the correctness of their answers on next route discovery phase and modify the direct trust values of the neighbours $W(DT^{A,\ Nj})$ accordingly (e.g. increase the direct trust value of a node who gave a reputation that was proved correct). This is the reason of the direct trust value selection, instead of the sum of direct and indirect trust values. The metrics that allow node A to evaluate node's B trustworthiness in this case are the node's direct trust value, which includes its responsiveness in the reputation scheme implementation as well as the provided reputation value.

3.1.3. Total Trust evaluation

The total trust evaluation node A of node B, i.e. $TT^{A,B}$ is performed by applying the following equation:

$$TT^{A,B} = W(DT^{A,B}) DT^{A,B} + W(IT^{A,B}) IT^{A,B} \quad (5)$$

Where $DT^{A,B}$ and $IT^{A,B}$ are A's direct and indirect trust values of *B*, $W(DT^{A,B})$ and $W(IT^{A,B})$ are a weighting factors reflecting A's direct and indirect trust value of *B*. Since A can be sure only about the first-hand information, the weighting factor of the Direct Trust Value will be larger than the weighting factor of the Indirect Trust value.

3.2. Secure Service Discovery

Automated service discovery is an important functionality in a dynamic environment such as WSNs, e.g., sensor nodes need to find where the sinks are. Yet, the variable connectivity of the nodes coupled with a hostile environment may cause non-uniform or even false service information propagation. In such environment, the service discovery scheme may be based on a push-style information dissemination method, or on pulling information out of the network when needed. In the former case,

client nodes passively monitor the network and get to know about possible services, e.g., this is how IPv6 Router Advertisements work. Client nodes can also actively query the network for services; in IPv6, a node can send a Router Solicitation and force routers to tell about them. We can also use hybrid controlled dissemination where information about a certain service is not stored everywhere in the network, but at one, or a handful, of nodes; this can reduce the signaling load in the network but brings new authorization concerns, e.g., how can we trust the information coming from some intermediate node. The discovery, whether passive or active, can be further enhanced through coupling with route discovery; we can piggy-back service announcement and query messages in routing protocol messaging and thus, at the same time, gain knowledge of routes and available services.

Important and very difficult challenges in service discovery are the authentication of parties involved in the service discovery, and the confidentiality of the service signaling. Even if the content of a service is public information, we still want to confirm the source of the information and make sure the content is valid. For example, a news service or announcements at an airport are public information, but users probably want to be sure that the source is truly the news company or the airport authority, and not someone fooling around on purpose, or even trying to send unsolicited spam. In other services, only authorized requesters are allowed to receive an answer to a query and the requester needs some guarantees that the service is valid. The attacks against a service discovery system are listed in Table 3. Their detection is difficult. In the fake services case, we may not know where the information was altered, while in the Advertisement and query flooding case can be coupled with Cybil attack which makes it harder to identify. Finally, Listening & Profiling is a passive attack and in general we can not detect it.

Table 3: Attacks against service discovery

Attack	The attacker actions	Consequences
Fake services	An attacker responds to service queries even if it doesn't have the service or provides misleading information	Battery drain, unintentional Denial of Service
Advertisement and query flooding	An attacker sends massive number of advertisements or queries	The network spends resources in forwarding the messages The recipients spend CPU cycles and energy in receiving and processing the messages
Listening & Profiling	The attacker observes and profiles both the service provider and the client. By passively listening to the communication	Sensitive information is received by an unauthorized entity

The simple solution here is to have the right certificates and encryption keys at the receiver to verify or decrypt information. If the communication is fully encrypted, the attacker must first fight the encryption before anything else can be done; thus, only some sort of flooding attacks can be performed but if we assign per-host rate limits in routing we can reduce the effect. The more challenging situation is with unencrypted messaging, when we need some mechanism to get the right certificates.

The deployment scenarios in WSNs are very challenging, since we have different sensors and need to make services available between them. Typical fixed, or even ad-

hoc, network protocols can not be employed. Thus, we investigate and work on a hybrid system, which includes alternative signaling mechanisms, proxies and information caching, coupled with a trust mechanism between entities. Since routing in this environment has also similar challenges to the above, we investigate ways to couple routing and service discovery together and use a unified trust management scheme to counter the various attacks.

3.3. Intrusion detection, intruder identification and recovery

Key management, secure routing and secure services can be considered as a first line of defense, aimed at preventing malicious nodes to break into the network or to retrieve confidential information. However, there is a non-negligible possibility that an intruder, finally becomes successful. Thus, a second line of defense is needed, able to detect third party's attacks and raise alarms, even if the attacks haven't been experienced before. The proposed WSN Intrusion Detection System (WIDS) takes care of this role. WIDS differ in many ways from the one used in legacy networks. In order to achieve an efficient, secure and lightweight WIDS, the proposed system uses innovative architectures and algorithms that we outline hereafter.

Network Architecture. Usual IDS are typically *stand-alone IDS*, where each node runs an independent intrusion detector. Such systems are very limited in AWSNs, since local audit data is not enough to have a good comprehension of what is happening on the network. Cooperation between the different nodes is compulsory in order to achieve efficient detection, because local evidences are inconclusive. Since the network infrastructures that AWSNs can be configured to are either flat or multilayered, the same approach can be used for intrusion detection systems. *Hierarchical IDS* are systems where specific nodes are in charge of monitoring their neighbours, with various level of cooperation between cluster heads [5]. *Distributed IDS* meet the decentralized nature of ad hoc wireless sensor networks, where each node is responsible for collecting local audit data, and this knowledge is shared globally in order to carry out a global intrusion detection system [6], [7]. We propose a mixed approach, where the WIDS inside a cluster will be fully distributed, and cluster heads are responsible for exchanges and decisions at the upper level.

Collecting audit data. Data is collected by local agents analyzing local sources of information, which can be hardware or network based. Due to the wireless, ad-hoc nature of the network, nodes don't only analyse packets sent to them, but can also overhear traffic passing from a neighbouring node and act as a watchdog, detecting nodes forwarding selectively packets, or modifying them [8].

Intrusion Detection. Detection techniques can be either *Misuse detection* (where audit data is compared with known attack patterns), *Anomaly detection* (detect when the network behaviour differs from 'normal' behaviour, established via automated training) and *Specification-based detection* (similar to the former one, but the correct behaviour is manually defined) [7]. The proposed WIDS will have a modular software architecture where one can plug different detection mechanism (available as plug-ins), depending on the hardware of the nodes and what one intends to monitor.

Detection & Recovery. Once a local IDS agent has raised an alarm internally, the next question is who is going to make the final decision that a node is effectively an intruder. *Independent decision-making systems* are usually used in cluster-based architectures because they leave the decision that a node is effectively an intruder to certain nodes. The alternative solution is *Cooperative Intrusion Detection Systems*.

When an attack seems to have been detected, the node appeals to neighbouring nodes in order to output a global decision. The proposed WIDS uses a mixed approach, being cooperative inside a cluster, but relying on cluster heads at the upper level. Once a node has been assessed as malicious, the alert is forwarded to the modules in charge of isolating it, like the routing module and secure service discovery.

3.4. Highly secure nodes

The nodes in a sensor network are by nature distributed and thus, in the vast majority of the cases, they are very vulnerable to side attacks. These attacks are based on measuring characteristics of the processing activity on a node such as power consumption, electromagnetic emission, timing. By analyzing those measurements the attacker may recover all or part of the secret information stored in the node (e.g. the key used in the majority of the security algorithms). AWISSENET designs a new node architecture which utilizes the newly introduced extremely low-cost and low power Field Programmable Gate Arrays; this node is practically invulnerable to such side attacks. The implementation will be mainly based on two methods: Dual Rail encoding and Masking [10]. These implementation techniques, together with architectural design methodologies such as spreading processing tasks in random time periods, render the measurements performed by the side attacker useless for him/her.

4. The AWISSENET test bed

The efficiency of the proposed toolbox will be first assessed through exhaustive simulations and then the system will be validated in a trial involving 100 sensor nodes [10]. The trial consists of different "sensor nodes islands" involving different technologies which will be linked together using an IP network. As shown in Figure 1 we are setting up 4 different linked scenarios: one homogeneous and one heterogeneous mote islands, one RFID island and one multimedia island.

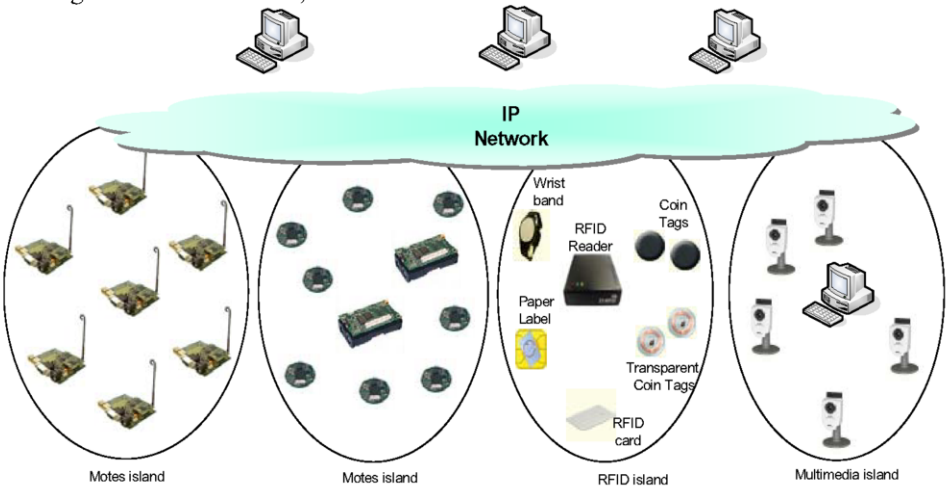

Figure 1: The AWISSENET Trial architecture

The aim of the trial is twofold: demonstrate the correct and efficient working of the technologies against attacks and prove the applicability to Personal Area Network and

sensor application scenarios. We aim at validating our developments in 4 different environments: industry, home, roads and disaster recovery. For this purposes each island will be equipped with adequate sensor types in each node which can be used to validate the environment we are testing. For example, the multimedia island in the home environment using smart cameras or microphones attached to the nodes can be used for creating a trustworthy and secure surveillance system which can demonstrate the applicability of the proposed solution. For the security validation, we are contemplating also the testing of cross domains and cross island communications which will give the final conclusions of the reliability and trustworthiness of the solutions described in the paper.

5. 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. We propose to implement and pack a set of security-aware protocols from the network to the application layer in a flexible security toolbox which can then be used in a variety of wireless devices [11]. The goal is to efficiently defend against a great number of attacks including side channel attacks focusing on those dealing with service discovery, routing, and intrusion detection. Our concept will be validated through a large and heterogeneous test-bed.

Acknowledgment: The research leading to these results has received funding from the European Community'sSeventh Framework Programme (FP7/2007-2013) under grant agreement n° 211998 (AWISSENET project).

References

[1] K. Papadopoulos, S. Voliotis, A. Ktena, P. Trakadas, Th. Zahariadis, "Security Aspects in Wireless Sensor Networks," Int. Conference on Telecommunications and Multimedia (TEMU 2008), Ierapetra, Crete, Greece, 16-18 July 2008

[2] Ivan Stojmenovic, "Handbook of Sensor Networks: Algorithms and Architectures", John Wiley & Sons, 2005, Ch.1.

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

[4] Bounpadith Kannhavong, Hidehisa Nakayama, Yoshiaki Nemoto, And Nei Kato, Abbas Jamalipour, "A survey of routing attacks in mobile ad hoc networks", IEEE Wireless Communications, October 2007, pp. 85-91.

[5] O. Kachirski, R. Guha, D. Schwartz, S. Stoecklin, and E. Yilmaz, "Case-based agents for packet-level intrusion detection in ad hoc networks," in Proc. of the 17th Int. Symposium on Computer and Information Sciences. CRC Press, October 2002, pp. 315–320.

[6] Ioanna Stamouli, "Real-time Intrusion Detection for Ad hoc Networks", Master of Science dissertation, University of Dublin, 2003.

[7] K. Ioannis, T. Dimitriou, F. C. Freiling, "Towards Intrusion Detection in Wireless Sensor Networks", 13th European Wireless Conference, Paris, April 1997

[8] R. Roman, J. Zhou, J. Lopez, "Applying Intrusion Detection Systems to Wireless Sensor Networks", Consumer Communications and Networking Conference, 2006, pp. 640-644.

[9] K. Tiri and I. Verbauwhede, "A Logic Level Design Methodology for a Secure DPA Resistant ASIC or FPGA Implementation," in Proc. of Design Automation and Test in Europe Conference (DATE 2004), pp. 246-251, 2004

[10] P. Trakadas, T. Zahariadis,H.C. Leligou,S. Voliotis, K. Papadopoulos, "AWISSENET: Setting up a Secure Wireless Sensor Network," 50th International Symposium ELMAR-2008, focused on Mobile Multimedia, Zadar, Croatia, 10-13 September 2008, pp. 519-523

[11] K. Papadopoulos, S. Voliotis, H.C. Leligou, D. Bargiotas, P. Trakadas, Th. Zahariadis, "A Lightweight Trust Model for Wireless Sensor Networks," Numerical Analysis and Applied Mathematics (ICNAAM 2008), Kos, Greece, 16-20 September 2008, pp.420-423