Trust and Reputation Systems for Wireless Sensor Networks

Rodrigo Román, M. Carmen Fernández-Gago, and Javier López

Department of Computer Science, University of Malaga, 29071, Málaga, Spain
{roman,mcgago,jlm}@lcc.uma.es

1 Introduction

The concept of trust has become very relevant in the late years as a consequence of the growth of fields such as internet transactions or electronic commerce. In general, trust has become of paramount importance for any kind of distributed networks, such as wireless sensor networks (WSN in the following). By considering trust as a factor to take into account on the relationship between two peers, it is possible to deal with the inherent uncertainty of the cooperation process. Differing on the underlying model trust management systems are classified into credential-bases trust management systems (i.e. based on the identity of a node) or behaviour based trust (i.e. based on the actions of a node).

From their humble beginning, WSN have evolved into a useful network paradigm applicable to many existing problems, such as environmental and structural monitoring, e-Health, and many others. In these networks a set of resource-constrained devices, called sensor nodes, measure the physical information (e.g. temperature, light) of their environment. Later, they work collaboratively to send those measurements using a wireless channel to a central device, called base station. Their importance is indicated by the increasing number of prototypes and research projects that take advantage of their specific capabilities.

Trust management systems for WSN could be very useful for detecting misbehaving nodes (faulty or malicious) and for assisting the decision-making process. Very little has been done so far in the area of trust management systems for WSN [16, 13]. Most of the work on this field has been made in the last few years. Big efforts, however, have been made in related areas such as P2P and Ad-Hoc networks [31, 46]. Thus, some of the approaches adopted for WSN try to imitate those for Ad-hoc or P2P networks [5, 47]. However, this is not always possible due to the difference in the features of these networks. For a start, the computational power and energy-constraint that reign in WSN make very hard to adapt the systems of Ad-Hoc networks. The size of the networks also becomes an issue. P2P networks are usually large in size of nodes whereas this is not always the case in WSN.

The important aspects to be considered when designing a trust management system for WSN is the type of problem that the system aims to solve. Thus, the nature of this problem determines the nature of the information that should be gathered and used in order to derive trust and reputation. It also determines

how the overall architecture of the system should be, how the information that the system needs should be gathered, and how the reputation and trust values should be obtained.

In this chapter of the book, we try to give a general overview of the state of the art on trust management systems for WSN and also try to identify the main features of the architectures of these trust management systems.

2 Trust Management

The concept of trust derives from sociological or psychological environments. Trust is an essential factor in any kind of network, social or computer networks. It becomes an important factor for members of the network to deal with uncertainty about the future actions of other participants. Thus, trust becomes specially important in distributed systems or internet transactions.
Even though there is not a consensus on the definition of trust, it is usually defined in terms of a **trustor** (the subject that trusts an entity or a service) and a **trustee** (the entity that is trusted).

The term trust has been used with a variety of meanings [28]. The Oxford Reference Dictionary defines trust as

> 'the firm belief in the reliability or truth or strength of an entity.'

Usually, trust management systems can be classified into two categories: *credential-based trust management systems* and *behaviour-based trust management systems*. This classification is based upon the approach used in order to establish trust among the peers of a system.

Credential-Based Trust Management Systems In this type of systems, peers (or nodes) use credential verification in order to establish trust with other peers (or nodes). The primary goal of credential-based trust management systems is to enable access control. Therefore their concept of trust management is limited to verifying credentials and restricting access to resources according to application-defined policies. A peer requests for access to a restricted resource. The access is controlled by a resource-owner that provides access only if it can verify the credentials of the requesting peer. Trust of the requesting peer in the resource-owner is not usually included. Thus, this type of systems is useful when there is an implicit trust in the resource-owner. However, these type of systems do not incorporate the need of the requesting peer to establish trust on the resource-owner. For this reason they are not very good trust management solutions for all decentralized systems. Examples of credential-based trust management systems are PolicyMaker [12], its successor, KeyNote [11] or REFEREE [14].

Behaviour-Based Trust Management Systems These type of systems are also called experience-based. In these models an entity trusts another entity based on past experience or behaviour. Thus, entities can perform evaluation on the other entities based on these features. These systems are mainly based on the concept

of *reputation*, which is quite related to the concept of trust. As it happens with the term trust, there are several definitions of reputation. Abdul-Rehman and Hailes [4] define reputation as an expectation about an individual's behaviour based on information about or observations of its past behaviour. Jøsang et al [18] define reputation as a mean of building trust; one can trust another based on its good reputation.

There are some basic properties that any reputation-based trust model should fulfil regardless their field of application.

- The model of computation.
- The metrics. Usually these values are ranged between 0 and 1 or -1 and 1. They express the reputation of an entity as it is provided by a reputation manager. The values given can be discrete or continuous. Continuous values are considered more expressive than discrete ones.
- Type of reputation feedback. The information collected can be positive or negative. Some systems are based on negative or positive information whereas others are based on both types.
- Reliability.

One of the first attempts to build a trust system based on reputation for e-commerce online applications was SPORAS [48]. In this system users rate each other after a transaction with values from 0.1 for terrible, to 1 for perfect. Then the reputation values can be updated over time according to a SPORAS formula. Whereas SPORAS provides a global reputation value, HISTOS, developed by the same authors, takes also into account the standards or considerations of different groups within the social network. REGRET [33] is a reputation model in the context of agents. In order to obtain the reputation values the authors consider individual reputation which is the reputation value computed directly from the agent's impression database, social reputation which is the trust value derived as a consequence of the relation of the agent with a group of agents. At last, they also consider ontological reputation which is the reputation obtained from different concepts.

Based on beliefs, Jøsang proposes a subjective logic [19] in order to derive reputation values. Other systems use some probabilistic methods such as the Beta function [17].

Trust management for WSN is not a very explored area. For this reason first we will make a survey on the existing methods for similar networks such as Ad-Hoc and P2P networks.

2.1 Trust Management Systems for Ad Hoc Networks

In [25] the authors present a trust model for mobile Ad-hoc networks that can be used in a dynamic context within the routing process. Initially, each node is assigned a trust value according to its identity. For instance, if no information is available about the trustworthiness of a node the assigned value will be *unknown*. Each node records the trust levels about their neighbours. Then, by using simple, logical calculations similar to averages a node i can derive the trust level of

node j, $TL_i(j)$. In [46] secure routing is also considered but the way of assigning the trust levels is carried out by evaluation of nodes over other nodes. Trust is evaluated considering factors such as statistics, data value, intrusion detection or personal reference to other nodes. The trust evaluation values, $TE(i, j)$, are stored in a matrix. The final trust value is calculated via a linear function that uses the values stored in the matrix. Reputation is considered in [31] as a way for building trust. The mechanism builds trust through an entity called the *trust manager*. An important part of the trust manager is the reputation handling module. Each node monitors the activities of its neighbours and sends the information to the reputation manager. Then, the information is passed to the reputation handling module and the reputation values are obtained via simple metrics. Zhu *et al* [50] provide a practical approach to compute trust in wireless networks by viewing any individual mobile device as a node of a delegation graph G and mapping a delegation graph from the source node S to the target node T into an edge in the correspondent transitive closure of the graph G, from which the trust value is computed.

2.2 Trust Management Systems for P2P Networks

PET [47] is a personalized trust model that evaluates risk and reputation separately in order to derive trust values. Reputation is also used as a way to obtain trust in [5]. In this work, when an agent wants to evaluate the trustworthiness of another agent, it starts to search for complaints on it. Once the data about the complaints is collected, trust can be assessed by an algorithm introduced by the authors. Bayesian networks have also been used [8, 43]. Other approaches [39] use statistics methods such as standard deviation and mean in order to detect anomalies or malicious behaviour of peers. TrustMe [37] is a secure and anonymous protocol for trust management. This protocol provides anonymity for both the trust host peer and the trust querying peer. Other systems worth to be mentioned are for example, EigenTrust [20], PeerTrust [45] and NICE [36]. In the first two approaches the peers are given trust values according to different algorithms and considering different aspects. In NICE the peers come to the system with a pair of private and public keys, thus transactions in NICE are made by secure exchange of certificates.

3 Sensor Networks

3.1 Introduction

The main purpose of a WSN is to serve as an interface to the real world, providing physical information such as temperature, light, radiation, and others, to a computer system. A sensor network can be abstracted as a "living being", where honest and fully cooperative sensor nodes ("sensing cells") are managed by an entity named base station ("brain"). There is a high number of sensor nodes, usually densely deployed, that can perceive the physical events as they occur.

All these nodes process and forward their signals through a wireless channel to the base station that, based on that information, provides a number of services to an external system. An overview of the structure of a sensor network can be seen in Fig. 1.

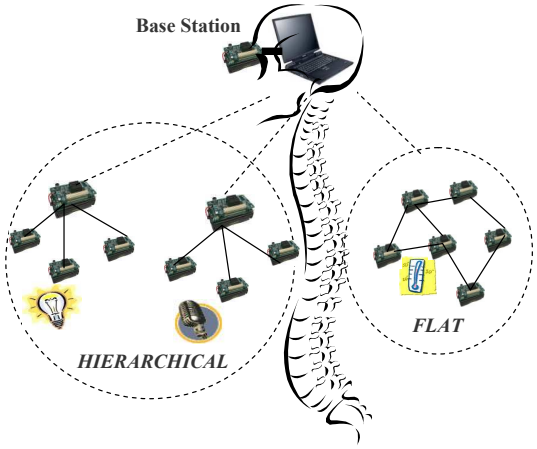

Fig. 1. Overview of the Architecture of WSN

The services that a WSN can offer are classified into three major categories: monitoring, alerting, and provisioning of information "on-demand". As for the first case, sensor nodes can continuously monitor certain features of their surroundings (e.g. measuring the ambient noise level) and timely send such information to the base station. In the second case, sensors can check whether certain physical circumstances (e.g. a fire) are occurring, alerting the users of the system when an alarm is triggered. In the last case, the network can be queried about the actual levels of a certain feature, providing information "on-demand". Due to the computational capabilities of the nodes, it is possible to re-program the network during its lifetime, or even use it as a distributed computing platform under specific circumstances.

As of 2007, the number of industrial applications based on sensor networks have been very few. However, there have been a large number of prototypes, both coming from the industry and from the academia. Those prototypes show the important role that the sensor network technology can play in the future. A traditional scenario where sensor networks technology has been applied is agriculture (e.g. maintenance of vineyards [10]) and environmental monitoring (e.g. analysis of seismoacustic data related to volcanoes [44]). Sensor networks have been also successfully employed in health-related prototypes as, for example, wireless vital sign sensors [1]. Other scenarios include office management [29], critical infrastructure protection [2], fire-fighting [3], etc.

It is important to point out that the architecture and network model used in a particular sensor network deployment is highly dependant on the requirements of the scenario. For example, a simple vineyard monitoring application will consist of static nodes that will periodically send their information to the central base station. On the other hand, a health monitoring application may require of a mobile base station (e.g. worn by a nurse) that receives the data from its patients. The scenario also has a great influence over all the aspects of a sensor network application: nodes can only afford to have specific components related to its functionality (e.g. its protocols, its security mechanisms, etc) due to their high hardware constraints. Note that most of the existing prototypes are based on static networks, that is, networks with nodes that do not move from their initial deployment point. As a result, most protocols and services are prepared to manage only static networks.

3.2 Elements and Network Models

As aforementioned, the elements of a sensor network are the sensor nodes and the base station. The main tasks of a sensor node are the following:

- to get the physical information of the surroundings using its built-in sensors,
- to process the raw information by benefiting from its limited computational capabilities, and
- to communicate with other nodes in the area around using a wireless channel.

All sensor nodes are battery-powered; hence, totally independent and able to operate autonomously, if required. Nevertheless, they can also collaborate with other nodes in pursuing a common goal, such as vehicle tracking. On the other hand, the base station is the element for accessing to the services provided by the sensor network. All data coming from the sensor nodes, as well as all control commands that can be issued to those nodes, will traverse the base station.

Although sensor nodes are considered to have limited computational capabilities, there are in fact different types of nodes with different levels of constraints: "Weak" nodes, "Normal" nodes, and "Heavy-Duty" nodes.

"Weak" nodes are extremely constrained, with resources as low as 4Mhz, 64B of RAM memory and 1.4kB of instruction memory. These nodes usually perform just sensing operations, without participating in other protocols of the network such as routing. "Heavy-Duty" nodes have PDA-like capabilities (>100Mhz, >256kB RAM, >4MB instruction memory), and can be used as constrained base stations, or as cluster heads. Finally, "Normal" nodes are the most common type of sensor node device with enough resources to create a fully functional sensor network (8-16 Mhz, >4kB RAM, >48kB instruction memory), but constrained enough to be careful on the development of the application.

The network model of a WSN is mainly determined by the way these sensor nodes organize themselves (in groups or purely distributed) and behave in their

deployment field (remaining static or being mobile). Regarding organization, there are two basic sensor architectures, hierarchical and flat, that specify how the sensors group themselves in order to achieve specific goals.

In flat configurations, all the nodes contribute in the decision-making process and participate in the internal protocols (like routing). Conversely, in hierarchical configurations the network is divided into clusters or group of nodes. Organizational decisions, like data aggregation, are made by a single entity called "cluster head". It should be noticed that it is also possible to have a combination of the two previous configurations into the same network; for instance, to avoid situations where the "spinal cord" of the network - the cluster heads - fail and the information must be routed to the base station.

The nodes of a sensor network can also be either static or mobile. In static networks, which are the most common configuration of WSN, nodes do not move from their deployment place in all their lifetime. If these networks are not deployed optimally, they may face problems related to network availability (e.g. when a node is isolated due to environmental factors or internal problems), limited sensing (i.e. when an important event occurs on a scarcely populated area of the network due to bad deployment planning), etc. In mobile networks, nodes with limited mobility coexist with static nodes, in order to solve problems such as network availability. These nodes are able to move to certain points of the deployment field when the application requires it. This is rather problematic from the point of view of implementing correct and optimal protocols, and there are many open research issues in this field.

The network model is not only determined by the organization of the nodes, but by the organization of the base station as well. In most cases, a base station is static, and does not change its position during the lifetime of the network. In other scenarios, such as oceanographic scenarios, a base station can be mobile, positioning itself in the deployment field based on the information supplied by the sensor nodes. Either static or mobile, most scenarios assume one single base station. Still, in the cases where the information obtained by the WSN has to be accessible from more than one point, the coexistence of several base stations is feasible. Even more, it can be possible to have "delegated" base stations, PDA-like devices used by a human operator, that access the information of the WSN on the spot. Finally, it is also possible to have "zero" base stations. That is, a sensor network where the base station is not present at all times, and that only appears when certain data has to be collected [26].

3.3 Security Problems in WSN

Since sensor networks is a young technology there are many interesting research problems, like development of models and tools for the design of better WSN architectures, elaboration of standard protocols adapted to work robustly on certain scenarios, etc. However, one of the most important issues that remains mostly open is *security*. Sensor nodes are highly constrained in terms of computational capabilities, memory, communication bandwidth and battery power.

Additionally, it is easy to physically access the nodes because they must be located near the physical source of the events, and they usually are not tamper-resistant due to cost constraints. Furthermore, any device can access the information exchange because the communication channel is public.

As a result, any malicious adversary can manipulate the sensor nodes, the environment, or the communication channel for its own benefit. For these reasons, it is necessary to provide the sensor network with basic security mechanisms and protocols that can guarantee a minimal protection to the services and the information flow. This means to provide protection on the hardware layer, the communication stack, and the "core protocols". In other words, (i) it is necessary to protect the hardware of the nodes against attacks, (ii) the communication channels must meet security goals (like confidentiality, integrity and authentication), and (iii) the core protocols of the network must be robust against any possible interferences.

It has been aforementioned that in most cases a sensor node neither has tamper protection nor is enclosed on a tamper-resistant package. A malicious adversary sufficiently skilled could take a node and subvert it, obtaining information such as secret credentials that would allow him to create a node that maliciously interact with the network. However, it is possible to use *data and code obfuscation* schemes that generate different versions of the sensor software for each node (cf. Alarifi and Du ([6])), obliging thus the adversary to employ a non-trivial effort to understand the behaviour of the node and to track down the information he needs.

Although a node cannot protect itself, it is possible for others to check its state. Using a procedure called *code attestation* [30], it is possible to dynamically check whether a node is running the program that it should contain. It is even possible to force the node to reprogram itself, after the attestation process, with a "good" copy of code [40]. Nevertheless, it is not possible at this moment to have both code obfuscation and code attestation on the same node. Achieving this remains as an open research problem.

Regarding the communication flow, the nodes need to make use of the basic security primitives in order to authenticate the peers involved in the information exchange while protecting the confidentiality and integrity of the channel. Those primitives are *symmetric key encryption* (SKE) schemes, *message authentication codes* (MAC), and *public key cryptography* (PKC) schemes. the implementation of those security primitives in the existing hardware for sensor nodes has been very challenging, but the state of the art in these areas is quite advanced. Still, the extended use of PKC is a special case and it presents many challenges that need to be solved.

There are software-based SKE schemes, like TinySec [22], that provide block ciphers such as Skipjack or RC5 in CBC mode with a minor overhead - less than 10%. Moreover, in nodes with radio chips conforming to the 802.15.4 standard, SKE is actually provided by the hardware in form of the AES stream cipher. MAC are usually computed using a cipher block chaining construction, called CBC-MAC, that takes advantage of the existing SKE primitives. So far,

it has been possible to implement PKC on sensor nodes by using elliptic curve cryptography (ECC) instead of other more traditional algorithms, such as RSA, that become more "expensive" in these scenarios. As shown in [42], a node can perform a public key signature in 1.92s, and verify it in 2.41s.

A problem associated with the existence of the security primitives is the need of having a *key management system* (KMS). The security solutions need certain security credentials, i.e. pairwise secret keys, in order to work. The KMS is in charge of creating and providing these keys, hence constructing a secure key infrastructure. There have been multiple KMS suggested by the research community that allow two neighbouring nodes to share a secret key, but it is difficult to figure out which is the KMS that better suits to a certain context or application. As suggested in [7], this can be achieved by analyzing if the properties offered by a particular KMS match the requirements of the scenario where the nodes are going to be deployed. In their work, they consider properties such as memory and communication overhead, processing speed, network resilience, confidentiality, connectivity, scalability, and energy usage.

It is easy to understand that protecting the communication channel between two nodes does not entirely guarantee the security of a sensor network. The "core protocols" of the network, that is, the minimal set of protocols required to provide services also must be secure in order to withstand errors coming from faulty nodes, as well as attacks initiated by malicious elements (from outside and inside the network). We are meaning, for instance, the protocols associated to *routing, data aggregation, and time synchronization*. For those protocols, the services provided are, respectively, transmitting a packet from one node to another node, briefing many sensor readings into one single piece of data, and synchronizing the clocks of the network.

There are multiple attacks that can be performed against these core protocols, as shown by Karlof and Wagner [21], Sang et. al. [34], and Manzo et. al. [27]. The field of time synchronization is fairly advanced and many protocols have been proposed to provide that service in a secure way. Unfortunately, it is not the same case for routing and aggregation. The reason is that though there are multiple protocols that provide those services, very few have been specifically designed to deal with errors or malicious insiders. On the other hand, this area of research is advancing at a steady pace.

Consequently, protecting a sensor network is not a trivial task. Actually, it goes even beyond the protection of the services and the information flow. There are more issues that need to be addressed, such as robust and secure location methods for the nodes, secure management of mobile nodes and base stations, delegation of tasks, data privacy, authentication of broadcasted messages coming from the base station, support for automatic and secure code updates, and many others. In any case, it is clear that since a WSN has to be self-sufficient and self-configurable, it is essential to create an infrastructure that is aware of the current situation of the network and that could help the nodes to manage themselves. Such task can be fulfilled by using a Trust Management System.

4 Trust Management for Wireless Sensor Networks

4.1 The Importance of Trust in Wireless Sensor Networks

Trust is a very important factor in the decision-making processes of any network where uncertainty is a factor. That is, when the outcome of a certain situation cannot be clearly established or assured. With no uncertainty, there is no need for a trust management system: if an element of the network knows in advance the actual behaviour of their partners (e.g. collaborative, malicious, faulty,...), it can make a flawless decision. As a result, in order to know whether a trust management system can be applied to a WSN, it is necessary first to analyze the importance of uncertainty in such environment.

Uncertainty originates basically from two sources [38]: information asymmetry (a partner does not have all the information it needs about others), and opportunism (transacting partners have different goals). On the context of sensor networks, opportunism is not a problem. All the elements of the network work towards the same goal, and they have neither reason nor the will to behave egoistically. On the other hand, a sensor node does not have information regarding others that will allow it to know in advance how a transacting partner is going to behave. Therefore, there is some information asymmetry that the node must deal with.

Since all nodes belong to the same "living being", it is possible to think that the existence of information asymmetry is not a real problem. When a sensor node chooses a partner to collaborate with, such partner is supposed to be honest and fully collaborative. However, this is not entirely true. As well as living beings are affected by illnesses, sensor networks can suffer the attack of malicious nodes or the existence of faulty nodes. As a result, uncertainty in sensor networks is a problem that must be dealt with.

Once the importance of trust have been clarified, it is necessary to describe where it could be of use in a WSN context. Its primary purpose is to allow self-sufficiency: a Wireless Sensor Network must be able to configure itself during its lifetime in presence of extraordinary events. By knowing the reputation of their neighbourhood and their actual behaviour, it is possible for the nodes to calculate a trust value and choose a suitable course of action when taking operational decisions (knowing who is the best partner for starting a collaboration) or in extreme situations (e.g. nodes malfunctioning).

Self-configuration is not the only benefit of trust: a trust management system can also assist and/or take advantage of other security protocols. Regarding hardware protection, existing code obfuscation and code attestation schemes can be easily integrated into a trust management system as tools for testing the integrity of an untrusted node. The existence of a trust management system can also assist the activities of Key Management Systems (KMS) by, for example, revoking the keys of an untrusted entity. Finally, complex services such as secure location and intrusion detection systems can benefit from the existence of a trust management system, either by using the output of the system as an assistant in

their decision-making process, or by providing useful trust inputs that could be of use for any other service.

4.2 Trust Management Architectures for WSN

We have discussed in Section 4.1 the importance of trust for WSN and why we need trust management systems for these type of networks. The architectures considered in the literature for solving the challenge of managing trust relationships differ depending on the underlying problem. For sensor networks, it is necessary to have a lightweight distributed architecture that tries to assure coverage of the whole network. This architecture must be "behaviour-based" in order to react to the events that may occur during the lifetime of the network. These requisites are given by the decentralized nature of WSN and its specific characteristics and constraints.

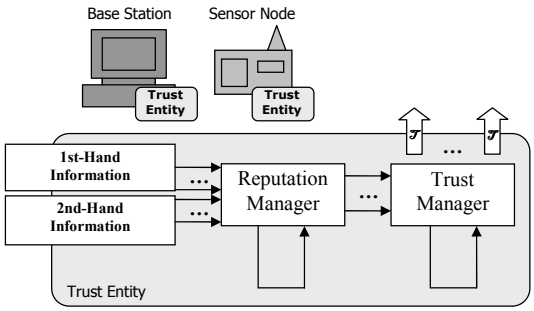

Fig. 2. Structure of a Trust Entity for Sensor Networks

An important element of any trust management system is the trust entity. This entity is in charge of obtaining, calculating and maintaining reputation and trust values. For sensor networks, it is possible to define the structure of a generic trust entity, as shown in Figure 2. In this structure, the "information" modules obtain information about the behaviour of the members of its neighbourhood, either through observation and experience (i.e. "first-hand information") or by sharing the observed events with other entities (i.e. "second-hand information"). After this process, the "reputation manager" module can use this list of events to infer and store the reputation of the members of its neighbourhood. Such reputation will be later used by the "trust manager" module to obtain the trust values. They can be used to decide which is the best partner for a certain operation, or discover if one entity is behaving maliciously. Both modules need to maintain and update their values during the lifetime of the network.

This structure is clearly applicable to wireless sensor networks, because a sensor node can obtain information about its surroundings either directly or indirectly. In addition, the sensors have limited computational capabilities. Consequently, by using lightweight algorithms, they can be able to infer the reputation

of its neighbours and decide if they trust them for certain operations. Moreover, this model of a trust entity fits in the design of most of the existing work on trust for sensor networks, although only a few of those works take reputation explicitly into account [13, 16, 49]. Still, having both reputation and trust in the same system is essential. By not calculating the trust directly from the behaviour of a node, it is possible to better handle aspects such as the evolution of the node, aging, etc.

It is clear that the trust entities have to be located in the sensor nodes. There is a question, though: how to distribute them inside the network? A very common solution for both flat and hierarchical configurations is the use of clusters [35, 49]. This approach makes easier the process of calculating and storing data, as we should remind WSN are always subject to energy-constraint problems and computational power. The cluster head maintains the communication with the base station and usually stores the trust or reputation values, minimising in this way the energy consuming of the simple nodes. An additional problem in these cases is how to elect a cluster head that does not misbehave or it is not compromised.

For purely distributed networks, it is necessary to have a trust entity inside every node of the network. The reason is simple: in a flat configuration, all sensor nodes participate on the protocols that support the network, such as routing. The decisions regarding the execution of the protocols (e.g. who could be the next node in the routing path when transporting an "Out-of-Band" message) are usually made by the nodes on their own, and in exceptional situations with the help of its direct neighbourhood (e.g. when aggregating some data). Finally, faulty and/or malicious nodes may appear on any part of the network. Therefore, nodes need to know whether they can trust their neighbourhood in order to deal with uncertainty.

Note that, as pointed out by Tanachaiwiwat [41], the sensor nodes are not the only devices that are part of the trust management architecture. The base station can have its own trust entity as well. Due to its role as a network manager and data repository, the base station receives information from all the nodes in the network. As a result, its information asymmetry is reduced: it has a global point of view of the state of the network, whereas sensor nodes can only manage to observe their immediate surroundings. The base station can take advantage of this wealth of information to observe and analyze the behaviour of its nodes, storing their reputation and making global trust decisions. Although it cannot directly influence the behaviour of the nodes, it can issue orders that those nodes must fulfill.

Once we have described how the nodes and the base station are organised in order to design a better trust management system, now we will concentrate on how these systems and its components should. Previously we have highlighted what are the components that a successful trust management system should have. In Sections 5.1 and 5.2 we will see how these components and the main elements of the system should work.

4.3 Current Trust Management Solutions for Wireless Sensor Networks

Research on trust and reputation systems for Wireless Sensor Networks is at a very early stage. Some works have been done in recent years, however most of them are designed in order to solve a very specific problem and do not deal with all the features that a good trust management system for sensor networks should possess. Other works design a trust management model for WSN but without pursuing a particular purpose. Note that as it was mentioned in Section 3 the majority of the network prototypes and scenarios mentioned in this work are developed for static WSN. In the following, we will discuss some of the most relevant existing approaches of trust management systems for WSN.

In [16] the authors propose a reputation-based framework for sensor networks where nodes maintain reputation for other nodes and use it to evaluate their trustworthiness. Reputation is represented through a Bayesian formulation, more specifically, a beta reputation system.

The architecture of the system consists of a watchdog mechanism, reputation, second hand information, trust and behaviour. The mathematical tool used for representing and updating the reputation values is the beta distribution of Jøsang. A watchdog mechanism is also used in [13]. This work uses reputation in order to define the trust management system for WSN. They consider the concept of *certainty* for trust. The first-hand information is gathered by using a watchdog mechanism. Then a reputation space, RS is defined considering the positive and the negative outcomes. A trust space is defined from the reputation space.

In [35] The authors introduce a lightweight group- based trust management system scheme (GTMS) for distributed wireless sensor networks. GTMS uses a hybrid trust management approach instead of using completely centralized or distributed schemes. The group-based trust model works is three phases:

1. Trust calculation at node. At this level trust is calculated based on past interactions or recommendations.
2. Trust calculation at cluster head. The cluster head requests trust values from all the nodes in a group. The members of that group send the requested trust values of other members to the cluster head. The trust vector of the cluster head, $\overrightarrow{Tv_{ch}}$, is defined as

$$\overrightarrow{Tv_{ch}} = (Tv_{ch,1}, \ldots, Tv_{ch,n})$$

   where $Tv_{ch,i}$ is the of node i, which is calculated from the previous trust values.
3. Trust calculation at base station (BS). On request of the BS the cluster heads forward their trust vectors and recommendations based upon past interactions to the BS. The BS then maintains the trust matrix. Based on that matrix the BS calculates the trust value of each group and then classifies the groups in *trusted*, *untrusted* or *uncertain* depending whether the trust values fall into different thresholds.

The problem isolating misbehaving nodes is also addressed in [32]. In this work the authors consider as a basis for their the work the case of industrial sensor networks which are static. They associate a Suspicion Level (SL) with each sensor. SL represents the belief that the sensor is not acting reliably according to the expectations of sensor behaviour formed before the actual interactions with the sensor. SL takes values on the interval $(0, 1]$.

Other systems deal with specific problems such as detecting misbehaving nodes in "core protocols" (such as routing) or electing non-compromised cluster heads. [41] presents a location-centric architecture for isolating misbehaviour nodes and establishing trust in sensor networks. The underlying problem the authors target is a misbehaviour model in which a compromised or faulty node consistently drop data packets while participating in signaling and routing protocols, always over static networks. The trust routing protocol is called TRANS. The main modules of this location-centric architecture are trust routing, installed in the base station and all the nodes; and insecure location discovery and isolation, installed only in the base station. TRANS selects a secure path that avoids insecure locations by using the concept of trust. Trust values are assigned depending on the replies that the base station receives from other nodes. Also, each sensor node calculates trust values for its neighbours' location. The trust values are obtained based on trust parameters and encouraging factor, β. If the trust value drops below a certain threshold constantly this could indicate a potential misbehaving insecure location. The base station would then isolate the node by using different schemes (refer to [41] for more details). TIBFIT [23] is a protocol that aims to detect and mask arbitrary node failures in an event-driven wireless sensor networks. The nodes are organised into clusters. They can fail in an arbitrary manner generating missed event reports, false reports or wrong location reports. A trust index is assigned to each node indicating its track record in reporting past events correctly. The cluster head s in charge of analysing the trust index and making event decisions. This trust index (TI) is a real number ranging from zero to 1. Initially it is set to 1. For each report a node makes that the cluster head estimates is incorrect the TI assigned to such a node decreases. The protocol is also able to determine locations of the event reports.

The aggregation problem is considered in [49] The authors consider the problem of aggregation for WSN. The sensor network is organized into clusters where the cluster head acts as a gateway between the cluster and the BS. Some nodes on the network act as "aggregators", responsible for aggregating data and reporting the information to the cluster head. They used Jøsang's belief model in order to deal with uncertainty in data streams. The aggregator collects information from other nodes. This information is given in forms of reputation. The aggregator node classifies the nodes into different groups based on their reputations. After calculating reputations (by using the formulation in the paper) for nodes the aggregator determines whether there are compromised nodes. The best way to do that is by predefining a threshold.

The problem of electing cluster heads is considered in [15]. If this cluster head is malicious or compromised this could mean a breach in the WSN. Thus,

the authors introduce a trust- based framework that reduces the likelihood of a malicious or compromised node from being elected as a cluster head. The trust parameters used are measurable and observable networks events. These events are, for example, packet forwarding, data packet modified or packet address modified. Thus, the trust level, $T_N(X_i)$, that node N has computed about node X_i, is calculated as a weighted summation of the parameters mentioned before. Each node stores a trust table where it records the trust values about the other nodes. In [51] This work proposes a security framework with trust management on a distributed trust model, which enables the nodes to evaluate their node's behaviour and make decisions. The trust values are obtained taking into consideration different parameters: *personal reference* and *reference*. The personal reference ($T_{pr(i)}$) that a node which is computing trust (*judge*) has on another node (*suspect*) is obtained by considering aspects such as the forwarding of packets, availability or confidentiality among others. Reference ($T_{r(i)}$) is the kind of recommendation provided by the *juries* (nodes that maintains the trust value of the suspect with the judge and sends out the corresponding opinion periodically). Reference is obtained via a recommendation protocol to specify how the judge and the jury communicate to exchange the information about the trust values. At last, the final trust value is obtained as a weighted summation of the personal reference and the reference.

5 Analysis and Features of a Trust Management System for WSN

5.1 Information Gathering

5.1.1 Foundations of Information Gathering For the development of a behavioral-based trust management system, it is necessary to collect information regarding to the behaviour of the nodes of the network. In a wireless sensor network, this information gathering process can be done in several ways. As we have seen in the description of the current solutions presented in Section 4.3 the way information is gathered could be either distributed or centralized. If the system is centralized it is the base station(BS) which is in charge of maintaining and distributing the information. In distributed systems each node keeps its own measurements on first and second hand information. Only Shaikh et. al [35] consider a hybrid approach in order to gather information: distributed within the cluster and centralized in between each group or cluster and the base station. A watchdog mechanism is used in [16] and [13].

Another important aspect in the process of information gathering is the initialization of the gathering process and the kind of information that is gathered. Initial information does not seem to be a crucial problem for sensor networks. Before deployment, sensor nodes are programmed in a controlled environment by the network manager, with similar tasks and services. Thus, at the beginning or their life they can be completely trusted: their hardware is supposed to be tested for failures before deployment, and also at this stage any malicious

adversary had neither the time nor the chance to influence or subvert a node. Initial reputation should not affect negatively both trust and the decisions made by the nodes. Note that some systems tend to link initial reputation and trust values with authenticating the nodes. However, in a realistic sensor network setting, any node with no credentials should be expelled from the network, since the communication channel needs to be protected with cryptographic primitives due to its public nature.

Once the network starts functioning, the kind of information that is gathered will determine how to calculate the reputation and trust values of a node. This information could be first or second-hand information. The sources for obtaining either of them could be, for instance, forwarding or dropping packets [51, 41, 16] or event reports [23]. The behaviour of a node in a WSN is the key aspect for the decision making process. Thus, sources of mistrust could be if a node is detected inactive for a long period of time or appears and disappears from the network constantly should be considered un-trusted. Most of the existing works on trust management for WSN, however, concentrate on the communication layer of the network. The most important events to report for these methods are those related to the forwarding or dropping of packets. An analysis of the most general sources of information gathering are discussed in section 5.1.2.

The kind of information collected depends also on the type of application that the trust managements system aims to solve. It is a very well-known property of trust that it is not universal. This means that an entity A may trust another entity B to perform an action but the level of trust changes if the action is different (I must trust a mechanics to fix my car but not to fix my teeth). Some systems, such as [13], gather information and classify it into positive or negative events without specifying which kind of events should be considered. Thus, this system is supposed to work for whatever the events are, as it is not designed in order to solve a determined problem. On the contrary, other approaches try to solve a specific problem, and the information gathered is only meaningful for such problems. Achieving a balance on this subject is therefore desirable.

Concerning the trust management systems that we mentioned in Sections 2.1 and 2.2 for P2P and Ad-Hoc networks, the process of data collecting is different. This is mainly due to the fact that the behaviour of nodes in a WSN is different from the other type of networks, and behaviour is precisely, the main source of information for deriving trust or reputation, and also the way to collect information. Thus, for example, some approaches give initial values [46, 25] which is not necessary for sensor networks as all the nodes are supposed to function properly initially.

5.1.2 Sources of Information In a wireless sensor network there are a large set of events that can be used as inputs for a trust management system. These events may provide the necessary information to model the behaviour of a certain sensor node. Examples of events are the number of messages relayed from one node to another, or the contents of an aggregation report. Some events are related to the specific protocols that implement the particular application provided by

the network, but others are more generic and exist regardless of the underlying protocols and services.

Examples of application-specific events that are relevant for discerning the real behaviour of a node can be found in the actual state of the art on trust management systems. For example, the detection of node misbehaviour in aggregation processes or in the election of cluster heads are analyzed in [15, 41, 23, 49] (cf. section 4.3). In all these cases the purpose of the system determines the data and the information that should be gathered in order to derive reputation and trust values. Still, it is important to name and consider the generic events that may indicate the behavior of a node in every kind of application. This is extremely useful for creating the foundations of a trust system, which can be further expanded with application-specific mechanisms.

There are many generic events that can be important to the system, such as hardware-related errors and deviations from the sensor readings. However, as aforementioned, the major source of generic information are the events that occur on the communication layer. From the number and format of the messages sent inside the network, it can be possible to infer several extraordinary situations: the existence of repeated or malformed packets, the creation of packets, and the selective delaying or dropping of packets.

The existence of *repeated packets* should indicate the possibility of a problem in the wireless channel. Even though, if the network load is not high and the source of the packets is near enough the destination, then the source node should be mistrusted. Also, if the repeated packets are *malformed* (i.e. with an invalid integrity code), it is a clear sign of the existence of an (possibly external) malicious node. In any case, as the packets can be produced by an external entity, the system should be careful with mistrusting a node of the network if there is no proper authentication mechanism.

Another possible source of mistrust is the *creation of packets* out of a specific time period. Besides alarms and queries, a sensor network usually produces messages during a specific period of time or "burst time", when the data packets containing sensor readings are forwarded to the base station. If a node creates a packet outside this period, it may be an indication of problems inside that node. Other causes of concern are the existence of alarms (e.g. temperature sensors reporting a fire) where the physical surroundings are calm, and the existence of nodes reporting an answer to a non-existent query of the base station.

A major cause of suspicion is the *selective forwarding* of packets. If certain packets get dropped by a specific node, and the overall load of the section of the network where that node belongs is not high, it is mostly sure that the node is behaving maliciously. A node can also be considered non-trusted if the time consumed on forwarding an incoming message to its destination is higher than the average of the network, *delaying* the routing process. All this information can be obtained thanks to the broadcast nature of communications, although the asymmetry of those communications have to be taken into account.

From the perspective of the *hardware*, a node that is not detected as alive for a long period of time [9] should be considered suspicious of being tampered by

an adversary. A node that appears and disappears ("blinks") from the network under normal conditions should not be considered trusted either. For this particular case, the reason of mistrust should be the belief that the node is starting to malfunction and cannot properly provide services to the other nodes.

Regarding to *sensor readings*, the inherent redundancy of the network can help on detecting problems, although the exact nature of the application will influence over what can be considered a major deviation from normal readings. For example, in a wildfire monitoring scenario, it is expected to monitor high temperatures, thus it is more important to consider abnormal fluctuations and inconsistent readings. On the other hand, in scenarios such as office monitoring, extreme readings may inform of a malfunction in the sensor.

A node should not only take into consideration the reports produced by itself while observing other nodes, but also the *reports* produced by its neighbouring nodes ("second-hand" information). While using these reports it is necessary that the node assures the authenticity of its sources and the integrity of their contents in order to avoid the participation of external entities. Unfortunately, it is also possible to receive malicious reports from tampered nodes. For this reason, there should be a mechanism for assuring the correct management of the information. The inherent redundancy of sensor networks can help to develop this kind of mechanism, since the existence of a malicious report (e.g. a bad-mouthing attack) that is not coherent with the state of the neighbourhood can be an indicative of a malicious presence.

It is important to note that a source of second-hand information can be a sensor node accusing itself of being malicious. Following the simile of the "living being", this entire process is similar to the concept of *apoptosis*, when a cell suicides due to malfunctioning, virus infection, or other reasons [24]. Due to the embedded intelligence of a sensor node, it can detect whether its batteries are low, its readings are inconsistent with its neighbourhood, or its transceiver seems to not work. On discovering these issues, the sensor node can try to alert its neighbourhood about its state. It is not possible for a malicious adversary to take advantage of this kind of "second-hand" information, since a subverted node can only accuse itself of being malicious, thus alerting the network and the base station about its existence.

5.2 The Model of Computation

5.2.1 Information Modelling on Sensor Networks Once the information, either "first-hand" or "second-hand", has been gathered the trust entity is able to output trust measurements based on existing reputation values. The task of calculating and storing the reputation of a node relies on the module known as reputation manager. Due to its memory constraints, a sensor node cannot store all the events that its neighbours produce during its lifetime. Therefore, it is necessary to create a lightweight reputation manager that could capture and efficiently store the behaviour of other entities in the previous interactions, while being able to update it with new information if possible.

The other part of the trust entity, the trust manager, is in charge of calculating a certain trust measurement of a node using as an input its existing reputation, and providing the trustor with a measurement that can help it to make a decision over a certain trustee. This trust measurement should be obtained by taking into account the risk of the interaction between the trustor and the trustee, and according to the importance of the reputation value, and that specific interaction. Risk and importance are significant factors in the calculation of trust, but they also influence the selection of a threshold. That is, when a certain trust value labels a trustee as "trusted" or "untrusted" for a certain operation. There are other, non-exclusive ways to use the trust values, such as when one trustor have to choose over a group of trustees.

While calculating the reputation of a node and its trust measurements, it is essential to take into account the granularity of the trust management system. As aforementioned, the reputation of a certain node is built according to its behaviour and the events it triggers. Most systems simplify the reputation into one single set of values. However, the actions of the nodes are not reduced to the execution of one task. For example, a node can route information to the base station, and read the physical measurements of its environment using the sensors, amongst others. A node needs to maintain separate opinions about the existing actions of their peers, thus it needs a different set of reputation values. A consequence of this fact is the need of linking the existing events with the different reputation values they influence.

The existence of different reputation values also implies the existence of different trust values. A specific trust value (e.g. routing) will help the node to decide about the possible outcome of a specific interaction with another peer. On the other hand, that value cannot be used in most cases to deduce what the peer could do in a different task (e.g. sensing). For example, a node that loses data while forwarding packets cannot be trusted as a message forwarder, but it may be trusted as a possible source of data.

The dimension of a sensor network as a balanced "living being", that should have none or little deviation from its behavioral patterns, affects over the functionality of both trust and reputation manager. For example, a node that acts truly maliciously in the context of a sensor network will most surely keep such evil behaviour in further interactions. Therefore, "bad" reputation should not be forgotten easily while updating the reputation values. The evolution of the reputation on the aging process is also an important factor that a node cannot ignore: a trust entity should remember if a node achieved high "bad" reputation ratings on the past.

Also, the occurrence of certain events have a more direct impact on the reputation of a node. These events, like selective forwarding, are a clear indicative of malicious or erroneous activities. As a consequence, a node exhibiting such behaviour should be flagged with a very low reputation value. In addition, the consistence of the trust readings is also significant. A normal sensor network environment should produce little or none reports regarding malicious activities.

Therefore, the existence of different and contradictory reports should be evidence enough of malicious activity and source of mistrust.

Finally, all the important decisions made by the nodes, such as node exclusion, should be notified to the base station. It does not mean that the trust management system has to be centralized, but that the base station, as the user of the network, should know about the internal status of the network for logging, monitoring and maintenance purposes. Also, the existence of a strange situation can be the symptom of a greater problem, since a sensor network behaves satisfactorily by default. As a result, the trust entity that exists inside the base station can use this newly acquired information for the benefit of the whole network.

5.2.2 Existing Computation Models The derivation of the reputation and trust values is done by using a certain computational model. Usually, these methods are based on mathematical models mainly statistics or probability theory. This is very similar to approaches for Ad-Hoc and P2P networks. As these trust management systems are mainly behaviour-based in order to compute the trust or reputation values mathematical tools are used.
Some times the trust values are calculated via simple or weighted summations of the different data collected [35, 51]. Simple mathematical functions are also used in [41]. In this approach the trust value is calculated by a product of different parameters:

- Cryptography, (C),
- Availability, A_i, and
- Packet forwarding

The trust value is calculated as a product of these three parameters and an *encouraging factor*, β, that helps to encourage the packet forwarding in the initial phase of the packet forwarding. Thus, $T_i = C_i A_i \beta P_i$.
Linear functions are also used in [35]. We already mentioned (see Section 4.3) that in this approach the calculation of trust values is done in three different phases: at the node, at the cluster head and at the base station. In each of these phases the calculation is done, for instance, by functions such as

$$T\nu_{x,y} = \frac{(PI_{x,y}) + PR_{x,y}}{2}$$

which is the the trust value node x wants to calculate on node y. $PI_{x,y}$ is the past interaction trust value and $PR_{x,y}$ is the peer recommendation trust value of node y calculated by node x. These values are calculated using also similar linear functions.

The two approaches described above use simple mathematical functions such as summation or product. The exponential function can be also used, and this is

the case in [23]. As we describe in Section 4.3, TIBFIT tries to combat failures in the reporting event, thus each node is assigned a TI, maintained at the cluster head. The TI is calculated as

$$TI = e^{-\lambda\nu}$$

where λ is a proportionality constant that is application dependent and ν is a variable for each node maintained by the cluster head. This variable is incremented every time the node makes a faulty report. An exponential function is also used in [32].
The beta distribution of Jøsang [17] is used in some systems developed for Ad-Hoc and P2P networks as well as for some systems for WSN [16, 49]. The advantage of using this model is that is supported by a robust mathematical tool. This method is based on the definition of an *opinion* that express the degree of belief in the truth of a statement. This model is very suitable for the problem of aggregation in sensor networks [49], as the aggregation of data problem is infiltrated with uncertainties due to the unavoidable sampling errors, false data injected by either compromised nodes or aggregators.

Probability theory is used in [13]. As we mentioned in Section 4.3 this approach uses a watchdog mechanism in order to gather firs-hand information. This watchdog mechanism also records the outcomes of several events and classifies them into positive or negative events, $< p, n >$. According to the Bayes theorem the probability of a positive outcome, x, is defined as

$$P_{<p,n>}(x) = P(x| < p, n >) = \frac{P(< p, n >, x|x)P(x)}{\sum P(< p, n >, x|x)P(x)} = \frac{(p+n+1)!}{p!n!}x^p(1-x)^n$$

This conditional probability is the posterior probability of reputation $< p, n >$.

6 Conclusions

In this chapter of the book we have tried to give an overview of the state of the art of trust management systems for Wireless Sensor Networks as well as an analysis of the main features that these trust management systems possess.

We also outline the importance of trust for these kind of networks, as for any networks where uncertainty is a fact. Thus, trust management systems will become an assistant for solving the decision-making problem.

The design of trust management systems for WSN is constrained by the nature and features of these type of networks (computational power, energy constraint) and also depending on the underlying problem that the trust management aims to solve. Thus, a system designed for detecting misbehaving nodes could be different than another one designed, for instance, for routing.

Special attention should be paid to the way of gathering information and what sort of information is relevant to be gathered. Thus, causes of mistrust could be dropping packets or, appearing or disappearing from the network without an apparent reason.

Once the information is gathered the underlying mathematical model used for computing the trust or reputation values of the nodes is also different from one model to the other. Even if in some cases simple averages or linear functions like a product are used, the values obtained in these cases might no be very significant. Theory of probabilities and some theories developed for these purposes such as the belief theory of Jøsang provide a well founded mathematical tool for trust management systems in general.

References

1. The CodeBlue Project. http://www.eecs.harvard.edu/ mdw/proj/codeblue. Harvard University (2006).
2. WINES II - Smart Infrastructure. http://www.winesinfrastructure.org, University of Cambridge and Imperial College London (2006).
3. FIRE Project. http://fire.me.berkeley.edu/, University of California, Berkeley (2006).
4. A. Abdul-Rahman and S. Hailes. Supporting Trust in Virtual Communities. In *Proceedings of the 33rd Hawaii International Conference on System Sciences*, 2000.
5. K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In Henrique Paques, Ling Liu, and David Grossman, editors, *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM01)*, pages 310–317. ACM Press, 2001.
6. A. Alarifi and W. Du. Diversifying sensor nodes to improve resilience against node compromise. In *Proceedings of The 4th ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2006)*, Alexandria, USA, October 2006.
7. Cristina Alcaraz and Rodrigo Roman. Applying key infrastructures for sensor networks in cip/ciip scenarios. In *1st International Workshop on Critical Information Infrastructures Security (CRITIS 2006)*, 2006.
8. T. Bearly and V. Kumar. Expanding Trust Beyond Reputation in Peer to Peer Systems. In *15th International workshop on Database and Expert Systems Applications (DEXA'04), IEEE Computer Society*, 2004.
9. A. Becher, Z. Benenson, and M. Dornseif. Tampering with Motes: Real-world Physical Attacks on Wireless Sensor Networks. In *3rd International Conference on Security in Pervasive Computing (SPC 2006)*, York, UK, April 2006.
10. R. Beckwith, D. Teibel, and P. Bowen. Report from the field: Results from an agricultural wireless sensor network. In *Proceedings of the 1st IEEE Workshop on Embedded Networked Sensors (EmNetS-I 2004)*, Tampa, USA, November 2004.
11. M. Blaze, J. Feigenbaum, and A. D. Keromytis. KeyNote: Trust Management for Public-Key Infrastructures (position paper). *Lecture Notes in Computer Science*, 1550:59–63, 1999.
12. M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized Trust Management. In *IEEE Symposium on Security and Privacy*, 1996.
13. Haiguang Chen, Huafeng Wu, Xi Zhou, and Chuanshan Gao. Reputation-based Trust in Wireless Sensor Netwoks. In *In Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering (MUE'07)*, 2007.
14. Y.-H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. REFEREE: Trust Management for Web Applications. *Computer Networks and ISDN Systems*, 29:953–964, 1997.

15. Garth V. Crosby, Niki Pissinou, and James Gadze. A Framework for Trust-based Cluster Head Election in Wireless Sensor Networks. In *In Proceedings of the Second IEEE Workshop on Dependability in Sensor Networks and Systems (DSSNS'06)*. IEEE Computer Society, 2006.
16. S. Ganeriwal and M. B. Srivastava. Reputation-Based Framework for High Integrity Sensor Networks. In *2nd ACM Workshop on Security of Ad Hoc and Sensor Networks.*, pages 66–77, Washington, DC, USA, 2004.
17. A. Josang and R. Ismail. The Beta Reputation System. In *15th Bled Electronic Commerce Conference e-Reality: Constructing the e-Economy*, Bled, Slovenia, June 2002.
18. A. Josang, R. Ismail, and C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*, 2006.
19. A. Jsang and S. J. Knapskog. A metric for trusted systems. In *21st National Information Systems Security Conference (NIST-NCSC 1998)*, 1998.
20. S. D. Kamwar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proceedings of the 12th International Conference on World Wide Web (WWW'03)*, ACM Press, pages 640–651, Budepest, Hungary, 2003.
21. C. Karlof and D. Wagner. Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. *Elsevier's Ad Hoc Networks Journal, Special Issue on Sensor Network Applications and Protocols*, pages 293–315, September 2003.
22. Chris Karlof, Naveen Sastry, and David Wagner. Tinysec: a link layer security architecture for wireless sensor networks. In *Proceedings of the 2nd international conference on Embedded networked sensor systems (SenSys '04)*, pages 162–175, New York, NY, USA, November 2004. ACM Press.
23. Mark Krasniewski, Padma Varadharajan, Bryan Rabeler, Saurabh Bagchi, and Y. Charlie Hu. Tibfit: Trust index based fault tolerance for arbitrary data faults in sensor networks. In *International Conference on Dependable Systems and Networks (DSN'05)*, pages 672–681, Los Alamitos, CA, USA, 2005. IEEE Computer Society.
24. A. Lawen. Apoptosis – An Introduction. *BioEssays*, 25:888–896, 2000.
25. Z. Liu, A. W. Joy, and R. A. Thompson. A Dynamic Trust Model for Mobile Ad-hoc Networks. In *10th IEEE International Workshop on Future Trends of Distributed Computing Systems*, pages 80–85, Suzhou, China, May 2004.
26. D. Ma and G. Tsudik. Forward-secure Sequential Aggregate Authentication. In *IEEE Symposium on Security and Privacy (S&P'07)*, Oakland, CA, May 2007.
27. M. Manzo, T. Roosta, and S. Sastry. Time Synchronization Attacks in Sensor Networks. In *3th ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2005)*, Alexandria, USA, October 2006.
28. D.H. McKnight and N. L. Chervany. The Meanings of Trust. Technical Report 96-04, MISRC Working Paper Series, University of Minesota, Management Information Systems Research Center, 1996.
29. D. Minder, P. J. Marrón, A. Lachenmann, and K. Rothermel. Experimental construction of a meeting model for smart office environments. In *Proceedings of the First Workshop on Real-World Wireless Sensor Networks (REALWSN 2005)*, Stockholm, Sweden, June 2005.
30. T. Park and K. G. Shin. Soft Tamper-Proofing via Program Integrity Verification in Wireless Sensor Networks. *IEEE Transactions on Mobile Computing*, 4(3):297–309, May-June 2005.
31. Y. Rebahi, V. E. Mujica-V, and D. Sisalem. A Reputation-Based Trust Mechanism for Ad-hoc Networks. In *10th IEEE Symposium on Computers and Communications (ISCC 2005)*, 2005.

32. T Ryutov and C Neuman. Trust-based Approach for Improving Data Reliability in Industrial Sensor Networks. In Marsh-S. (Boston: Springer) Etalle, S., editor, *In IFIP International Federation for Information Processing, Trust Management*, volume 238, pages 349–365, 2007.
33. J. Sabater and C. Sierra. REGRET: A Reputation Model for Gregarious Societies. In *Fourth Workshop on Deception Fraud and Trust in Agent Societies*, ACM Press, 2001.
34. Y. Sang, H. Shen, Y. Inoguchi, Y. Tan, and N. Xiong. Secure Data Aggregation in Wireless Sensor Networks: A Survey. In *7th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT 2006)*, Taipei, Taiwan, December 2006.
35. Riaz Ahmed Shaik, Hassan Jameel, Sungyoung Lee, Saeed Rajput, and Young Jae Song. Trust Management Problem in Distributed Wireless Sensor Networks. In *12th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA'06)*, IEEE Computer Society, 2006.
36. R. Sherwood, L. Seungjoon, and B. Bhattacharjee. Cooperative peer groups in nice. *Computer Networks*, 50(4):523–544, 2006.
37. A. Singh and L. Liu. TrustMe: Anonymous Management of Trust Relationships in Decentralized P2P Systems. In *Third International Conference on Peer-to-Peer Computing (P2P'03)*. IEEE, 2003.
38. A. Srinivasan, J. Teitelbaum, H. Liang, J. Wu, and M. Cardei. Reputation and Trust-based Systems for Ad Hoc and Sensor Networks. In A. Boukerche, editor, *On Trust Establishment in Mobile Ad-Hoc Networks*. Wiley & Sons, 2007.
39. N. Stakhanove, S. Basu, J. Wong, and O. Stakhanov. Trust Framework for P2P Networks using Peer-Profile based Anomaly Technique. In *25th IEEE International Conference on Distributed Computing Systems Workshops (ICDCSW'05)*. IEEE, 2005.
40. M. Strasser and H. Vogt. Autonomous and distributed node recovery in wireless sensor networks. In *Proceedings of The 4th ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN 2006)*, Alexandria, USA, October 2006.
41. S. Tanachaiwiwat, P. Dave, R. Bhindwale, and A. Helmy. Location-centric Isolation of Misbehavior and Trust Routing in Energy-Constrained Sensor Networks. In *IEEE COnference on Performance, Computing and Communications*, pages 463–469, 2003.
42. H. Wang and Q. Li. Efficient Implementation of Public Key Cryptosystems on MICAz and TelosB Motes. Technical Report WM-CS-2006-07, College of William & Mary, October 2006.
43. Y. Wang and J. Vassileva. Trust and Reputation Model in Peer-to- Peer Networks. In *Third International Conference on Peer-to-Peer Computing (P2P'03)*, 2003.
44. G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, and M. Welsh. Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing, Special Issue on Data-Driven Applications in Sensor Networks*, March-April 2006.
45. L. Xiong and L. Liu. Peertrust: Supporting reputation-based trust for peer-to peer electronic communities. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):843–857, 2004.
46. Z. Yan, P. Zhang, and T. Virtanen. Trust Evaluation Based Security Solutions in Ad-hoc Networks. In *NordSec 2003, Proceedings of the Seventh Nordic Workshop on Security IT Systems*, 2003.

47. W. Shi Z. Liang, and. PET: A Personalized Trust Model with Reputation and Risk Evaluation for P2P Resource Sharing. In *38th Hawaii International Conference on System Sciences*, 2005.
48. G. Zacharia and P. Maes. Trust management through reputation mechanisms. *Applied Artificial Intelligence*, 14(9):881–907, 2000.
49. Wei Zhang, SajalK. Das, and Yonghe Liu. A Trust Based Framework for Secure Data Aggregation in Wireless Sensor Networks. In *In Proceedings of the IEEE SECON 2006*, Reston, VA, September 2006.
50. H. Zhu, F. Bao, and K. Kim. Computing of Trust in Wireless Networks. In *60th IEEE Vehicular Technology Conference*, Los Angeles, California, September 2004.
51. Z.Yao, D. Kim, I. Lee, K. Kim, and J. Jang. A Security Framework with Trust Management for Sensor Networks. In *Workshop of the 1st International Conference on Security and Privacy for Emerging Areas in Communication Networks*, pages 190–198, 2005.