UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

TESIS DOCTORAL

Trust-ware: A Methodology to Analyze, Design, and Secure Trust and Reputation Systems

Autor:

David Fraga Aydillo

Ingeniero de Telecomunicación

Directores:

José Manuel Moya Fernández

Doctor Ingeniero de Telecomunicación

Zorana Banković

Doctor Ingeniero de Telecomunicación

2015

David Fraga Aydillo
E-mail: dfraga@die.upm.es

Ph.D. Thesis

Título:	Trust-ware: A Methodology to Analyze, Design, And Secure Trust and Reputation Systems
Autor:	DAVID FRAGA AYDILLO
Tutor:	JOSÉ MANUEL MOYA FERNÁNDEZ ZORANA BANKOVIĆ
Departamento:	DEPARTAMENTO DE INGENIERÍA ELECTRÓNICA

Miembros del tribunal:

Presidente:
Secretario:
Vocal:
Vocal:
Vocal:

Suplente:
Suplente:

Los miembros del tribunal arriba nombrados acuerdan otorgar la calificación de:

Madrid, de de 2015

"Trust is the glue of life. It's the most essential ingredient in effective communication. It's the foundational principle that holds all relationships."

— Stephen Covey

Acknowledgments

"If I have seen a little further it is by standing on the shoulders of Giants."

— Isaac Newton

El camino que ha sido necesario recorrer para llegar hasta aquí no ha sido corto. De hecho, ha sido muy largo. Quizás no tanto en duración, pero sí en *intensidad*. Durante mucho tiempo llegué a pensar que nunca me vería en la situación de escribir estas líneas. Me alegro de haberme equivocado.

Gran parte de la culpa de que al final este trabajo haya podido llegar a su fin recae en toda esa gente que confió en mi incondicionalmente y siempre me apoyó a que siguiera adelante. Para todos ellos son estos agradecimientos.

En primer lugar quiero dar las gracias a josem. Si ya fué mi maestro, y auténtico referente, en el desarrollo de mi vida como ingeniero, ahora lo ha vuelto a conseguir en mi vida como investigador. Pragmatismo, optimismo, paciencia y dispersión a partes iguales. Aunque casi, no hay dos como él.

A continuación, mención especial para mis dos compañeros de venturas y desventuras en la línea de TRS: Zorana y Juan Carlos. No puedo tener mejor recuerdo de los meses que pudimos trabajar juntos. Una máquina perfectamente engrasada en la conseguimos que el equipo fuera incluso mejor que la suma de sus partes (que se dice pronto).

Por otro lado me gustaría agradecer todo su apoyo a esas otras dos *familias investigadoras* de las que he podio formar parte: la gente de GreenLSI, que me acogió en los momentos más duros para que pudiera continuar con mi investigación, y la gente del GTH del 039. Gracias por hacerme sentir tan a gusto entre vosotros (aunque fuera una anomalía).

Ahora es el turno de mis otras dos familias, las de verdad. Gracias a José Antonio y a Amalia, por su confianza y apoyo incondicionales y por ser los mejores padres que alguien podría tener, y gracias a Susana, por ser la sonrisa que siempre alegra nuestros días. Y por supuesto, gracias a Patricia, por estar siempre ahí. Tengo un millón de cosas que agradecerte, pero también toda una vida para hacerlo.

Ya por último, por no perder la costumbre, quería dar las gracias a *Los Hijos del Trueno* y a *Dani Filth* por hacer del mundo un lugar tan especial.

Abstract

"I suppose the most obvious question is: how can I trust you?"
— Neo talking to The Oracle, *The Matrix Reloaded*

By collective intelligence we understand a form of intelligence that emerges from the collaboration and competition of many individuals, or strictly speaking, many entities. Based on this simple definition, we can see how this concept is the field of study of a wide range of disciplines, such as sociology, information science or biology, each of them focused in different kinds of entities: human beings, computational resources, or animals.

As a common factor, we can point that collective intelligence has always had the goal of being able of promoting a group intelligence that overcomes the individual intelligence of the basic entities that constitute it. This can be accomplished through different mechanisms such as coordination, cooperation, competence, integration, differentiation, etc.

Collective intelligence has historically been developed in a parallel and independent way among the different disciplines that deal with it. However, this is not enough anymore due to the advances in information technologies. Nowadays, human beings and machines coexist in environments where collective intelligence has taken a new dimension: we yet have to achieve a better collective behavior than the individual one, but now we also have to deal with completely different kinds of individual intelligences. Therefore, we have a double goal: being able to deal with this heterogeneity and being able to get even more intelligent behaviors thanks to the synergies that the different kinds of intelligence can generate.

Within the areas of collective intelligence there are several open topics where they always try to get better performances from groups than from the individuals. For example: collective consciousness, collective memory, or collective wisdom. Among all these topics we will focus on collective decision making, that has influence in most of the collective intelligent behaviors.

The field of study of decision making is really wide, and its evolution has been completely parallel to the aforementioned collective intelligence. Firstly, it was focused on the individual as the main decision-making entity, but later it became involved in studying social and institutional groups as basic decision-making entities.

The first studies within the decision-making discipline were based on simple paradigms, such as pros and cons analysis, criteria prioritization, fulfillment, following orders, or even chance. However, in the same way that studying the community instead of the individual meant a paradigm shift within collective intelligence, collective decision-making means a new challenge for all the related disciplines. Besides, two new main topics come up when dealing with collective decision-making: centralized and decentralized decision-making systems. In this thesis project we focus in the second one, because it is the most interesting based on the opportunities to generate new knowledge and deal with open issues in this area, as well as these results can be put into practice in a wider set of real-life environments.

Finally, within the decentralized collective decision-making systems discipline, there are several basic mechanisms that lead to different approaches to the specific problems of this field, for example: leadership, imitation, prescription, or fear. We will focus on trust and reputation. They are one of the most multidisciplinary concepts and with more potential for applying them in every kind of environments. Besides, they have historically shown that they can generate better performance than other decentralized decision-making mechanisms.

Shortly, we say trust is the belief of one entity that the outcome of other entities' actions is going to be in a specific way. It is a subjective concept because the trust of two different entities in another one does not have to be the same.

Reputation is the collective idea (or social evaluation) that a group of entities within a system have about another entity based on a specific criterion. Thus, it is a collective concept in its origin.

It is important to say that the behavior of most of the collective systems are based on these two simple definitions. In fact, a lot of articles and essays describe how any organization would not be viable if the ideas of trust and reputation did not exist. From now on, we call Trust an Reputation System (TRS) to any kind of system that uses these concepts.

Even though TRSs are one of the most common everyday aspects in our lives, the existing knowledge about them could not be more dispersed. There are thousands of scientific works in every field of study related to trust and reputation: philosophy, psychology, sociology, economics, politics, information sciences, etc. But the main issue is that a comprehensive vision of trust and reputation for all these disciplines does not exist.

Every discipline focuses its studies on a specific set of topics but none of them tries to take advantage of the knowledge generated in the other disciplines to improve its behavior or performance. Detailed topics in some fields are completely obviated in others, and even though the study of some topics within several disciplines produces complementary results, these results are not used outside the discipline where they were generated.

This leads us to a very high knowledge dispersion and to a lack in the reuse of methodologies, policies and techniques among disciplines.

Due to its great importance, this high dispersion of trust and reputation knowledge is one of the main problems this thesis contributes to solve.

When we work with TRSs, all the aspects related to security are a constant since it is a vital aspect within the decision-making systems. Besides, TRS are often used to perform some responsibilities related to security. Finally, we cannot forget that the act of trusting is invariably attached to the act of delegating a specific responsibility and, when we deal with these concepts, the idea of risk is always present. This refers to the risk of generated expectations not being accomplished or being accomplished in a different way we anticipated.

Thus, we can see that any system using trust to improve or enable its behavior, because of its own nature, is especially vulnerable if the premises it is based on are attacked.

Related to this topic, we can see that the approaches of the different disciplines that study attacks of trust and reputation are very diverse. Some attempts of using approaches of other disciplines have been made within the information science area of knowledge, but these approaches are usually incomplete, not systematic and oriented to achieve specific requirements of specific applications. They never try to consolidate a common base of knowledge that could be reusable in other context.

Based on all these ideas, this work makes the following direct contributions to the field of TRS:

- The compilation of the most relevant existing knowledge related to trust and reputation management systems focusing on their advantages and disadvantages.
- We define a generic architecture for TRS, identifying the main entities and processes involved.
- We define a generic security framework for TRS. We identify the main security assets and propose a complete taxonomy of attacks for TRS.
- We propose and validate a methodology to analyze, design, secure and deploy TRS in real-life environments. Additionally we identify the principal kind of applications we can implement with TRS and how TRS can provide a specific functionality.
- We develop a software component to validate and optimize the behavior of a TRS in order to achieve a specific functionality or performance.

In addition to the contributions made directly to the field of the TRS, we have made original contributions to different areas of knowledge thanks to the application of the analysis, design and security methodologies previously presented:

- Detection of thermal anomalies in Data Centers. Thanks to the application of the TRS analysis and design methodologies, we successfully implemented a thermal anomaly detection system based on a TRS. We compare the detection performance of Self-Organized-Maps and Growing Neural Gas algorithms. We show how SOM provides better results for Computer Room Air Conditioning anomaly detection, yielding detection rates of 100%, in training data with malfunctioning sensors. We also show that GNG yields better detection and isolation rates for workload anomaly detection, reducing the false positive rate when compared to SOM.

- Improving the performance of a harvesting system based on swarm computing and social odometry. Through the implementation of a TRS, we achieved to improve the ability of coordinating a distributed network of autonomous robots. The main contribution lies in the analysis and validation of the incremental improvements that can be achieved with proper use information that exist in the system and that are relevant for the TRS, and the implementation of the appropriated trust algorithms based on such information.

- Improving Wireless Mesh Networks security against attacks against the integrity, confidentiality or availability of data and communications supported by these networks. Thanks to the implementation of a TRS we improved the detection time rate against these kind of attacks and we limited their potential impact over the system.

- We improved the security of Wireless Sensor Networks against advanced attacks, such as insider attacks, unknown attacks, etc.

  Thanks to the TRS analysis and design methodologies previously described, we implemented countermeasures against such attacks in a complex environment. In our experiments we have demonstrated that our system is capable of detecting and confining various attacks that affect the core network protocols. We have also demonstrated that our approach is capable of rapid attack detection. Also, it has been proven that the inclusion of the proposed detection mechanisms significantly increases the effort the attacker has to introduce in order to compromise the network.

Finally we can conclude that, to all intents and purposes, this thesis offers a useful and applicable knowledge in real-life environments that allows us to maximize the performance of any system based on a TRS.

Thus, we deal with the main deficiency of this discipline: the lack of a common and complete base of knowledge and the lack of a methodology for the development of TRS that allow us to analyze, design, secure and deploy TRS in a systematic way.

Resumen

Entendemos por inteligencia colectiva una forma de inteligencia que surge de la colaboración y la participación de varios individuos o, siendo más estrictos, varias entidades. En base a esta sencilla definición podemos observar que este concepto es campo de estudio de las más diversas disciplinas como pueden ser la sociología, las tecnologías de la información o la biología, atendiendo cada una de ellas a un tipo de entidades diferentes: seres humanos, elementos de computación o animales.

Como elemento común podríamos indicar que la inteligencia colectiva ha tenido como objetivo el ser capaz de fomentar una inteligencia de grupo que supere a la inteligencia individual de las entidades que lo forman a través de mecanismos de coordinación, cooperación, competencia, integración, diferenciación, etc.

Sin embargo, aunque históricamente la inteligencia colectiva se ha podido desarrollar de forma paralela e independiente en las distintas disciplinas que la tratan, en la actualidad, los avances en las tecnologías de la información han provocado que esto ya no sea suficiente. Hoy en día seres humanos y máquinas a través de todo tipo de redes de comunicación e interfaces, conviven en un entorno en el que la inteligencia colectiva ha cobrado una nueva dimensión: ya no sólo puede intentar obtener un comportamiento superior al de sus entidades constituyentes sino que ahora, además, estas inteligencias individuales son completamente diferentes unas de otras y aparece por lo tanto el doble reto de ser capaces de gestionar esta gran heterogeneidad y al mismo tiempo ser capaces de obtener comportamientos aún más inteligentes gracias a las sinergias que los distintos tipos de inteligencias pueden generar.

Dentro de las áreas de trabajo de la inteligencia colectiva existen varios campos abiertos en los que siempre se intenta obtener unas prestaciones superiores a las de los individuos. Por ejemplo: consciencia colectiva, memoria colectiva o sabiduría colectiva. Entre todos estos campos nosotros nos centraremos en uno que tiene presencia en la práctica totalidad de posibles comportamientos inteligentes: la toma de decisiones.

El campo de estudio de la toma de decisiones es realmente amplio y dentro del mismo la evolución ha sido completamente paralela a la que citábamos anteriormente en referencia a la inteligencia colectiva. En primer lugar se centró en el individuo como entidad decisoria para posteriormente desarrollarse desde un punto de vista social, institucional, etc.

La primera fase dentro del estudio de la toma de decisiones se basó en la utilización de paradigmas muy sencillos: análisis de ventajas e inconvenientes, priorización basada en la maximización de algún parámetro del resultado, capacidad para satisfacer los requisitos de forma mínima por parte de las alternativas, consultas a expertos o entidades autorizadas o incluso el azar. Sin embargo, al igual que el paso del estudio del individuo al grupo supone una nueva dimensión dentro la inteligencia colectiva la toma de decisiones colectiva supone un nuevo reto en todas las disciplinas relacionadas. Además, dentro de la decisión colectiva aparecen dos nuevos frentes: los sistemas de decisión centralizados y descentralizados. En el presente proyecto de tesis nos centraremos en este segundo, que es el que supone una mayor atractivo tanto por las posibilidades de generar nuevo conocimiento y trabajar con problemas abiertos actualmente así como en lo que respecta a la aplicabilidad de los resultados que puedan obtenerse.

Ya por último, dentro del campo de los sistemas de decisión descentralizados existen varios mecanismos fundamentales que dan lugar a distintas aproximaciones a la problemática propia de este campo. Por ejemplo el liderazgo, la imitación, la prescripción o el miedo. Nosotros nos

centraremos en uno de los más multidisciplinares y con mayor capacidad de aplicación en todo tipo de disciplinas y que, históricamente, ha demostrado que puede dar lugar a prestaciones muy superiores a otros tipos de mecanismos de decisión descentralizados: la confianza y la reputación.

Resumidamente podríamos indicar que confianza es la creencia por parte de una entidad que otra va a realizar una determinada actividad de una forma concreta. En principio es algo subjetivo, ya que la confianza de dos entidades diferentes sobre una tercera no tiene porqué ser la misma.

Por otro lado, la reputación es la idea colectiva (o evaluación social) que distintas entidades de un sistema tiene sobre otra entidad del mismo en lo que respecta a un determinado criterio. Es por tanto una información de carácter colectivo pero única dentro de un sistema, no asociada a cada una de las entidades del sistema sino por igual a todas ellas.

En estas dos sencillas definiciones se basan la inmensa mayoría de sistemas colectivos. De hecho muchas disertaciones indican que ningún tipo de organización podría ser viable de no ser por la existencia y la utilización de los conceptos de confianza y reputación. A partir de ahora, a todo sistema que utilice de una u otra forma estos conceptos lo denominaremos como sistema de confianza y reputación (o TRS, Trust and Reputation System).

Sin embargo, aunque los TRS son uno de los aspectos de nuestras vidas más cotidianos y con un mayor campo de aplicación, el conocimiento que existe actualmente sobre ellos no podría ser más disperso.

Existen un gran número de trabajos científicos en todo tipo de áreas de conocimiento: filosofía, psicología, sociología, economía, política, tecnologías de la información, etc. Pero el principal problema es que **no existe una visión completa de la confianza y reputación** en su sentido más amplio.

Cada disciplina focaliza sus estudios en unos aspectos u otros dentro de los TRS, pero ninguna de ellas trata de explotar el conocimiento generado en el resto para mejorar sus prestaciones en su campo de aplicación concreto. Aspectos muy detallados en algunas áreas de conocimiento son completamente obviados por otras, o incluso aspectos tratados por distintas disciplinas, al ser estudiados desde distintos puntos de vista arrojan resultados complementarios que, sin embargo, no son aprovechados fuera de dichas áreas de conocimiento.

Esto nos lleva a una **dispersión de conocimiento** muy elevada y a una **falta de reutilización de metodologías, políticas de actuación y técnicas** de una disciplina a otra.

Debido su vital importancia, esta alta dispersión de conocimiento se trata de uno de los principales problemas que se pretenden resolver con el presente trabajo de tesis.

Por otro lado, cuando se trabaja con TRS, todos los aspectos relacionados con la seguridad están muy presentes ya que muy este es un tema vital dentro del campo de la toma de decisiones. Además también es habitual que los TRS se utilicen para desempeñar responsabilidades que aportan algún tipo de funcionalidad relacionada con el mundo de la seguridad. Por último no podemos olvidar que el acto de confiar está indefectiblemente unido al de delegar una determinada responsabilidad, y que al tratar estos conceptos siempre aparece la idea de riesgo, riesgo de que las expectativas generadas por el acto de la delegación no se cumplan o se cumplan de forma diferente.

Podemos ver por lo tanto que cualquier sistema que utiliza la confianza para mejorar o posibilitar su funcionamiento, por su propia naturaleza, es especialmente vulnerable si las premisas en las que se basa son atacadas.

En este sentido podemos comprobar (tal y como analizaremos en más detalle a lo largo del presente documento) que las aproximaciones que realizan las distintas disciplinas que tratan la violación de los sistemas de confianza es de lo más variado. únicamente dentro del área de las tecnologías de la información se ha intentado utilizar alguno de los enfoques de otras disciplinas de cara a afrontar problemas relacionados con la seguridad de TRS. Sin embargo se trata de una aproximación incompleta y, normalmente, realizada para cumplir requisitos de aplicaciones concretas y no con la idea de afianzar una base de conocimiento más general y reutilizable en otros entornos.

Con todo esto en cuenta, podemos resumir contribuciones del presente trabajo de tesis en las siguientes.

- La realización de un completo **análisis del estado del arte dentro del mundo de la confianza y la reputación** que nos permite comparar las ventajas e inconvenientes de las diferentes aproximación que se realizan a estos conceptos en distintas áreas de conocimiento.
- La definición de **una arquitectura de referencia para TRS** que contempla todas las entidades y procesos que intervienen en este tipo de sistemas.
- La definición de un **marco de referencia para analizar la seguridad de TRS**. Esto implica tanto identificar los principales activos de un TRS en lo que respecta a la seguridad, así como el crear una tipología de posibles ataques y contramedidas en base a dichos activos.
- La **propuesta de una metodología para el análisis, el diseño, el aseguramiento y el despliegue de un TRS** en entornos reales. Adicionalmente se exponen los principales tipos de aplicaciones que pueden obtenerse de los TRS y los medios para maximizar sus prestaciones en cada una de ellas.
- La generación de un **software que permite simular cualquier tipo de TRS** en base a la arquitectura propuesta previamente. Esto permite evaluar las prestaciones de un TRS bajo una determinada configuración en un entorno controlado previamente a su despliegue en un entorno real. Igualmente es de gran utilidad para evaluar la resistencia a distintos tipos de ataques o mal-funcionamientos del sistema.

Además de las contribuciones realizadas directamente en el campo de los TRS, hemos realizado aportaciones originales a distintas áreas de conocimiento gracias a la aplicación de las metodologías de análisis y diseño citadas con anterioridad.

- Detección de anomalías térmicas en Data Centers. Hemos implementado con éxito un sistema de deteción de anomalías térmicas basado en un TRS. Comparamos la detección de prestaciones de algoritmos de tipo Self-Organized Maps (SOM) y Growing Neural Gas (GNG). Mostramos como SOM ofrece mejores resultados para anomalías en los sistemas de refrigeración de la sala mientras que GNG es una opción más adecuada debido a sus tasas de detección y aislamiento para casos de anomalías provocadas por una carga de trabajo excesiva.
- Mejora de las prestaciones de recolección de un sistema basado en swarm computing y odometría social. Gracias a la implementación de un TRS conseguimos mejorar las capacidades de coordinación de una red de robots autónomos distribuidos. La principal contribución reside en el análisis y la validación de las mejoras incrementales que pueden conseguirse con la utilización apropiada de la información existente en el sistema y que puede ser relevante desde el punto de vista de un TRS, y con la implementación de algoritmos de cálculo de confianza basados en dicha información.
- Mejora de la seguridad de Wireless Mesh Networks contra ataques contra la integridad, la confidencialidad o la disponibilidad de los datos y/o comunicaciones soportadas por dichas redes.
- Mejora de la seguridad de Wireless Sensor Networks contra ataques avanzamos, como insider attacks, ataques desconocidos, etc. Gracias a las metodologías presentadas implementamos contramedidas contra este tipo de ataques en entornos complejos. En base a los experimentos realizados, hemos demostrado que nuestra aproximación es capaz de detectar y confinar varios tipos de ataques que afectan a los protocoles esenciales de la red. La propuesta ofrece unas velocidades de detección muy altas así como demuestra que la inclusión de estos mecanismos de actuación temprana incrementa significativamente el esfuerzo que un atacante tiene que introducir para comprometer la red.

Finalmente podríamos concluir que el presente trabajo de tesis supone la **generación de un conocimiento útil y aplicable a entornos reales, que nos permite la maximización de las prestaciones resultantes de la utilización de TRS en cualquier tipo de campo de aplicación**.

De esta forma cubrimos la principal carencia existente actualmente en este campo, que es la falta de una base de conocimiento común y agregada y la inexistencia de una metodología para el desarrollo de TRS que nos permita analizar, diseñar, asegurar y desplegar TRS de una forma sistemática y no artesanal y ad-hoc como se hace en la actualidad.

Contents

List of Tables

List of Figures

1. Introduction

"You must trust and believe in people or life becomes impossible."

— Anton Chekhov

This introductory Chapter presents the motivation, problem context and a brief state of the art on the work presented in this Ph.D. Thesis. Besides, the main contributions of this work are highlighted and an overview of the structure of this Ph.D. Thesis is also provided.

1.1 Motivation

The study of Trust and Reputation Systems (TRS) is a discipline that belongs to the field of decentralized decision-making techniques. They have a wide range of uses, and they are increasing their importance as heterogeneous and distributed systems are more and more present in any facet of our lives.

However, there is not a systematic way to analyze, design, secure and deploy TRS into real-life scenarios. Therefore, working with TRS becomes more a craft-work than an engineering process.

Even though TRSs are one of the most common everyday aspects in our lives, the existing knowledge about them cannot be more dispersed. There are thousands of works in every field of study related to trust and reputation: philosophy, psychology, sociology, economics, politics, information sciences, etc. But the main issue is that a comprehensive vision of trust and reputation for all these disciplines does not exist.

Every discipline focuses its studies on a specific set of topics but none of them tries to take advantage of the knowledge generated in the other disciplines to improve its behavior or performance. Detailed topics in some fields are completely obviated in others, and even though the study of some topics within several disciplines produce complementary results, these results are not used outside the discipline where they were generated. This lead us to a very high knowledge dispersion and to a lack in the reuse of methodologies, policies and techniques among disciplines.

This Ph.D. Thesis addresses the definition of a set of conceptual models and methodologies in order to allow a more precise, systematic, complete, and secure analysis and design of this kind of systems.

1.2 Context

In order to explain the context where this thesis has been developed, more details on the state-of-the art of the main topics related to this work are given in the next sections. Section 1.2.1

gives an overview of the main topics regarding Collective Intelligence. Then, Section 1.2.2 describes some classical approaches to decentralized decision-making techniques. Finally, Section 1.2.3 details how different fields of knowledge study trust and reputation.

1.2.1 Collective Intelligence

By collective intelligence [1]–[3] we understand a form of intelligence that emerges from the collaboration and competition [4] of many individuals, or strictly speaking, many entities. Based on this simple definition, we can see how this concept is the field of study of a very wide range of disciplines, such as sociology, information science or biology. Each of them focused in different kinds of entities: human beings, computational resources, or animals.

As a common factor we point that collective intelligence has always had the goal of being able of promoting a group intelligence that overcomes the individual intelligence of the basic entities that constitute it. This can be accomplished through different mechanisms such as coordination, cooperation, competence, integration, differentiation, etc. It can also be understood as an emergent property from synergies among information, knowledge, software, hardware, and living entities that continuosly learns from feedback to produce knowledge for better decisions than these elements acting alone.

The idea emerged from the writings of Douglas Hofstadter [5], Pierre Levi [6], Howard Bloom [7], Francis Heylighen [7], Douglas Phillip Brown [1] and other theorists and writers.

In order to understand the proposed thesis, we need to identify some of the constitutive elements of collective intelligence. Specifically we are going to focus our analysis on who are the entities involved in any collective intelligence behavior, and how they try to achieve their goals [8].

Dealing with collective intelligence organization, we can find two basic structures: hierarchy and crowd [2], [9].

In traditional hierarchical organizations [10], someone in authority assigns a particular person or group of people to perform a task. In this way, the collective behavior is driven by an individual intelligence.

In crowd collectives [11], activities can be assumed by anyone in a large group who chooses to do so, without being assigned by someone in a position of authority. In this way, crowd becomes a central feature of any collective intelligence system.

Based on this feature, we will analyze the behavior of different decentralized decision-making techniques, where TRSs are included.

1.2.2 Decentralized Decision-making Techniques

Introduction

There are two possible alternatives related to decentralized decision-making techniques when talking about decisions that are made by crowd collectives, namely Individual Decisions and Group Decisions.

The Individual Decision occurs when members of a crowd make decisions that, though informed by a crowd input, do not need to be identical for all of them. On the other hand, Group Decision [12], [13] occurs when inputs from members of the crowd are assembled to generate a decision that holds for the group as a whole.

Based on these two vectors we can identify some paradigmatic approaches.

Social Networks

Social Networks are one of the most important techniques from the point of view of our analysis.

In Social Networks, members of a crowd form a network of relationships that might be translated into levels of trust, similarity of taste and viewpoints, or other common characteristics that might cause individuals to feel an affinity for one another.

Then, crowd entities assign different weights to individual inputs on the basis of their relationship with the entities who provided them and then make individual decisions. Among many other applications, social network relationships and this preference-making behavior, are extremely useful to implement collaborative filtering/decision-making techniques [14].

Markets

In Markets, there is some kind of formal exchange (usually money) involved in the decisions. Each entity of the crowd makes an individual decision about what products to buy or sell. All purchasing decisions made by buyers in the crowd together, determine the collective demand. This demand affects the availability of products and, therefore, their prices. On the other hand, the quantities and prices of the goods that are sold in the system influence the purchasing decisions, and so on.

SWARM Intelligence

Swarm Intelligence systems [15] are one of the most studied and applied collective intelligence behaviors. Besides, they are specially interesting from the point of view of our analysis because they are a link between individual decision and group decision techniques. Each entity or individual makes decisions based on its knowledge of the system, but it is the collection of all these individual decisions what makes possible to solve the global objective.

Swarm Intelligence systems are typically made up of a population of simple agents interacting locally with one another and with their environment. The group of individuals acting in such a manner is referred to as a swarm. Individuals within the group interact by exchanging locally available information such that the global objective is solved more efficiently than if it is done by a single individual. Therefore, decision-making or problem-solving behavior that emerges from such interactions is called swarm intelligence. Thus, structures and solutions appear at the global level of a system from interactions among its lower-level components.

The basic components of this kind of self-organizing mechanisms are:

- Positive feedback: examples are recruitment and reinforcement
- Negative feedback: counterbalances positive feedback. Examples are saturation, exhaustion or competition.
- Amplification and randomness: it enables to discover new non-trivial solutions.
- Multiple interactions: a minimal density and amount of individuals are required to create an effective swarm intelligence.

Averaging

Averaging belongs to the category of group decisions [16]. It is very common in cases where decisions involve picking a number. The most common behavior is to average the numbers contributed by the members of the crowd.

Averaging is commonly used in systems that rely on a point scale for quality rating. For example, most of web-based collective systems, such as Amazon [17], IMDB [18], hotel booking systems, etc. are based on this kind of techniques.

Consensus

Consensus means that all group members agree on the final decision. It seems like a complex technique in order to achieve a stable decision, but it is very common in some on-line systems, such as Wikipedia [11], [19]–[21], where articles that remain unchanged are those for which everyone who cares is satisfied with the current version.

Another example is reCAPTCHA [22]. reCAPTCHA is a Web security, where two words are displayed on the screen. Users are asked to type both to gain access to a Web page. One

of the words is a security key and the other a word previously scanned as part of a project to digitize old books. The words that the recognition software finds difficult to read are served to several users as one half of each reCAPTCHA. When transcriptions provided by multiple users reach a level of consensus, that word is considered to have been correctly transcribed.

Voting systems

New technologies make the voting techniques [23] feasible in many situations where it would not otherwise have been practical. The technique is quite simple: the most voted option becomes the chosen one. It is used in a wide range of applications: from news websites [24] to collective chess matches [25].

Two important sub-variations are implicit voting and weighted voting. In implicit voting, some actions (different to the act of voting) are counted as votes, because they show some level of preference related to the entities or items involved in the action (*e.g.*,the number of times a photography has been downloaded can be used to choose the rank of most popular photographs). In weighted voting, the weight of the votes depends on their source (*e.g.*,the Google's Page-Rank algorithm [26] gives more weight to links from sites that are, themselves, more popular).

Prediction Markets

A useful way of letting crowds estimate the probability of future events is with prediction markets. In prediction markets [27], [28], people buy and sell shares or options of predictions about future events. If their predictions are correct, they are monetarily rewarded, either with real money or with some kind of bonus or points that can be redeemed for prizes or cash.

They are extremely useful when crowd is considered to have enough knowledge to solve a problem, but it is needed to bring this decentralized knowledge to the attention of these people who can act on in.

Trust and Reputation Systems

TRS stands as one of the most wide spectrum and most common collective decision-making techniques It is based on the well-known concepts of *trust* and *reputation*.

Shortly, we say trust is the belief of one entity that the outcome of other entities' actions are going to be in a specific way. It is a subjective concept because the trust of two different entities in another one does not have to be the same. In this sense, trust-based system are a paradigmatic example of individual decision systems.

On the other hand, reputation is the collective idea that a group of entities within a system have about another entity based on a specific criterion. Thus, it is a collective concept in its origin but it is not different for every single entity. It rather has the same value for all the entities throughout the system. Therefore, reputation-based systems are a clear example of group decision systems.

The behavior of TRSs can be described as follows [29]. The TRS assigns a lower reputation to the entities where it detects anomalous activities, or a bad throughput, or any other ratio that can be interesting in order to evaluate the performance of the whole system. Besides, every entity is being examined by at least another entity, that generates a subjective value of *trust*. Furthermore, entities advocate avoiding any contact with the entities that have low trust or reputation. In this way, the anomalous entities remain isolated from the system and has no role in its further operation [30].

1.2.3 T&R in different fields of knowledge

After giving an overview of the main topics regarding Collective Intelligence, describing some classical approaches to decentralized decision-making techniques, and defining the basic concepts related to TRSs, this section details how different fields of knowledge study *trust* and *reputation*.

Introduction

Even though TRSs are one of the most common everyday aspects in our life's, the existing knowledge about them cannot be more dispersed. There are thousands of scientific works in every field of study related to *trust* and *reputation*: philosophy, psychology, sociology, economics, politics, information sciences, etc. But the main issue is that a comprehensive vision of trust and reputation for all these disciplines does not exist.

Every discipline focuses its studies on a specific set of topics but none of them try to take advantage of the knowledge generated in the others disciplines to improve its behavior or performance. Detailed topics in some fields are completely obviated in others, and even though the study of some topics within several disciplines produce complementary results, these results are not used outside the discipline where they were generated.

This lead us to a very high knowledge dispersion and to a lack in the reuse of methodologies, policies and techniques among disciplines.

Due to its great importance, this high dispersion of trust and reputation knowledge is one of the main problems this thesis will try to solve.

T&R in Philosophy

According to the Stanford Encyclopedia of Philosophy, "trust is important but dangerous". Since trust allows us to form relationships with others and to rely on others for advice, help, etc., trust is regarded as a very important factor in our life that compels others to give us such things in an altruistic way, with no outside force such as the law [31].

On the other hand, since trust requires taking a risk that the trustee may not behave as the trustor expects, trust is dangerous implying the possible betrayal of trust.

In Lagerspetz's book titled Trust: The Tacit Demand [32], it is described the author's view on trust as a moral relationship in human society. Langerspetz believes that investigations of trust reveal that human individuals, their beliefs, desires and actions are only intelligible against the background of existing social practices and social ties.

Thus, trustful or betrayal actions can occur between a trustor and a trustee based on nature of their relationships, in this case, their personal relationships.

In this way, we can see that **trust**, **loyalty**, **moral boundaries** and **betrayal** are the most studied topics of TRS from the point of view of Philosophy.

T&R in Psychology

From the point of view of Psychology, trust starts from the moment of birth of the child. As the child grows older, trust also grows stronger. However, the root of trust derives from the relationship between the mother of the child, since the strength of the family relies on trust. If the child is raised in a family which is very accepting and loving, the child also returns those feelings to others by trusting them. But if trust is lost, it is hard to recover it again.

In this sense, trust in psychology emphasizes the cognitive process that **human beings learn trust from their experiences**.

Deutsch [33] defines trust as the confidence that one will find what is desired from another rather than what is feared. In addition, Hardin [34] and Rotter [35] observed in their experiments that **past experiences** may affect later capacity for trust. For example, bad experience with people will lower the trust level, leading to fewer relationships with people [36].

Besides, high trustors are less likely to lie or cheat or steal. Also they are less likely to be unhappy, conflicted, or unstable. Even though high trustors are deceived more often in novel or unknown situations, low trustors are also losing effectiveness in their relationships by distrusting trustworthy people, thereby losing the advantages that high trustors may have [35].

In this way, we can see that **trust as a learning process**, **memory** and the effects of **trust/distrust** are some of the most studied topics of TRSs from the point of view of Psychology.

T&R in Sociology

The Italian social scientist Diego Gambetta is one of the most influential researchers in this field. Gambetta's notion of trust [37] is popularly called sociological trust and is defined as an assessor's a priori subjective probability that a person (or agent, or group) will perform specific actions that affect the assessor.

Thus, Gambetta [37] describes the nature of trust as subjectivity, an indicator for future actions, and dynamism based on continuous interactions between two entities.

Adams et al. [38] rephrased Gambetta's trust concept quantifying trust based on the **acceptance of risk**. Thus, he stressed that risking betrayal is an important aspect in building trust.

Luhmann [39] also emphasized the importance of trust in society as a mechanism for **building cooperation** among people to extend human interactions for future collaboration.

The last main concept we can draw from sociology is based on the importance of prejudices or preconceptions. Tajfel, H. and Turner J.C. [40] describe these concepts as a way for social groups to build and reinforce relationships among their members (in-group favoritism), even though, at the same time, they are a way to exclude and degrade relationships with those not belonging to the group (out-group derogation).

In this way, we can identify that **subjectivity**, **prejudices/preconceptions acceptance of risk**, and trust as a mechanism to **build cooperation and predict actions** are some of the most studied topics of TRSs from the point of view of Sociology.

T&R in Economy

Economy is one of the first fields that distinguishes between the personal or informal trust (that comes from your relationships), and the impersonal or institutionalized trust (that comes from your financial status). In fact, this institutionalized trust is closer to the concept of reputation than to the concept of trust.

In economics, trust is represented as an expectation that applies to situations in which trustors take **risky** actions under **uncertainty** or information **incompleteness** [41].

Besides, trust in economics is based on the assumption that humans are rational and maximizers of their own interest or incentives [42]. Thus, although the assumption of selfish entities is reasonable, altruistic behaviors can emerge from mechanisms that may be initially purely selfish [43] if the incentives for collaboration are high enough based on their interest.

The study of **redemption** mechanisms in another important topic that Economy deals with.

In this way, we can identify that **maximization**, **risk** and **redemption** are some of the most studied topics of TRSs from the point of view of Economy.

T&R in Organizational Management

In this field, the concept of trust is defined at different levels.

First, it can be applied to the relationship between employers and employees, or between team managers and workers. In this context trust is defined as the extent to which one party is willing to count on someone or something with a feeling of relative security in spite of possible negative consequences, emphasizing the possibility of facing risk [44].

Moreover, Organizational Management add a new facet to the meaning of trust. They identify efficiency and proficiency as two of the main components of trust [45]. Derived from this idea, they also explain that trust is not necessarily mutual and is not reciprocal.

Finally, trust in Organizational Management can give us insights on how to measure trust by investigating methods to measure ability, integrity, and benevolence of member of the organization or work team.

Thus, we can say that **proficiency**, **efficiency** and **measurability** are some of the most studied topics of TRSs from the point of view of Organizational Management.

T&R in Corporations

Nowadays, improving public reputation has become one of the highest priority challenges for corporations all around the world.

In this context, corporate reputation usually derives from terms such as innovation, financial soundness, the use of corporate assets and social responsibility [46], [47].

Obtaining the reputation based on the point of view of the general public, customers, employees, suppliers and investors is other common technique in this field [48], [49]. These models measure perceptions of an organization in terms of social expectations of dimensions such as products and services, vision and leadership, work place environment and social responsibility. Related to the responsibility topic, international standardization organizations have published standards such as ISO26000 [50].

Finally, other common approaches [51] try to identify the corporate personality through surveys to customers and employees in terms of their perceptions of organization's personality, focusing on dimensions such as agreeableness, competence and enterprise.

In this way, we can say that dealing with **disperse and diverse sources of information** and the **process of abstract information** are some of the most studied topics of TRSs in this field.

T&R in Personal Branding

Personal Branding derives from the concept of corporate reputation. Tom Peters wrote the first article [52] where personal branding was cited, The Brand Called You. He explored the evolution of career development, and exposed that instead of relying on a company for career guidance, it is up to the individuals to take ownership of their own brand [53].

The basic idea that underlies Personal Branding is understanding the unique attributes of the individuals (strengths, skills, values, and passions) and using them to separate them from their competitors. Thus, Personal Branding is becoming increasingly essential to entrepreneurs, consultants, or even corporate employees.

The basic process to develop a personal brand are: discover, create, communicate and maintain. These two last steps are the focus of all the trust and reputation analysis in this field, and they are based on common sense: depending on your target audience, it is needed to adapt your message to properly communicate your brand, it is useful to communicate past actions to create a more compelling and appealing future brand, etc.

Thus, we can say that **creating and maintaining** reputation is the most studied topic of TRSs from the point of view of Personal Branding.

T&R in Communications and Networking

The concept of trust has been always very common to communication and network protocol designers. Trust not only enables secure communications, but trust relationships among participating nodes are critical in building cooperative and collaborative environments to optimize system objectives, such as scalability, reconfigurability, reliability or fault tolerance, etc.

Classical trust frameworks in this field (policy-based trust) are based in cryptographic algorithms that support Public Key Infrastructures [54], [55], enable nodes to share secret keys or, in a more wide range, provide mechanisms to ensure the identification or authentication processes and the confidentiality and integrity of the communications.

Therefore, we can say that **promoting collaboration and improving performance** through the use of trust and reputation is the most studied topic of TRSs from the point of view of communications and networking. Policy-based trust scheme are also common in this field.

T&R in Ad-hoc Networks

The main feature of ad-hoc networks [56] is that they dynamically change their structure really quickly [57]. This means different entities join and leave the system very often.

Entities are continuously confronted with other unknown entities, which can be of a great help to them if they can collaborate with each other. But collaboration between unknown

entities is not fully utilized, due to the fear of not being trusted and the potential risk of such collaboration [58].

Trust relationships in this kind of networks are established, evolved, propagated and expired on the fly. So, they are very susceptible to attacks. Nevertheless, **fast trust-generation mechanisms** [59]–[61] are one of the main contributions of the ad-hoc networks to the global field of TRS.

T&R in Wireless Sensor Networks

Due to its importance, we are going to analyze the works related to trust and reputation from the point of view of Wireless Sensor Networks (WSN), even though, based on the context, they can be categorized in some cases into Ad-Hoc networks.

TRSs in WSN networks add a great value in constructing the network and making easier the addition and deletion of sensor nodes. They also improve the mechanisms to replace failing or unreliable nodes in a transparent way [62].

Due to the intrinsic features of WSN (dynamism, low computational and communication resources, etc.) [63], the creation, operation and management of this kind of networks are dependent upon the cooperative and trusting nature of its nodes. Thus, the trust establishment between nodes is a desirable requirement.

However, using the traditional tools such as cryptographic processes to generate public key infrastructure and establish trust based on them are not possible in a WSN, due to the resource limitations of sensor nodes [57]. Therefore, TRS are a perfect approach to offer the needed mechanisms and to cope with these resource limitations [64].

The main researches in this field are focused on: the development and evaluation of **algorithms** to calculate trust and reputation, the identification and characterization of **sources of information** (direct and indirect information) to calculate both parameters, and the study of methods to **secure the basic network protocols** (aggregation of sensed values, time synchronization, and routing) [65].

T&R in Online Services

The provision of online services [66] is one of the most prolific fields related to the study and deployment of TRS.

eBay [67], Amazon [17], Booking [68] or AirBnB [69] are good examples of online marketplaces that use reputation mechanisms.

All these models consider reputation as a global property, and use a single value that is not dependent on the context nor on the entity. Thus, we say that they are pure-reputation-based systems, since there is no trust (subjective values) at all.

The information source used to build the reputation value is the information that comes from other entities that previously interacted with the target entity.

They do not often provide explicit mechanisms to deal with users that provide false information. In this context, only redundancy, a great number of opinions about the same subject, is the only way to increase the reliability of the global reputation value.

The main researches in this field are focused on: the development of **reputation calculation algorithms** in order to enable an maximize transactions between the entities belonging to the system, and the **dissemination of trust and reputation information** throughout global scope systems.

T&R in P2P

TRSs in the context of peer-to-peer (P2P) networks are distributed [70], [71]; there is no centralized entity to analyze the behavior of entities in a network, so individual nodes keep track of their peers' behavior and exchange this information directly with others. In this way, these systems are mainly based on the idea of trust and, in some cases in the management of local reputation values [72].

Discipline	Main topics
Philosophy	loyalty, delegation, risk, betrayal, moral boundaries
Psychology	learning process, memory, punishment, trust/distrust
Sociology	subjectivity, prejudices, mechanism to build cooperation
Economy	personal vs. institutionalized trust, maximization, risk, redemption
Organizational Management	proficiency, efficiency, measurability
Corporations	disperse sources of information, processing of abstract information
Personal branding	sources of information, creating and maintaining reputation
Communication&Networking	promoting collaboration, improving performance
Ad-hoc Networks	fast trust-generation mechanisms
WSN	calculation algorithms, sources of information, securing protocols
Online Services	calculation, revocation, enabler of interactions, dissemination
P2P networks	cooperation over risk, massively distributed trust systems
Social Networks	reputation sources, dissemination, virtual vs. real-life

Table 1.1: Main trust and reputation topics of study by field of knowledge

Besides, these systems try to counter selfish behavior of nodes by enforcing nodes to **cooperate** with each other in order to rise their own trust and obtain more benefits from the system.

T&R in Social Networks

Research works are focused on two main topics: the increasing importance of virtual reputation in virtual worlds, and the increasing influence of virtual relationships in real-life reputation.

Related to the first topic, researches have analyzed the sources of virtual trust and reputation in different kinds of social networks, creating user-centered models [73], [74] based on message forwarding, like actions, etc.

Related to the influence of virtual activity in the real-life reputation, Golbeck [75]–[77] proposes a trust concept derived from a sociological point of view where virtual relationships expand their influence and they have to be taken into account outside those virtual environments in the same way than other real-life relationships (family, friends, work colleges, etc.) are considered.

Thus, we can say that **identifying reputation sources** and **analyzing reputation dissemination** are the most studied topics of TRSs from the point of view of Social Networks.

Conclusions

Derived from the previous analysis, we can see how every discipline focuses its studies on a specific set of topics, but none of them tries to take advantage of knowledge generated in other disciplines to improve its behavior or performance. Detailed topics in some fields are completely obviated in others, and even though the study of some topics within several disciplines produces complementary results, these results are not usually used outside the discipline where they were generated. Main topics for each field are compiled in Table 1.1

This lead us to a high dispersion of knowledge and to a lack in the reuse of methodologies, policies and techniques among different fields.

Due to its great importance, this high dispersion of trust and reputation knowledge is one of the main problems this thesis will try to solve.

1.3 Contributions

This Ph.D. Thesis addressed the improvement of the models and methodologies related to TRS in order to allow a more precise, systematic, complete and secure analysis and design of this kind of systems.

Regarding the proposition of novel models and methodologies to improve the general understanding and the definition of TRS, the main contributions of this PhD thesis are:

- The compilation of an extensive literature and knowledge about the goals, utilization, and the characteristics of TRS in different fields of knowledge: philosophy, psychology, sociology, economics, business management, communications and networking, online services, etc.

- The definition of a **generic architecture for TRS**, identifying the entities and processes involved in this kind of systems regardless of the field of knowledge or the specific case of use where we apply them.

- The definition of a **analysis methodology for TRS** that systematically allows to identify all the assets and process involved in this kind of systems. This methodology allows systematizing the process of understanding and predicting the behavior of any TRS independently of the field of knowledge or the specific case of use where we apply them.

- The definition of a **design methodology for TRS**. This methodology describes the steps to systematically select all the components and processes involved in the development and deployment of a TRS in a real-life environment. As a further result of this design methodology a **taxonomy of types of TRS** according to their functional objectives is proposed.

Regarding the proposition of novel frameworks and methodologies to improve the security of TRS, the main contributions of this PhD thesis are:

- The definition of **a generic framework for analyzing security** of any kind of system. Based on this framework all assets and processes prone of being attacked can be identified in a systematic way.

- The definition of a **taxonomy of attacks against TRS**. This taxonomy will allow us to learn and study attacks that had not yet been identified in the literature.

- The definition of a **methodology to systematically analyze vulnerabilities and possible countermeasures of any TRS in real environments**. Therefore, we will be able to make design decisions that minimize the probability of an attack being successfully completed, as well as we can make design decisions to minimize the impact of an attack in those cases where it cannot be completely avoided.

- The development of TRS simulator that allows to analyze the performance of applying a TRS with a specific set of features to any type of environment.

Finally, we have made original contributions to different areas of knowledge thanks to the application of the models and methodologies previously presented. The fields of knowledge addressed and their corresponding contributions are:

- **The detection of thermal anomalies in Data Centers**. Thanks to the application of the TRS analysis and design methodologies, we successfully implemented a Thermal Anomalies Detection System based on a TRS. Its main contribution is the autonomous management of the diverse trust and reputation information available in the data center.

- **The improvement of the performance of a harvesting system based on swarm computing and social odometry**. Through the implementation of a TRS we achieved to improve the ability of coordinating a distributed network of autonomous robots. The main contribution lies in the analysis and validation of the incremental improvements that can be achieved with proper use information that exist in the system and that can be relevant for the TRS, and the implementation of the appropriated trust algorithms based on such information.

- **The improvement of Wireless Mesh Networks security** against attacks against the integrity, confidentiality or availability of data and communications supported by these networks. Thanks to the implementation of a TRS we improved the detection time rate against these kind of attacks and we limited their potential impact over the system.

- **The improvement of Wireless Sensor Networks behavior against advanced attacks**, such as insider attacks, unknown attacks, etc. Through the deployment of a TRS we can implement countermeasures against such attacks in a complex environment.

1.4 Structure

This Ph.D. thesis is organized as follows:

- Chapter 2 presents the basic concepts and the state of the art on Trust Management Systems. It will help us to understand the difficulties derived from these diversity of models and technologies in order to create a knowledge base about TRS.

The rest of the document is divided into two main sections. The first part focuses on presenting the theoretical contributions of this work and it is organized as follows:

- Chapter 3 describes an architecture that copes with the previously described complexity and allows us to identify the main entities and processes related to any kind of TRS, no matter the field of knowledge where it is applied. Besides, a methodology to analyze TRS is presented.

- Chapter 4 presents a methodology to design a TRS in order to attain a specific set of goals. A taxonomy of TRS application patterns is proposed based on the functional areas where TRS are highly effective.

- Chapter 5 presents a generic security framework to analyze attacks against any kind of system. Subsequently, this framework is applied to TRS analyzing the entities and processes identified in Chapter 3. This yield a complete and novel taxonomy of TRS attacks.

The second part focuses on presenting the practical application of the previous models and methodologies to solve problems or improve the performance of real-life scenarios. It is organized as follows:

- In Chapter 6 we apply the models and methodologies to the field of energy consumption in data center. In this scenario, a TRS is designed to detect and isolate thermal anomalies in data centers. This scenario serves as an example of how a TRS can **detect and isolate anomalous behaviors**.

- In Chapter 7 different designs and implementations of a TRS are applied to improve the performance of a swarm of autonomous robots. This scenario serves as an example of how a TRS can be use to **minimize the degradation of a system or maximize its performance**.

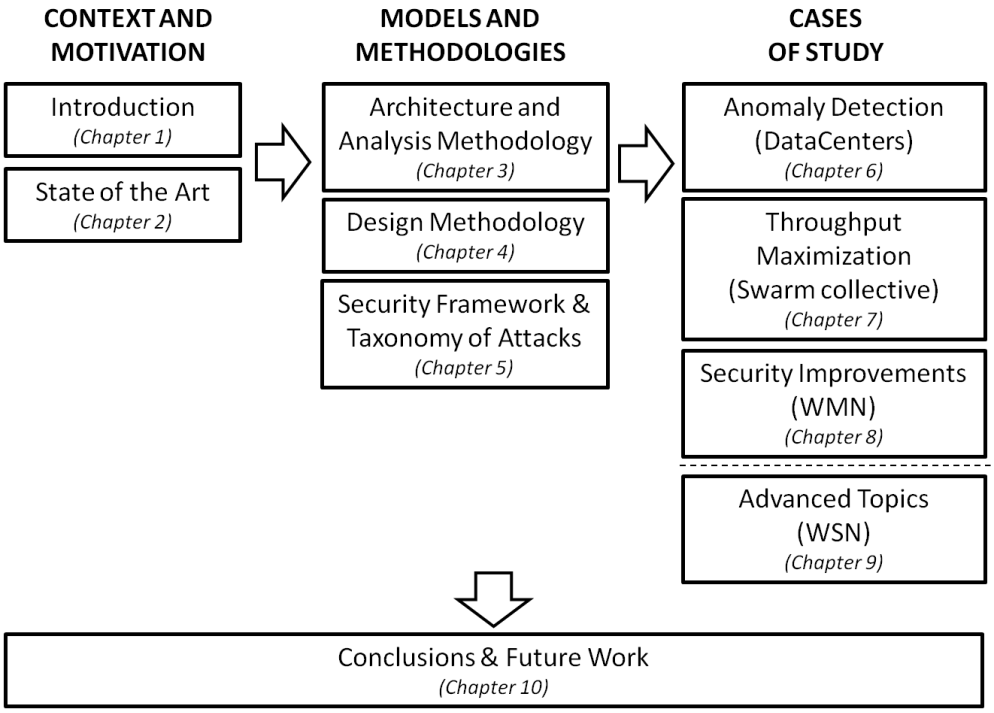


Figure 1.1: Overview of the Ph.D. Thesis structure and chapter organization

- Chapter 8 describes the **improvement of the security of a WMN** through the use of a TRS specially designed to achieve this goal. This scenario serves as an example of how a TRS can be use to minimize the likelihood and the impact of attacks against a distributed and complex system.
- Chapter 9 analyzes in detail some **advanced topics** derived from the use of TRS. In the context of the security of a WSN, we describe the consequences of make some design decisions, and analyze the performance of a TRS under different environment hypothesis and attacks of different strength.

Finally, Chapter 10 summarizes the conclusions derived from the research that is presented in this Ph.D. thesis, as well as the contributions to the state-of-the-art on analyzing, designing, and securing TRSs. The Chapter also includes a summary on future research directions.

Figure 1.1 provides the reader with an overview of the structure of this Ph.D. thesis and how the Chapters are organized.

1.5 Publications

The results of this PhD Thesis, together with other related research have been published in international conferences and journals. In this section we briefly present these publications and highlight the chapter in which the specific contributions can be found.

1.5.1 Journal papers

In terms of scientific publications, this Ph.D. thesis has generated the following articles in international journals:

- D. Fraga, Á. Gutiérrez, J. C. Vallejo, *et al.*, "Improving social odometry robot networks with distributed reputation systems for collaborative purposes", *Sensors*, pp. 11 372–11 389, 2011 [JCR Q1 IF=1.870] *(Chapter 4 and Chapter 7 of this Ph.D. Thesis)*

- Z. Bankovic, D. Fraga, J. M. Moya, *et al.*, "Improving security in wmns with reputation systems and self-organizing maps", *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 455 –463, 2011, Efficient and Robust Security and Services of Wireless Mesh Networks, ISSN: 1084-8045 [JCR Q2 IF=0.660] *(Chapter 4 and Chapter 8 of this Ph.D. Thesis)*

- Z. Banković, J. M. Moya, D. Fraga, *et al.*, "Distributed intrusion detection system for wireless sensor networks based on a reputation system coupled with kernel self-organizing maps", *Integr. Comput.-Aided Eng.*, vol. 17, pp. 87–102, 2 2010, ISSN: 1069-2509 [JCR Q2 IF=2.042] *(Chapter 9 of this Ph.D. Thesis)*

- M. Zapater, D. Fraga, P. Malagón, *et al.*, "Self-organizing maps versus growing neural gas in detecting anomalies in data centres", *Logic Journal of the IGPL*, vol. 23, no. 3, pp. 495–505, 2015 [JCR Q3 IF=0.458]

- Z. Banković, D. Fraga, J. M. Moya, *et al.*, "Bio-inspired enhancement of reputation systems for intelligent environments", *Inf. Sci.*, vol. 222, pp. 99–112, Feb. 2013, ISSN: 0020-0255 [JCR Q4 IF=0.205]

- Z. Banković, J. C. Vallejo, D. Fraga, *et al.*, "Detecting false testimonies in reputation systems using self-organizing maps", *Logic Journal of the IGPL*, vol. 21, no. 4, pp. 549–559, 2013 [JCR Q3 IF=0.458]

- Z. Bankovic, D. F. Aydillo, J. M. M. Fernández, *et al.*, "Detecting unknown attacks in wireless sensor networks that contain mobile nodes", *Sensors*, vol. 12, no. 8, pp. 10 834–10 850, 2012 [JCR Q1 IF=1.870]

- J. Moya, Á. Araujo, Z. Banković, *et al.*, "Improving security for scada sensor networks with reputation systems and self-organizing maps", *Sensors*, vol. 9, no. 11, p. 9380, 2009 [JCR Q1 IF=1.870]

- J. M. Moya, J. C. Vallejo, D. Fraga, *et al.*, "Using reputation systems and non-deterministic routing to secure wireless sensor networks", *Sensors*, vol. 9, no. 5, p. 3958, 2009, ISSN: 1424-8220 [JCR Q1 IF=1.870]

1.5.2 Conference papers

Also, this Ph.D. thesis has generated the following articles in international peer-reviewed conferences:

- D. Fraga, Z. Bankovic, and J. M. Moya, "A taxonomy of trust and reputation system attacks", in *11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2012, Liverpool, United Kingdom, June 25-27, 2012*, 2012, pp. 41–50 [Core A conference] *(Chapter 5 of this Ph.D. Thesis)*

- Z. Bankovic, J. Moya, D. Fraga, *et al.*, "Holistic solution for confining insider attacks in wireless sensor networks using reputation systems coupled with clustering techniques", in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, 2011, pp. 61–72 [Core A conference] *(Chapter 9 of this Ph.D. Thesis)*

- Z. Bankovic, D. Fraga, J. C. Vallejo, *et al.*, "Improving reputation systems for wireless sensor networks using genetic algorithms", in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '11, Dublin, Ireland: ACM, 2011, pp. 1643–1650, ISBN: 978-1-4503-0557-0 [Core A conference]

- Z. Bankovic, D. Fraga, J. C. Vallejo, *et al.*, "Self-organizing maps versus growing neural gas in detecting data outliers for security applications", in *Hybrid Artificial Intelligent Systems - 7th International Conference, HAIS 2012, Salamanca, Spain, March 28-30th, 2012. Proceedings, Part II*, 2012, pp. 89–96

- Z. Bankovic, J. C. Vallejo, D. Fraga, *et al.*, "Detecting bad-mouthing attacks on reputation systems using self-organizing maps", in *Computational Intelligence in Security for Information Systems - 4th International Conference, CISIS 2011, Held at IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011. Proceedings*, Á. Herrero and E. Corchado, Eds., ser. Lecture Notes in Computer Science, vol. 6694, Springer, 2011, pp. 9–16, ISBN: 978-3-642-21322-9

- Z. Banković, J. M. Moya, D. Fraga, *et al.*, "Detecting unknown attacks in wireless sensor networks using clustering techniques", in *Proceedings of the 6th International Conference on Hybrid Artificial Intelligent Systems - Volume Part I*, ser. HAIS'11, Wroclaw, Poland: Springer-Verlag, 2011, pp. 214–221, ISBN: 978-3-642-21218-5

- Z. Bankovic, D. Fraga, J. M. Moya, *et al.*, "Detecting and confining sybil attack in wireless sensor networks based on reputation systems coupled with self-organizing maps", in *Artificial Intelligence Applications and Innovations - 6th IFIP WG 12.5 International Conference, AIAI 2010, Larnaca, Cyprus, October 6-7, 2010. Proceedings*, 2010, pp. 311–318 on Reputation Systems Coupled with Self-organizing Maps

1.5.3 Other publications

Finally, the author has also contributed in the following articles in international peer-reviewed conferences and journals, not specifically related to the contents of this Ph.D. Thesis:

- Z. Bankovic, J. M. Moya, E. Romero, *et al.*, "Using clustering techniques for intelligent camera-based user interfaces", *Logic Journal of the IGPL*, vol. 20, no. 3, pp. 589–597, 2012 [JCR Q3 IF=0.458]

- P. Arroba, D. Fraga, J. C. Vallejo, *et al.*, "A methodology for developing accessible mobile platforms over leading devices for visually impaired people", in *Ambient Assisted Living - Third International Workshop, IWAAL 2011, Held at IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011. Proceedings*, 2011, pp. 209–215

- Z. Banković, E. Romero, J. Blesa, *et al.*, "Using self-organizing maps for intelligent camera-based user interfaces", in *Hybrid Artificial Intelligence Systems, 5th International Conference, HAIS 2010, San Sebastián, Spain, June 23-25, 2010. Proceedings, Part II*, 2010, pp. 486–492

- E. Romero, Á. Araujo, J. M. Moya, *et al.*, "Image processing based services for ambient assistant scenarios", in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, 10th International Work-Conference on Artificial Neural Networks, IWANN 2009 Workshops, Salamanca, Spain, June 10-12, 2009. Proceedings, Part II*, 2009, pp. 800–807 [JCR Q4 IF=0.402]

- Á. Araujo, D. Fraga, J. M. Fernandez, *et al.*, "Domotic platform based on multipurpose wireless technology with distributed processing capabilities", in *Proceedings of the IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2004, 5-8 September 2004, Barcelona, Spain*, 2004, pp. 3003–3007

1.6 Research Projects

During the development of the Ph.D. Thesis the author has participated in the following R&D projects and industrial contracts:

- LPCloud project: This project focuses on the optimum management of low-power modes for cloud computing. Funded by the National Program for Public-Private Cooperation, INNPACTO (MINECO) of the Spanish Ministry of Economy and Competitiveness. [September 2013]

This work was partially supported by:

- DGUI de la Comunidad Autónoma de Madrid and Universidad Politécnica de Madrid under Research Grant CCG07-UPM/TIC-1742
- Campus of International Excellence (CEI) of Moncloa, under Research Grant of the Program for Attracting Talent (PICATA).
- P8/08 within the National Plan for Scientific Research, Development and Technological Innovation 2008-2011
- The Spanish Ministry of Industry, Tourism and Trade, under Research Grant TSI-020301-2009-18 (eCID).
- The Spanish Ministry of Science and Innovation, under Research Grant AMILCAR TEC2009-14595-C02-01, and the CENIT Project Segur@.
- The N4C.Networking for Challenged Communications Citizens: Innovative Alliances and Test beds project, funded by the Seventh Framework Program (FP7-ICT-223994-N4C) of the European Commission.

2. Related work

2.1 Introduction

In this chapter we will present the concept of Trust Management System (TMS). Basically, a TMS is specific approach for dealing with the ideas of trust of reputation. They define the topics and elements that are essential for their proposed models, the involved processes to manage trust and reputation, etc. Each TMS can specify different architectural components and processes based on its priorities or its field of application. Therefore, we can find a number of different TMS for a specific field. As we have already described in Chapter 1, trust and reputation have been studied in a wide range of fields of knowledge. Therefore, this multiplicative factor (*i.e.*,many TMS for fields of knowledge multiplied by many field of knowledge working with trust and reputation) yields to the existence of a a huge number of TMS described in the literature.

The knowledge derived from this compilation of TMS analysis, will lead us to propose a meta-model or generic architecture for TRS in Chapter 3. This architecture will allow us to express in its terms all the TMS in the literature. Furthermore, it will allow us to describe and analyze any kind of system dealing with trust and reputation, even if it is not based on a previously known TMS.

In the Section 2.2 we present the main concepts regarding Trust Management Systems. Section 2.3 provides a comprehensive analysis of TMSs found in the literature. Finally, in Section 2.4 we draw some conclusions about the diversity and complexity of the existing TMS.

2.2 Trust Management Systems

Any framework to define trust and reputation dynamics are known as Trust Management Systems in the literature. Historically, Trust Management appears as a special case of Risk Management [98], an area of knowledge deeply studied specially in Business and Organization Management. Traditionally, a Trust Management System includes the definition of processes such as: trust establishment, trust update, and trust revocation [56], [99].

They can be classified based on three main dimensions or features:

- **Policy-based vs. reputation-based**: there are two main approaches to evaluate trust in the literature, namely: policy-based trust management and reputation-based trust management [100]–[103].

  Policy-based trust management is based on strong and objective security schemes such as cryptographic processes, logical rules, signed credentials, etc. This policy-based trust management approach usually makes a binary decision according to which the requester is trusted or not. They are usually oriented to grant access to requested resources.

  On the other hand, reputation-based trust management uses numerical and computational mechanisms to evaluate trust. Typically, trust is calculated by collecting, aggregating, processing, and disseminating trust and reputation throughout the system.

- **Evidence-based vs. monitoring-based**: based on Li and Singhal [104] there are two main approaches to calculate trust regarding the sources of information they use. Evidence-

based trust management is based on knowledge that can unambiguously prove trust among nodes: public keys, identity, challenge processes.

Monitoring-based trust management is based or rating the trust level of each entity based on behavioral information. This information can be obtained by direct observation or communicated by other entities within the system.

This classification can be called as **Certificate-based vs. behavior-based** in the literature [105].

- **Positive, negative, and mixed TMS**: Adams [106] proposes these three types of TMS. Positive reputation systems only consider observations of the positive behaviors of an entity, and negative reputation only take into account observations of the negative behaviors of and entity. This classification is more useful when there is a default state of trust (trusted/untrusted) and the current trust values are calculated in a negative way (based on the observed behaviors in opposition to this default state).

As we described in Section 1.2, without loss of generality, we will focus our analysis in *reputation-based*, *monitoring-based*, and *mixed* Trust Management Systems. Therefore, in the next sections some TMS belonging to these classification are described in detail.

2.3 State of the Art

2.3.1 Marsh

The work of Marsh [107] is said to be the first work on trust in computer science. Marsh concentrates on modeling trust between only two agents. Thus, the trust management does not treat the collection of recommendations provided by other entities.

This model identifies knowledge, utility, importance, risk, and perceived competence as important aspects related to trust.

The model defines three types of trust: dispositional trust, the trust of an entity independent from the possible cooperation partner and the situation; general trust, the trust of an entity in another one, independently of the specific situation; and situational trust, which describes the trust of an entity in another one in a specific situation or context.

2.3.2 Fortune's Most Admired Companies List

The Fortune's Most Admired Companies List (MAC List) [108] surveys CEOs and financial analysts about their view of listed companies in terms of issues such as financial soundness, innovation, use of corporate assets and social responsibility.

The list is developed by the Fortune's editorial panel in discussion with business leaders and financial analysts and try to identify features that executives and financial experts admire in companies.

2.3.3 Castelfranchi and Falcone

The model proposed by Castelfranchi and Falcone [109] was a forefather of cognitive trust models, and it is the base of later models in literature.

They claim that trust is the mental background of delegation. Thus, trust becomes a mental state that yields one entity to delegate a task to other. This concept of trust is based on a number of basic beliefs: competence belief (the other entity can actually do the task), dependence belief (the other entity is necessary or a better choice to perform the task), willingness belief (the other entity is supposed to be willing to do the task), and persistence belief (the other entity is stable on its intentions of performing the task).

2.3.4 Sporas

In this model [110], only the most recent rating between two entities is considered. Besides, entities with very high reputation values experience much smaller reputation changes after each update than entities with a low reputation.

Sporas incorporates a measure of the reliability of the entities' reputation based on the standard deviation of reputation values. It is robust to changes in the behavior of an entity and the reliability measure improves the usability of the reputation value.

2.3.5 Histos

Histos [110] was designed as a response to the lack of personalization that **Sporas** reputation values have. The model can deal with direct information and witness information. In this case, the reputation value is a subjective property assigned particularly by each individual (actually becoming a trust value).

The treatment of direct interaction in this reputation model is limited to the use of the most recent experience with the agent that is being evaluated.

The strength of the model relies on its use of witness information. Ratings are represented as a directed graph. The reputation of an agent at level n of the graph (with $n > 0$) is calculated recursively as a weighted mean of the rating values that entities in level $X - 1$ gave to that entity.

A drawback of this model is the use of the reputation value assigned to a witness also as a measure of its reliability.

2.3.6 Abdul-Rahman and Hailes

The trust model presented by Abdul-Rahman and Hailes [111] is focused on virtual communities related to e-commerce and artificial autonomous agents.

The model defines *direct trust* and *recommender trust*. Direct trust is the trust of an entity in another one based on direct experience, whereas recommender trust is the trust of an entity in the ability of providing good recommendations.

Trust can only have discrete labeled values, namely Very Trustworthy, Trustworthy, Untrustworthy, and, Very Untrustworthy for direct trust, and Very good, good, bad and, very bad for recommender trust.

The difference between two ratings from different entities can be computed as semantic distance. This semantic distance can be used to adjust further recommendations. The combination of ratings is done as a weighted sum, where the weights depend on the recommender trust.

2.3.7 Schillo et al.

This trust model [112] is oriented to scenarios where the result of an interaction between two entities is good or bad. This value is a subjective property assigned particularly by each individual and it does not depend on the context.

It is based on Prisoner's dilemma set of games [42] with a partner selection phase.

Each agent receives the results of the game it has played plus the information about the games played by a subset of all players (its neighbors). The model is based on probability theory that uses the number of times that the target entity was honest.

Besides, an entity can get information from other agents that it has met before. The answer of witnesses to a query is the set of observed experiences, and not a summary of them.

The model assumes that witnesses never lie but that can hide (positive) information in order to make other agents appear less trustworthy.

2.3.8 Yu and Singh

The TRS model proposed by Yu and Singh [113] uses two information sources.

The first one contains the entity's belief built as a result of its direct interaction with other entities. The second one includes the testimonies of third-parties that can be beneficial in the absence of local ratings.

The model propose a trust network which tries to locate the most appropriate witnesses in a multi-agent system. When a requesting entity wants to evaluate the trustworthiness of other entity , it sends a query to the neighbors of that entity asking for their perception regarding the target entity.

This model deals with malicious entities who deliberately disseminate misinformation through network.

2.3.9 REGRET

REGRET [114] is a decentralized TRS designed for complex e-commerce environments where various types of entities with different social relationships play important roles. It describes the social structure and relationships of the system through the ideas of cooperation, competition, and trade.

REGRET is based on a three-dimensional reputation model: Individual dimension or subjective reputation which calculates trust based on the direct impressions of an entity; social dimension, which is divided into three types of reputation: witness reputation, neighborhood reputation, and system reputation; and ontological dimension, which adds the possibility of combining different aspects of reputation to calculate a complex one. With the help of the ontological structure, each entity is capable of determining the overall reputation of a particular entity by assigning the appropriate influence degree to each aspect related with its demand.

In addition to the reputation value, REGRET gives a reliability measurement which reflects the confidence level of the produced reputation value.

2.3.10 Aberer and Despotovic

The model proposed by Aberer and Despotovic [115] is one of the first TMS focused on P2P networks.

It is based on the complaints a peer receives from other peers in the network. Although it improves network performance in stable environments, due to the naive of its approach, it is highly sensitive to malicious peers. However, it served as baseline to subsequent models in this area of application.

2.3.11 Esfandiary and Chandrasekharan

The model proposed by Esfandiary and Chandrasekharan [116] uses to sources of information: observation and interaction. The processing of observed information is based on Bayesian learning. The interaction is based on two main protocols: an exploratory protocol and a query protocol.

In the exploratory protocol, entities ask the other entities about known topics to evaluate their degree of trust. Answers consistent with their knowledge yield to consider an entity as trusted. In the query protocol, entities ask for advice to previously trusted entities.

The authors claim that the calculation of this trust interval is equivalent to the problem of routing in a communication network and, therefore, known distributed algorithms used to solve that problem can be successfully applied to this situation.

2.3.12 Afras

The main characteristic of this model [117] is the use of fuzzy sets to represent reputation values. Once a new fuzzy set that shows the degree of satisfaction of the latest interaction with a given entity is calculated, the old reputation value and the new satisfaction value are aggregated using a weighted aggregation. Besides, the weights of this aggregation are calculated from a single value that they call remembrance or memory.

Recommendations from other entities are aggregated directly with the direct experiences. If they come from a recommender with a high reputation, they have the same degree of reliability as a direct experience.

2.3.13 Azzedin and Maheswaran

Azzedin and Maheswaran [118] propose a TMS based on a combination of direct trust and reputation by weighting the two components differently.

It gives more weight to the direct trust. This direct trust o trust level is calculated based on past experiences and is given for a specific context.

Calculation of reputation values is based on a neural network approach.

2.3.14 Carter et al.

Carter et al. propose a complex but novel TMS [119] based on the concept of roles. They claim that the reputation of an agent is based on the degree of fulfillment of roles ascribed to it by the society. Therefore, if society judges that an entity has met its roles, it will be rated with a positive reputation.

They define five main roles: social information provider, interactivity role, content provider, administrative feedback, and longevity role. All of them oriented to promote a information-sharing society.

Finally, the entity's overall reputation is calculated as a weighted aggregation of the degree of fulfillment of each role. These weights are dependent on the specific society, and the society has a centralized mechanism that calculate and disseminate these reputation values, and monitors the society.

2.3.15 SECURE

The trust model and trust management in the SECURE project [120] aims to transfer a human notion of trust to ubiquitous computing.

The main aspect of the trust model is the distinction between unknown and untrustworthy. An entity b is unknown to an entity a, if a cannot collect any information about b. Whereas b is untrusted if a has information, based on direct interaction or recommendations, stating that b is an untrustworthy entity.

The trust propagation is based on policies. These policies allow entities to explicitly express whose recommendations are considered in a trust decision. And finally, the decision making is threshold based.

2.3.16 Wang and Vassileva

Wang and Vassileva [121] proposed a trust model using Bayesian networks based on the quality of services provided by entities.

The entities manage two different values of trust in another entities: competence in providing services, and reliability in providing recommendations about other entities.

The model uses binary events to qualify transactions (successful or unsuccessful transactions) between entities. Trust is modelled based on this transactions and it used to weight the direct and indirect information: the entity will discard the recommendations from the untrustworthy sources but will combine the recommendations from the trustworthy and unknown sources.

2.3.17 XenoTrust

XenoTrust is a TMS proposed by Dragovic [122]. It describes a novel approach to TMS because it is an event-based distributed trust management system. This event-based paradigm allows to reduce communication overheads and can simplify, and even enable, the use of TMS in systems with tight communication limitations. XenoTrust uses some performance criteria

(*i.e.,*reliability, honesty and throughput) to calculate the trust values assigned to other entities in the system.

2.3.18 Shand et al.

Shand et al. propose a TMS to facilitate secure collaboration in pervasive computer systems [123]. This is one of the first TMS that tries to overcome the performance of the policy-based trust models.

When applying policy-based TMS to very dynamic systems the policies are too strict to efficiently handle topology changing networks, nodes entering and exiting from the system, etc.

This model is based on the existence of some generic-policies and some local or node-specific policies that are combined in order to calculate trust values.

2.3.19 Reputation Quotient

The Reputation Quotient [48], [49] tries to obtain data on a company's reputation from the point of view of the general public, customers, employees, suppliers and investors. The model measures perceptions of an organization in terms of social expectations of dimensions such as products and services, vision and leadership, work place environment and social responsibility.

2.3.20 FIRE

In the FIRE model [124], trust is evaluated based on a different number of information sources: Interaction Trust (IT), that is built from the self experience of an entity with the other entities; Witness Reputation (WR) that is based on the direct observation of an entity's behavior by some third-party agent; Certified Reputation (CR), one of the novelties in the FIRE model, that consists of certified references disclosed by third-party agents; and Role-based Trust (RT), which models the trust across predefined role-based relationships between two entities.

The significance of each component in the trust calculation algorithm is adjusted according to changes in the environment.

Each component owns a trust algorithm with relevant rating weight function to determine the quality of ratings tailored to its responsibility. Thus, the weight algorithm for IT in based on the age of ratings whereas WR and CR have to take the credibility of rating into account as well. Credibility is based on a filtering mechanism that identifies inaccurate reports and penalizes misbehaving entities.

2.3.21 PeerTrust

PeerTrust is a trust model [125] with specific characteristics for peer-to-peer e-commerce communities.

It uses several factors to calculate the reputation values of the entities (peers): feedback which is a judgment of other peers regarding target peer; feedback scope, such as the amount of transactions the peer experienced with others; a credibility factor to evaluate the honesty of feedback sources; transaction context factor such as time and size of transactions; and community context factor.

This model proposes an innovative composite trust metric that incorporates the described parameters to enhance accuracy and reliability of predicted trustworthiness.

2.3.22 Corporate Personality Scale

The Corporate Personality Scale [51] surveys customers and employees in terms of their perceptions of organization's personality, focusing on dimensions such as agreeableness, competence and enterprise.

2.3.23 SPIRIT

The SPIRIT model [126] can be applied to survey Corporate Reputation from the perspective of customers, employees, suppliers, investors and community groups. It measures Corporate Reputation in terms of the experience, feelings and intentions of stakeholders towards a business.

2.3.24 TIBFIT

The model was proposed by Krasniewski and Varadharajan [127]. TIBFIT is a trust scheme implemented in the form of a communication protocol. It is designed to detect node failures in event-driven WSN. This detection is based on the analysis of binary reports from nodes close to any event in the system. If TIBFIT detects a node failure, it masks any communication related to this node. Additionally it can communicate this failure. Therefore, the system as a whole can try to take actions to deal with this situation.

2.3.25 UniTEC

UniTEC is the TMS proposed by Kinateder at al. [103]. This model is focused on experience as base of trust values. It uses direct and indirect information and performs direct and indirect trust values updates separately.

UniTEC gives more weight to the recent experience than to the old one, and previously calculated trust values are expressed as a binary metric (good or bad experiences). Thus, as a side-effect, the storage of old experience requires less resources than the new ones.

2.3.26 TRAVOS

The TRAVOS (Trust and Reputation model for Agent-based Virtual Organizations) system [128] is developed to ensure high-quality interaction between the entities of a large open system.

It uses two information sources to calculate the reputation of the entities: Direct Interaction and Witness Observation. However, this model relies greatly on its direct experiences and refuses to combine others' opinions unless they are really required.

For this purpose, it provides a confidence metric to determine whether the direct experiences are sufficient to make an acceptable review to a particular entity or not. If not, it disseminates queries to obtain additional observations from other witnesses who claim to have had previous interaction with that certain entity.

2.3.27 Crosby and Pissinou

Crosby and Pissinou proposed a mechanism for the election of cluster heads in WSN based on a distributed trust-base framework [129].

It is based on the use of direct and indirect information coming from previously trusted nodes. Trust is modelled using a feature extraction and weighting mechanism of some essential parameters from the communication protocol: packet drop rate, data packets and control packets. Each node stores a local trust table for all its neighbor nodes. Cluster nodes can ask for these tables. Therefore, they can update their reputation values over other cluster head nodes to improve their routing path policies.

2.3.28 BambooTrust

Proposed by Kotsovinos [130], BabooTrust is based on XenoTrust. It is focused on global public computing platforms such as grid computing systems and it is built as a P2P system.

It is a model with a high-performance regarding the distribution of trust information throughout the system.

It implements as Bamboo hash table in order to facilitate the performance, scalability, efficiency, and load-balancing of the whole system.

2.3.29 TidalTrust

This trust model proposed by Golbeck [131] is based on ten discrete trust values in the interval $[1, 10]$.

This model is based on the idea that humans are better in rating on a discrete scale than on a continuous one and the 10 discrete trust values should be enough to approximate continuous trust values. Besides, recursive trust or rating propagation allows to infer the rating of subjects by the ratings provided by other entities. Since each entity aggregates its collected ratings and passes only a single value to its ancestor in the recursion, the source cannot evaluate which nodes provided their rating.

2.3.30 Bayesian Reputation System

Bayesian Reputation System (BRS) was proposed by Jøsang et al [132]. It supports both binomial and multinomial rating models to allow rating supply happening in different levels. Mathematically, multinomial BRS is based on computing reputation scores by statistically updating the Dirichlet Probability Density Function (PDF) [133], [134].

In this context, entities are allowed to rate other entities within any level from a set of predefined ratings levels. In contrast, in binomial BRS which is based on Beta Distribution, the agents can only provide binary ratings for the others.

Both systems use the same principle to compute the expected reputation scores: combining previous interaction records with new ratings. Besides, in order to deal with dynamism in the participant's behavior, BRS provides a longevity factor which determines the expiry time of the old ratings and gives greater weight to more recent ones.

2.3.31 Reputation-based Framework for High Integrity Sensor Networks

RFSN was proposed by Ganeriwal and Srivastava [135]. It classifies the actions as cooperative and non-cooperative. It uses direct and indirect information, and the behavior of the node is decided upon a global threshold. If the trust value is below this threshold the node in considered a non-cooperative node, and any contact from the rest of the network nodes is avoided.

The network propagates only positive reputation information in order to avoid some WSN specific attacks such as bad-mouthing attacks. Finally, an aging factor in introduced to give more weight to recent interactions.

2.3.32 Distributed Reputation-based Beacon Trust System

This model was presented by Srinivasan and Teitelbaum [136] It is focus on keep the network performance through detecting malicious beacon nodes.

Each beacon node monitors and provides information about malfunction behaviors of beacons that are one hope from them. Therefore, nodes can choose to trust in a specific beacon based on this information. They use a voting approach to calculate this trust value. The voting process is based on the reputation tables of each node, that are generated by processing the reputation tables of the close beacon nodes.

2.3.33 Subjective Logic

This trust model presented by Jøsang [137] combines elements of Bayesian probability theory with belief theory.

Besides, related to belief theory, trust is represented by opinions which can be used to express the subjective probability that an entity will behave as expected in the next interaction.

Trust Management System	Main contributions
Marsh	Trust based on knowledge, utility, importance, risk, and perceived competence
	Types of trust: dispositional, general, situational
MAC List	Output performance as source of reputation
	Global dissemination process
Castelfranchi and Falcone	Cognitive trust calculation process. Trust based on beliefs
Sporas	Short-term trust
	Hysteresis Trust
Histos	Direct and witness information
Abdul-Rahman and Hailes	Direct trust and recommender trust
	Trust as discrete value
	Aggregated trust values
Schillo et al.	Bipolar trust
	Trustworthiness based. Independent on the context
Model by Yu and Singh	Direct and on-demand witness information.
	Countermeasures against misinformation attacks
REGRET	Complex trust sources: direct, witness, reasoning
	Introduction of mixed trust and reputation systems
Aberer and Despotovic	Trust in P2P networks. Negative trust model
Esfandiary and Chandrasekharan	Direct and indirect information. Q&A challenges
	Trust calculation based on distributed routing algorithms
Afras	Trust formulation: fuzzy values
	Direct observation plus communicated trust
	Quantification of memory
Azzedin and Maheswaran	Reputation as source of trust information. Context based
Carter et al.	Categorization: role-based reputation. Roles are society dependent.
	Calculation based on society principles.
SECURE	Concept of unknown' ' and untrustwhortiness
	Calculation and dissemination based on filters, and thresholds
Wang and Vassileva	Explicit difference between confidence for actions and reliability for communication capabilities
XenoTrust	Performance (reliability, honesty and throughput) as source of trust
	Event-based model
Shand et al.	Improved policy-based system

Table 2.1: Main contributions of TMS in the literature - I

In belief theory as introduced in an opinion can be expressed as a triple (b, d, u), where b represents the belief, d the disbelief, and u the uncertainty about a certain statement.

Finally, it defines operators for combining and recommending opinions.

2.4 Conclusions

In this chapter we have detailed the main concepts regarding trust and reputation as they are described in the literature. Besides, the Trust Management Systems have been presented as one of the common approaches when dealing with the ideas of trust of reputation.

There are a high diversity of TMS proposal in the literature. We can find the main contributions of each model to the field of TMS in Table 2.1 and Table 2.2.

However, each one is focused on specific problems or specific architectures. Therefore, this diversity and specificity prevents them from being used in different environments to those they were designed to.

Even though the lack of generality of each model, the compilation of such information

Trust Management System	Main contributions
Reputation Quotient	Ratings from internal members of an organization as sources of reputation
FIRE	Direct, witness, certified reputation, role-based trust.
	Introduction of categorization
	Trust algorithm as weight function
PeerTrust	Feedback based: information, scope, credibility, transaction, and community
Corporate Personality Scale	Combination of internal and external opinions as source of reputation
SPIRIT	Experiences and feelings as source of organizational reputation
TIBFIT	Trust integrated in a communication protocol
	Auto-filtering of malfunctioning nodes
UniTEC	Focused on experience. Simplification of memories
TRAVOS	Focused on subjective trust. Witness as fine tuning
	Minimum amount of information required to calculate valid trust values
Crosby and Pissinou	Introduction of local trust-tables
	Feature extraction from low level features of the underlying system
BambooTrust	Highly efficient and scalable trust dissemination process
Tidal Trust	Trust as discrete values. Trust transitivity based on weighting
Bayesian Reputation System	Dirichlet Probability Density and Beta Distribution
	Longevity factor.
RFSN	Direct and indirect information. Global threshold
	Only positive information is propagated
DRBTS	Local reputation based on voting over second-hand information
Subjective Logic	Combination of belief theory and Bayesian Reputation systems

Table 2.2: Main contributions of TMS in the literature - II

about the state-of-the-art in TMS gives us a wide perspective of the common elements of these models, their special features, their advantages and disadvantages, the processes involved in the different dynamics, etc. This knowledge will enable us to define a generic architecture for TRS.

Part I

Models and Methodologies

3. Architecture and Methodology to Analyze TRS

3.1 Introduction

After identifying the fields of application of TRS, the state of the art in the literature, and the main challenges this discipline faces, we are now in a position to start to present the set of methodologies developed in this Ph.D. Thesis.

As in many other areas of knowledge, the development of a specific field can be described based on the tools it has to cope with the description, the prediction, and the control' of all the elements and dynamics related to it. Therefore, a field of knowledge with tools that enable users to control its dynamics is much more evolved than a field of knowledge that only counts with tools to describe its dynamics.

If we apply this simple reasoning to the trust and reputation discipline, we can see that we don't even have tools to describe this kind of systems in a complete and systematic way.

The main goal of this chapter is to provide this basic but fundamental tool (a methodology) to identify and describe the architecture and the dynamics (components and processes) related to any kind of TRS, and to predict its behavior.

Despite its concision, this chapter is essential for understanding the contributions and implications of this Ph.D. Thesis.

In the Section 3.2 we present the proposed TRS architecture, describing both its origin and its main components and processes. Section 3.3 describes the methodology to analyze TRS based on the architecture. Finally, in Section 3.4 we draw some conclusions about the application spectrum and the limitations of the proposed methodology.

3.2 Proposed TRS Architecture

In the Chapter 2 we have presented a number of Trust Management Systems which cover a wide range of fields of application, technologies, and even architectures. However, this diversity and specificity prevents them from being used in different environments to those they were designed to.

A model with such features cannot be used as a framework to describe and analyze a generic TRS (independently of its field of knowledge, technologies used, etc.). However, the compilation of such information about TMS has given us a wide perspective of the common elements of these models, their special features, the processes involved in the different dynamics, etc. This global knowledge enables us to define a generic architecture, or meta-model, for TRS.

This generic TRS architecture will allow us to identify and analyze any kind of TRS and stands as one of the main contribution of this Ph.D. Thesis.

Our main goal when defining the architecture was to keep it as simpler as possible without loss of generality. As the result of this process we will present an architecture with only four components and five processes. However, all the previously presented TMS can be expressed in terms of this architecture.

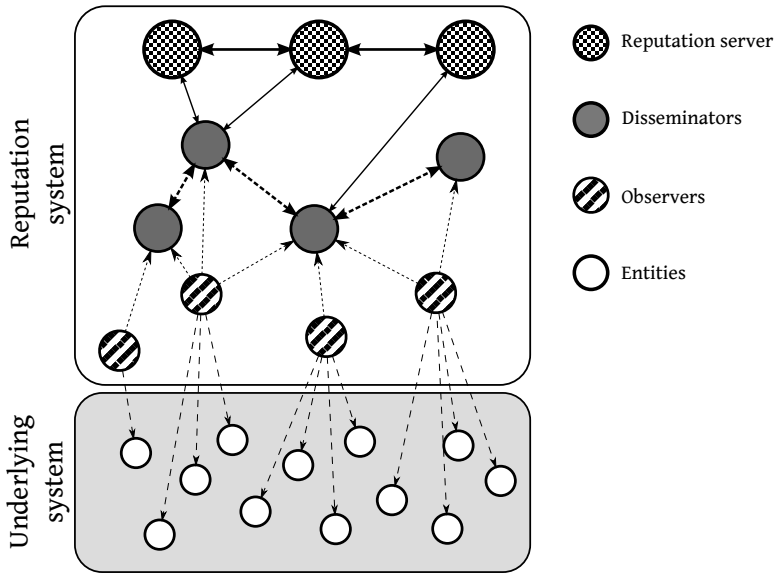


Figure 3.1: Generic TRS architecture components

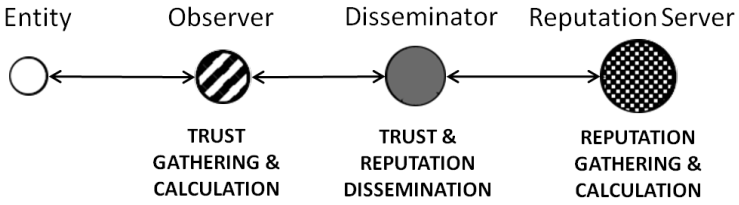


Figure 3.2: Generic TRS architecture processes

3.2.1 Architectural Components

- **Underlying System**. Trust and Reputation Systems exist to improve the performance of another system in a specific way. This system is called *underlying system*, and its basic components are called *entities*.
- **Observers**. They are the basic agents of the TRS. They create and manage values of trust for the entities.
- **Disseminators**. The trust values calculated by the *observers* can be used by other *observers* or can be used to calculate reputation values. In order to allow this transmission of information some agents in the TRS can have the capacity of relaying both trust and reputation information messages.
- **Reputation Servers**. Some agents in the TRS can use the trust information generated and distributed by the *observers* and *disseminators* to generate values of *reputation* for all the *entities*. As we said before, *reputation* is a global and objective concept in opposition to *trust*, that is a subjective and local concept.

3.2.2 Processes Involved

- **Trust Information Acquisition**. In order to create a useful value of trust for *entities*, *observers* can use any of these sources of information: they can use their *perception* to obtain information by direct observation of the *real world*; they can use their *memory*, so

they are able to evaluate the historical behavior of the entities; they can use information provided by other *observers* (*disseminated trust information*); they can use *categorization* as trust source information when the group the entities belong to is associated to a specific trust environment; and finally they can use the global *reputation* value of the entities (this is common in early interactions or when the global perception of an entity is more important than the local perception).

- **Trust Calculation Algorithm**. In order to create a useful value of trust, *observers* process all or some of the aforementioned sources of information with an internal algorithm. This is a key element in the whole reputation system so it has to be analyzed and designed very carefully, as we will see in the next section. As we have mentioned before, it is important to remark that trust should be a concept associated to an *entity* performing a specific *service*. It is not associated to an *entity* as a whole.
- **Dissemination Protocol**. The transmission of trust and reputation information carried out by the *disseminators* is based on the existence of a specific communication protocol that is commonly called *dissemination protocol*.
- **Reputation Information Acquisition**. In order to create a useful value of reputation for *entities*, *reputation servers* can use any of these sources: *trust*, previously calculated *reputation values*, and *external* reputation values, among others.
- **Reputation Calculation Algorithm**. In order to calculate valid reputation values, *reputation servers* use an internal algorithm similar to the *Trust Calculation Algorithm* but utilizing different sources of information, and with the goal of generating an global and objective concept (in opposition to *trust*, that is local and subjective).

3.3 Methodology

As we described before, the main goal of this chapter is to provide a methodology to identify and describe the architecture and the dynamics (components and processes) related to any kind of TRS, and predict its behavior.

The methodology is actually simple and straightforward. It is based on:

- Checking the presence of every component and process of the proposed architecture in the analyzed system.
- Identifying the main features of those components and processes based on the list of features provided in the next sections.
- Analyzing the consequences of all features in the behavior of the individual components and processes, and in the system as a whole.

Therefore, the methodology is essentially a checklist of items and implications that covers possible alternatives for the elements and dynamics belonging to the TRS. However, in order to facilitate understanding and use of this methodology we will not take into account the implications or consequences in the security of the TRS. Due to the complexity and specificity of these topics, they will be discussed in a dedicated chapter (Chapter 5).

3.3.1 Observers

They are the basic agents of the TRS. They create and manage the trust used by the whole system. To do that they are responsible for acquiring trust information and calculating trust values.

In the well-known scheme of "source that claim a quality over a target" proposed by Josang [101] they assume the role of source, because they are responsible for generating trust values for the entities of the underlying system.

The main features we identify and analyze in a TRS regarding *observers* are:

- *Number of observed entities*: the observer can be responsible for calculating trust values for a single *entity* or for several of them. A *single entity observer* is usually simpler than *multiple entity observer* due to fact that it needs less memory resources to store the trust information and less computational resources to calculate trust values. However, in many scenarios this difference do not have any significant impact if the required resources for every additional observed entity are insignificant compared to the available resources. Anyway, we should estimate the required resources needed to observe an *entity*. Thus, we can count with a basic unit of cost per observable entity.

- *Observation time*: the time needed to acquire trust information and calculate trust values could be of significant importance. Based on the maximum number of observable entities by an observer, we should estimate the required time to perform these processes. In order to calculate this ratio some matters such as the need of dedicated resources, or the capability of multiplexing data acquisition or trust calculation for several entities simultaneously has to be taken into account. Besides, it is important to identify if both processes are parallelizable.

- *Range of observation*: in a *multiple entity observer* scenario it is important to identify the range of observation of the *observer*. We understand by range of observation the area (physical or logical) that can be observed and has to be observed by the observer. Based on this data and depending on the topology and density of the observable *entities* of the underlying system, we can estimate some important features such as required storage and computation resources, variability of the number and density of the observable entities, etc. Observation time takes even more importance in this scenarios.

- *Area of influence or traceability* of the *entities*: this is a feature of the entities of the underlying system. However, regarding to the previous point, it is important to identify the area of influence of those entities: the maximum distance to an observer that makes the entity observable. This concept might be mixed up with the previous one. Thus, we will explain this feature with an hypothetical example: we can envision a scenario where a policeman is an observer and a thief is an entity that will be potentially observed (caught) by the policeman. The range of observation of the policeman can be a whole city. But the area of influence of the thief it is only a few meters (the distance where the policeman can actually catch the thief). If the act of observation by the policeman were "to see" instead of "to catch", the area of influence of the thief would be higher. And if the act of observation were "to find a clue", the area of influence of the thief would be even higher.

- *Internal or external observers*: most TRS are based on this architectural element, but in cases we can find scenarios where trust information acquisition and trust calculation are performed by external or uncontrollable agents. Examples of this kind of system can be pure-reputation systems, or some ranking or recommendation services. For example a website that ranks movies based on its visitors valuations could be considered as an external-observer-TRS, because the TRS is not involved in the process of visitors acquiring the information about the movie, nor in the process of visitors generating their opinions about the movie.

3.3.2 Trust Information Acquisition

As we described before, in order to create a useful value of trust for the *entities*, *observers* can use a number of sources of information.

After analyzing most of the TMS in the literature (Chapter 2) we can conclude that the main sources of trust information for any kind of TRS are:

- *Perception*, *direct observation*, or *first hand information*: an observer can obtain information by direct observation of the *real world*. Generally, this is a high quality source of information, because it takes into account fresh data, and it is not filtered by third parties. Actually, we should always try to incorporate this kind of information to our TRS.

- *Communicated information*, *witness information*, *indirect observation*, or *second-hand information*: an observer can obtain trust information provided by other entities in the system (usually other observers). Generally, the quality or reliability of this information is worse than the previous one. This information could have been processed and not exactly reflect the observable qualities of the observed entities. However, it has some advantages over direct observation. It allows to gather information beyond the direct acquisition capabilities of the observer (for example, an observer with a sensor temperature could send its data to another observer with a humidity temperature to improve its trust calculation process). Besides, it allow to expand the range of observation (by relying information, a group of observers can act as a bigger one).

- *Memory*: an observer can use previously acquired information in order to improve the performance of the trust calculation process. Memory utilization do not have many significant drawbacks, but the need of more storage resources in the observer. Memory is highly beneficial (almost indispensable) when the entities has to be evaluated for a long period of time or when their acts can influence trust and reputation values long time after those acts has happened. The main drawback of using this kind of information is that it confers a level of inertia to all the processes of the TRS. The more memory information is used the slower the variation of values of trust is. Based on the features of the analyzed system and the goals of the underlying system this inertia can become a critical issue.

- *Categorization*: an observer can use a pre-defined information about an entity based on a distinctive feature. In a social environment categorization is usually known as prejudices, but in this context it does not have any kind of negative connotation. This is specially useful when an entity is new in the system or the observer has not previously interacted with it. Categorization can be used as an accelerator of the initial trust calculation process because it allows to the observer to assign a default trust value to the entity or, at least, the observer has a previous knowledge about some significant feature of the entity. In both cases, it should allow to generate accurate initial trust values. They main drawback of this source of information resides in the fact that categorization can lead to an excessive generalization that do not reflect the actual features of specific entities. Besides, an erroneous categorization could lead to miscalculated trust values.

- *Reputation*: an observer can use the value of reputation assigned for the reputation server(s) to calculate the trust value for the observed entity. This can be useful in early interactions in the same way that categorization was. In a general case, it can be a good source of information when the global perception of an entity is more important than the local perception of its features. However, we cannot forget that reputation is a global and objective concept, but trust is local and subjective. Thus, the utilization of reputation to calculate trust can lead to biased trust values or values that do not reflect the real interactions (observations) of the observer over the entity. If this happens for a long period of time, the benefits of using the theoretically local and subjective concept of trust, can be lost.

- *Reasoning*: an observer can generate trust information about an entity by **processing** all or some of the sources of information previously described. Due to its importance, this process of reasoning stands as one of the main processes of any TRS. It is also known as Trust Calculation Algorithm, and it will be described in detail in the next section.

- *Trustworthiness*: this is one of the most common topics of the Trust and Reputation literature. Therefore, although this concept is not a new source of trust information but a sort of combination of some of the previous ones, we will give more details about its meaning. Based on Becerra [138], Trustworthiness is a characteristic of the trustee, while trust is the trustor's willingness to engage in risky behavior that stem from the trustor's vulnerability to the trustee's behavior.. This definition allow us to express trustworthiness as validated and consolidated trust values assigned to an entity. i.e: a calculated

(*reasoned*) trust value based only previously calculated trust values *memory*. Therefore, in its simplest version, we could express trustworthiness as sort of local reputation from the point of view of the specific observer that calculated it.

Other main features we could identify and analyze in a TRS regarding *Trust Information Acquisition* are:

- *Nature of the information*: information can be quantitative or qualitative. Quantitative information usually leads to more accurate trust values, but qualitative information can be extremely useful when observing abstract or complex features of the entities.
- *Certainty/reliability*: when dealing with information we always have to know the reliability of every trust information source. Ideally, information under a specific threshold of reliability should be discarded or, at least, the system has to be aware of it, and weight it in some way.
- *Redundancy*: it is important to know if a specific quality of an entity is observed through several features of that entity, or if several observers can observe that same quality. Redundancy of information is the main mechanism to deal with low reliability sources of information. The more redundancy, the less reliability we should be willing to accept. Some drawbacks derived from the use of redundant information are: a higher processing complexity of the information, and a new uncertainty factor derived from dealing with different sources of information with different levels of reliability but reflecting the same feature of the entity.
- *Scope*: one of the most simple but important concepts about trust is the fact that trust is a concept associated to a *entity performing a specific service*. It should not be associated to a *entity* as a whole. Therefore, it's important to identify if the sources of information are actually related to a specific service or not.

3.3.3 Trust Calculation Algorithm

In order to create a useful value of trust, *observers* process all or some of the aforementioned sources of information with an internal algorithm. This is a key process for the whole TRS.

The main features we could identify and analyze in a TRS regarding the *Trust Calculation Algorithm* are:

- *Calculation time*: regarding the previous topic, time needed to calculate trust values has implications on most of the other processes involved in a TRS. A short calculation time yields to more flexible TRS and they usually require less time to adapt themselves to environmental changes. However, they could yield to a more unstable behaviors. The opposite advantages and disadvantages can be found when dealing with a long calculation time.
- *Required Computational resources*: the complexity of the algorithm is not the only factor determining the calculation time. Obviously, the resources available to execute the Trust Calculation Algorithm in the *observer* are a key factor too. However, the resources needed to execute the algorithm are not only a limiting element of the calculation time. A minimum amount of computational resources or memory storage can be a hard requirement depending on the algorithm selected. Therefore, the availability of such resources under diverse scenarios should be analyzed. In addition, a description of the degradation of the quality of calculated values should be provided for the TRS under study.
- *Number of observed entities*: the observer can be responsible for calculating trust values for a single *entity* or for several of them. A *single entity observer* is usually simpler than *multiple entity observer* due to fact that it needs less memory resources to store the trust information and less computational resources to calculate trust values. However, in many scenarios this difference do not have any significant impact if the required resources for

every additional observed entity are insignificant compared to the available resources. Anyway, we should estimate the required resources needed to observe an *entity*. Thus, we can count with a basic unit of cost per observable entity.

- *Nature of the information*: information can be quantitative or qualitative. Quantitative trust information usually performs better when: the concept of trust in the TRS is associated to a quantifiable feature of the entities; it is easy to define absolute minimum and maximum values of trust; or when a scale can define the trustworthiness of the entity. Qualitative trust information can be extremely useful when the concept of trust in the TRS is: associated to an abstract or complex feature of the entities in the underlying system; or when the definition of a finite number of statuses can explain the trustworthiness of the entity.

- *Required information*: it is important to identify the required information for the algorithm to be executed. We can find algorithms that can keep working in the absence of information but they provide low quality results when this happens. As they acquire more information, they calculate more precise values. In this scenarios, early stages can be prone of highly misestimate trust values. However, they can count with some of the advantages of the short calculation time algorithms previously described. Other algorithms do not calculate any trust value until the have all the required information. This scenarios have to be analyzed in detail because the absence of only one source of information could stop the whole trust calculation process. Anyway, this feature will always introduce a latency to all the process involved in the TRS.

- *Information consumption*: we have to identify if the algorithm consumes the processed information (i.e: the information can be used only once). This feature can impact in the observer requirements. Consuming-information algorithms might require shorter observation times for the observers to keep the same trust calculation rates than others with no-consuming-information algorithms.

In addition to the algorithmic features previously described, the essential properties of the *trust* information, as defined in the literature [139], have to be known. Some of them have been cited in previous chapters. Anyway, we will detail all of them for completeness.

The Trust Calculation Algorithm should provide trust values taking into account these properties:

- *Scope*: as we described in the previous section, one of the most simple but important concepts about trust is that it is *context specific* (*i.e.*,is a concept associated to a *entity performing a specific service*). It should not be associated to a *entity* as a whole. Therefore, calculating entity-wide trust values is always a delicate approach. It can yield to misestimate the trust values for the entity providing a specific service, and resulting in a degradation of the TRS performance. However, in some scenarios it might be the only kind of information available, or it can be useful as a first estimator in the same way that we discussed before about *categorization*.

  Some works in literate analyze in detail this topic. Marsh [107] defines three types of trust: dispositional trust, the trust of an entity independent from the possible cooperation partner and the situation; general trust, the trust of an entity in another one, independently of the specific situation; and situational trust, which describes the trust of an entity in another one in a specific situation or context.

- *Dynamic*: obviously, *trust* can increase or decrease with new experiences or observations. In addition, it may also decay with time, and new experiences are usually more important than old ones, since old experiences may become obsolete or irrelevant with time.

- *Non-transitive*: trust is not transitive due to the fact that it is a subjective concept. The dissemination of trust information allows an entity to use that information to calculate new trust values, but this dissemination do not imply at all that the disseminated trust

has to be directly assigned to the receptors of that information. In fact, this would lead to a severe malfunctioning and a degradation of the TRS applicability.

- *Asymmetric*: trust is not always symmetric due to the fact that it is a subjective concept. In fact, trust is typically asymmetric. An entity may trust another entity more than it is trusted back. However, when both parties are trustworthy in the long term, they will converge to high mutual trust after repeated interactions. Conversely, if one of the entities does not act in a trustworthy manner, the other entity will be forced to penalize him/her, leading to low mutual trust.

- *Asymmetric Hysteresis Trust Loop*: even though it is not a compulsory requirement, most real-life Trust Calculation Algorithms present a sort of asymmetric hysteresis loop in their calculated values. Trust values tend to increase slowly but they can decrease to zero almost immediately. It is important to identify the behavior of the algorithm in this sense. Usually, the behavior of the TRS will be more conservative, the more asymmetric the algorithm. And the behavior of the TRS will be more tolerant, the wider the hysteresis loop. This topic is related with the concepts of *trust* as a *self-reinforcing and event sensitive* value. *Trust* is usually *self-reinforcing* because entities act positively with other entities whom they trust. Similarly, if the trust between two entities is below some threshold, it is highly unlikely that they will interact with each other, leading to even less trust. And *trust* is usually *event sensitive* because takes a long time to build, but a single high-impact event may destroy it completely.

3.3.4 Disseminators

The trust values calculated by the *observers* can be used by other *observers* or can be used to calculate reputation values. In order to allow this transmission of information some *entities* in the TRS can have the capacity of relaying trust and reputation information messages.

The main features we could identify and analyze in a TRS regarding *disseminators* are:

- Features such as *Number of disseminated sources/observers*, *dissemination range*, *dissemination time*, etc. are completely analogous to those described for the *observers*. They should be analyzed in order to know if they can determine the behavior of the TRS.

- *Information confidentiality*: disseminator could have access to the retransmitted information. There are not too many advantages derived from disseminators accessing to retransmitted trust information, but the fact that if they are observers too, they can use this information to feed their Trust Calculation Algorithms. The main drawback is that disseminator can become target of attacks against the TRS because they have read-access to an important asset (the calculated trust information).

- *Information Filtering*: disseminators can be completely transparent regarding the retransmitted information, or they can modify it in some way. If they can modify the trust information the system can be benefited from processes such as error checking algorithms, or any kind of information sanitization (in a information level, not in a data level). However, disseminators can become target of attacks against the integrity of the TRS because they have write-access to an important asset of the system.

- *Reliability*: as we detailed before, when dealing with information we always have to know the reliability of every component processing that information. We should identify if the disseminators can lead to information loss. If so, we should model it, and analyze the impact on the TRS performance.

3.3.5 Dissemination Process

Transmission of trust and reputation information carried out by the *disseminators* is based on the existence of a specific communication protocol that is commonly called *dissemination protocol*.

Actually, the dissemination process do not have any special feature compared to a generic communication protocol. Thus, the main features we should identify and analyze in a TRS regarding *dissemination protocol* are topics such as: connection oriented vs. connectionless protocols, computational complexity, point-to-point vs. broadcast communications, confidentiality and integrity of the transmitted data, etc.

3.3.6 Reputation Servers, Information Sources, and Calculation Algorithms

Most features we could identify and analyze regarding these topics are analogous to their equivalents related to *observers* and the calculation of trust. Thus, we will not detailed them again and we will only focus on reputation-specific features, or features of special significance in this context.

- *Number of reputation servers*: a TRS usually has a number of *observers* but it is possible to find scenarios where there is only one reputation server, or even none.

  Scenarios without a reputation server are also known as trust-pure TRS. They are used when calculating and disseminating reputation values do not add value to the underlying system because: *i)* most, or all the interaction are local or subjective; *i)* the reputation calculation and dissemination process has a higher cost then the benefits for the observers to use that information in their Trust Calculation Algorithms.

  Scenarios with only one reputation server are common when there is a special entity in the underlying system with resources above the rest. Thus, it is suitable to implement more complex processes or to store more sensitive information. Anyway, the calculation and dissemination of reputation information is a expensive computational and communicational process. Therefore, reputation support should be implemented only when this global and objective concept can improve the behavior of the TRS based on the dynamics of the underlying system. We will analyze this topic in detail in the next Chapter.

  Scenarios with several reputation servers allow to deploy TRSs over larger underlying systems. The main drawback is the growing complexity of keeping information coherence between reputation servers. Hierarchical reputation servers can be implemented to cope with this complexity.

- *External vs. Internal reputation systems*: the reputation server is one of them main architectural elements of a TRS, but we can find scenarios where reputation information acquisition and calculation are performed by external or uncontrollable agents. This scenarios tend to be simpler than those implementing reputation management processes. However, due to the fact that this reputation processes are uncontrollable, they can yield to erroneous behaviors. The external reputation server might not have any feedback from the underlying system. Therefore, if it does not reflect correctly the reputation of the entities it can lead to a degradation of the performance of the TRS.

- *Reputation Information Sources*. Most common sources of reputation information are the disseminated *trust* information of every *entity*) within the system, and reputation values previously calculated by the reputation server itself. Other sources of information can be used such as reputation values from other services. As we said before, this can lead to a misestimate of the reputation value assigned to the service under study. However, it might be useful to accelerate the calculation of initial reputation values.

- *Publicity of the reputation values*. Trust information is very often internal to the TRS (some times, it is even private and accessible only to the *observer* that calculated it). However, as we have described in the previous topic, reputation information can be used for third parties. Therefore, we can identify and analyze if a TRS allows public, restricted or private access to the reputation information. This feature does not have a great computational impact in the behavior of the TRS. It could only demand more resources from the reputation server to manage and disseminate the reputation information to external

systems. However, it is important to remark the fact that information belonging to entities of the underlying system would be accessible for external systems. Obviously, this could impact the privacy of those entities.

3.3.7 Underlying system requirements

Besides of analyzing these architectural elements and processes we should take into account how the TRS is conditioned by the underlying system. The key topics subject of study in this area are:

- *Timing*. Trust information acquisition, calculation, dissemination, etc. are vital processes in any reputation system environment and *when* they happen can modify and determine the features and effectiveness of the reputation system. The three basic timing schemes are: **periodic**, **event oriented** and **periodic adaptive**. Periodic underlying systems usually lead to more complex TRS: the existence of a global trigger is usually required and the transmitted information is higher than in other approaches. Event oriented underlying system optimize the communication resources. TRS processes are executed only when there is new TRS events that actually require to be processed. Periodic adaptive is a trade-off scenario where a periodic polling approach is execute, but its period can change if the number of TRS events is to low or to high. In this way, it tries to optimize the communication and computational resources of the system.

- *Topology*. Related to the *dissemination protocol* we find that the topology of the underlying system is a key factor. We can find as many topologies as in a generic distributed system: client-server, multi-agent systems, ad-hoc networks, etc. Advantages and disadvantages will be those commonly associated to these topologies in a generic communication scheme regarding to: transmission times, reliability, transmission ranges, etc.

- *Limitations of the underlying system*. Before designing any TRS we must take into account all the possible limitations the underlying system can impose: communication or computational resources, storage capacity, power consumption, etc.

- *Requirements and goals of the underlying system*. TRSs are but a way to improve the underlying system performance in a number of specific tasks. Therefore, the most important thing we have to take into account in the TRS analyzing process is to identify if all these requirements and goals have been achieved and up to what point. We will discuss this in more detail in Chapter 4.

3.4 Conclusions

The main goal of this chapter was to provide a methodology to identify and describe the architecture and the dynamics (components and processes) related to any kind of TRS, and predict its behavior.

The compilation of knowledge about Trust Management Systems presented on Chapter 2 has given us a wide perspective of the common elements of these models, their special features, the processes involved in the different dynamics, etc. This global knowledge has allowed us to define a generic architecture for TRS.

The architecture is as simpler as possible but without loss of generality: it is composed of four components: Underlying System, Observers, Disseminators, and Reputation Servers; and five processes: Trust Information Gathering, Trust Information Calculation, Trust and Reputation Dissemination, Reputation Information Gathering, and Reputation Calculation.

Based on this architecture, a methodology to analyze TRS is presented. The methodology is based on checking the presence of every component and process of the proposed architecture, identifying their main features and analyzing the consequences of all those features in the behavior of the individual components and processes and in the system as a whole. To facilitate this task, we provide a checklist with the main features to be analyzed.

As main objection to this approach we could mention that we cannot claim that the checklist of features presented is complete.

However, this checklist has not been presented as it has to be complete at all. The essential process behind this methodology is to identify every component and process of the proposed architecture into the TRS under study. Specific TRS could require a more in-deep analysis of some elements of the architecture, where others could even omit the analysis of some topics. The proposed checklist is just a guide to help the analyst through the process, pre-identifying the most common and usual features that can determine the system behavior.

Finally, it is important to remark that the TRS architecture and the methodology presented in this chapter stand as one of the main contribution of this Ph.D. Thesis.

4. TRS Design Methodology

4.1 Introduction

After identifying the fields of application of TRS, the state of the art in the literature, the main challenges this discipline faces, and describing a methodology to analyze TRS, we will propose a methodology to design TRS in order to bring TRS technologies closer to real-life scenarios.

Following the analogy presented in the previous Chapter, as we have already described a methodology to cope with the description, and prediction of the behavior of any kind of TRS, now is the turn of moving on to the next stage. Therefore, the main goal of this chapter is to provide this basic but fundamental tool (a methodology) to control the architecture and the dynamics (components and processes) related to any kind of TRS in order to achieve a specific improvement in the performance of the underlying system.

In the Section 4.2 we present the main guidelines of the proposed design methodology. Next sections will provide a more in-deep analysis of main phases of the methodology: characterization (Section 4.3) and mapping (Section 4.4). Then, we present some topics related to the designing process in Section 4.5. Finally, in Section 4.6 we draw some conclusions about the application spectrum and the limitations of the proposed methodology.

4.2 Methodology

As we described before, the main goal of this chapter is to provide a methodology to control the architecture and the dynamics (components and processes) related to any kind of TRS.

Therefore, the designed TRS has to achieve some specific improvements in the performance of the underlying system.

The methodology is actually simple and straightforward. It is based on:

- Identify if TRS are a suitable approach to cope with the goals or requirements of the underlying system. TRS are not a universal solution for any kind of problem. They perform especially well when dealing with some specific types of issues.
- Identify the main limitations and restrictions of the underlying system. The designing process tries to bring TRS to real-life scenarios. Therefore, it is critical to adapt them to the actual limitations and resources of the underlying system.
- Define the concepts of *trust* and *reputation* by associating them to some specific features of the underlying system. Not every underlying system works directly with the concepts of *trust* and *reputation*, but we can propose analogies between *trust* and some subjective and local features of the entities in the underlying system, and between *reputation* and some objective and global features. This mapping from the application-problem domain to the TRS domain is essential.
- Define which elements in the underlying system will assume the roles of the components of the TRS. i.e: *observers*, *disseminators*, and *reputation servers*.
- Identify data and information in the underlying system that will constitute the sources of information for the *Trust Calculation and Reputation Calculation Algorithms*.

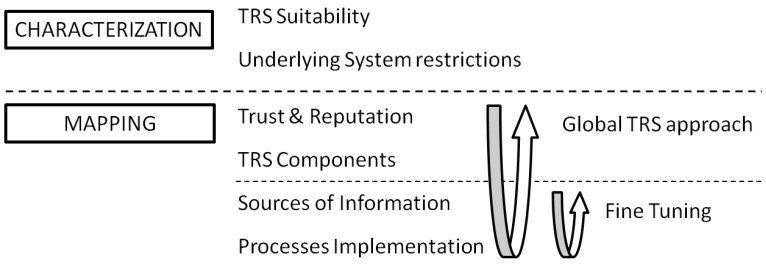


Figure 4.1: Design process. Iterative loops

- Propose and validate how the processes of the TRS architecture (*i.e.*,Trust and Reputation Information Acquisition and Calculation Algorithms, and Dissemination process) will be implemented over the underlying system.

As shown in Figure 4.1, the first two phases of the methodology define the *characterization process*, and they will be detailed in Section 4.3. The next four phases define the *TRS mapping process*, and they will be detailed in Section 4.4.

As we will analyze in this Chapter, some of this tasks can lead to re-evaluate previous phases if the results are far from the expected performance or functionality. Therefore, we will get to the final design and implementation through a iterative process.

4.3 Characterization

The first phase of the proposed design methodology is to identify if TRS is a suitable approach to cope with the goals or requirements of the underlying system.

As TRSs are not a universal solution for any kind of problem, we have to identify the main criteria that allow us to evaluate the performance of a TRS. Based on this criteria we can check if they are suitable to be applied to a specif problem because they are aligned in their goals and required performances of the underlying system.

4.3.1 Basic Criterion

As we have already presented in this chapter and as we will detail in next sections, the knowledge about the *trust* and *reputation* of the *entities* of an underlying system can be used in a number of different ways, depending on the purpose of the TRS.

However, regarding previous works in the literature, we find that TRS usually had a simpler approach: they were focused in utilizing TRS as a mechanism to make decisions in complex and distributed scenarios.

In these scenarios, the basic criterion to rate the performance of a TRS is the **accuracy and precision** of the estimated trust and/or reputation values regarding to the actual trustworthiness and/or reputation of those entities. Thus, the more accurate and precise the estimations, the more accurate and precise the decision will be.

Even though it might look like a simple approach, this criterion yield to the use of TRS in a wide number of practical applications as we presented in Chapter 1: decision-making, ranking engines, suggestion engines, etc.

However, from the point of view of this Ph.D Thesis this is an insufficient approach. The study of the main fields of application of TRS presented in Section 1.2.3, and the compilation and analysis of Trust Management Systems presented in Chapter 2 has given us a wide perspective that enables us to identify and propose a wider range of applications where TRS can be extremely efficient.

These fields of application are based on extended criteria that go beyond the basic criterion of improving the accuracy and precision of estimated *trust* and *reputation* values.

4.3.2 Extended Criteria

The essential idea behind the proposed extended criteria is based on analyzing *trust* and *reputation* as ratios that allow us to **compare entities**.

This comparison goes beyond just using this information to sort the entities from the highest-rated to the lowest-rated ones. It is focused on identifying groups of *entities* with similar *trust* and/or *reputation* values. This leads us to be able to identify anomalous *entities* (not belonging to any group) or even to identify anomalous groups of *entities* (groups different from the majority of groups).

Hereinafter, for the clarity of the explanation we will call both *anomalous* or *ill-behave* to this *entities* or groups of *entities*.

Therefore, if we focus on this new approach, we can identify the following criteria:

- **Response time**. It is the elapsed time since an anomaly behavior started until it is detected, i.e., the *trust*/*reputation* of ill-behave *entities* begin decreasing below a specific threshold.
- **Isolation capacity**. It is the portion of ill-behaved *entities* that are detected as anomalies.
- **System degradation**. It is the portion of well-behaved *entities* detected as anomalies.

4.3.3 Typology

Regarding the basic criterion previously presented, we have already identified that TRS are a good approach to cope with some specific underlying systems: those that are based on trying to achieve precise and accurate values of *trust* and *reputation* in order to make decisions, generate rankings of entities, suggest recommendations, etc.

The analysis of the extended criteria yield to a new set of applications where applying TRS can be an suitable approach to cope with the requirements and goals of an underlying system.

Therefore, we will explain in more detail the importance of each of them and the way they can be useful for presenting a typology of suitable TRS applications.

- **Minimization of the Response Time**: It is the determining factor for all those applications where the TRS is being used as a detection mechanism: attacks, anomalous behaviors, etc.

  Typical examples of applications that can deploy a TRS to minimize this factor are the Intrusion Detection Systems, both Network Intrusion Detection Systems and Host Intrusion Detection Systems.

- **Maximization of the Isolation Capacity**: It is the determining factor in all those applications where few elements can make a lot of damage to the system. The damage can be twofold: they can have a negative effect on other *entities*, either in degrading their performance or even rendering them completely useless; on the other hand, these elements may provide critical information or critical functionality and their incorrect functioning can highly degrade the performance of the complete system.

  Typical examples of applications that can use TRS to minimize this factor can be a system for contingency against DoS or DDoS, where we TRS can prevent further propagation of the damaging effects of the attack.

- **Minimization of the System Degradation**: It is the determining factor in the cases where it is more important as many entities functioning properly as possible. This added value can be found in three different types of scenarios: the entities provide a higher data throughput to the system, the entities provide higher processing capacity to the system, or the entities provide stronger validity to the information generated in the system.

  Typical examples of applications that can use TRS to maximize the performance can be auto-healing or even load-balancing system, both highly deployed in every kind of distributed systems such as WMN, ad-hoc networks, WSN, etc.

This typology of applications, based on the extended criteria previously presented, opens a new range of fields of applications where TRS can be a good approach to achieve the goals of the underlying system, or to improve its performance in a specific way. This topology overtakes the idea of a TRS as a simple decision-making mechanism and presents TRS as a suitable technology to implement detection and isolation systems, auto-healing and load-balancing policies, etc.

In order to validate this hypothesis, a number of TRS applied over real-life scenarios will be presented in the chapters of Part II of this Ph.D. Thesis.

4.3.4 Underlying System

Once we have identified if the requirements of the underlying system can be fulfilled or improved by applying a TRS because they are within the scope of application of this systems (*i.e.*,decision making, detecting anomalies, isolation anomalies, minimization of system degradation/maximization of system performance/throughput), we have to identify and analyze the main limitations that the underlying system can impose to the TRS.

We do not have to forget that the goal of the proposed design methodology is to bring TRS to real-life scenarios. Thus, it is essential to know the limitations and restrictions that a specific underlying system can set.

The key topics subject of study in this area are:

- *Resources*: the main limitations of the underlying system might come from available resources of the *entities*: communication or computational resources, storage capacity, power consumption, etc.
- *Reliability*: besides the quantitative quality of the *entities*' resources it is important to analyze their qualitative quality. Therefore, we should identify topics such as the guaranteed availability of the *entities*, the reliability of the information managed within the system, etc.
- *Topology*: TRS usually works over complex and distributed systems. Thus, we can find any topology: client-server, multi-agent systems, ad-hoc networks, P2P networks, etc. Regarding these topologies we should identify and analyze the limiting and conditioning factors of topics such as: transmission times, transmission reliability, transmission ranges, etc.
- *Dynamism*: temporal and spatial variability of the *entities* within a system can be decisive in order to effectively apply a TRS over it. Thus, we should identify some relevant rates such as: number and rate of *entities* in-to/out-of the system, the spatial distribution of those *entities* (*i.e.*,density), the range of location of every kind of *entity* (*i.e.*,mobility), etc.

Based on this main topics we should create a more precise image of the underlying system regarding the possibility of applying a TRS to improve its performance. All these features will be taken into account in the mapping phase, and in the implementation and deployment phases.

4.4 TRS Mapping

From the point of view of describing the design methodology, this is the most specific phase regarding TRS. The result of this phase will be deeply determined by the designer's knowledge about TRS advantages, characteristics, and limitations. Besides, as in any other design methodology, the knowledge about the field of application is essential.

However, the goal of this methodology is not to create a playbook with the best designing choices for every scenario, but to identify all the processes involved into applying TRS to the improvement of an underlying system. Thus, the designer will have a systematic methodology to face this task.

After identifying if a TRS can be an appropriate approach to cope with the underlying system, and analyzing its main limitations, the next phase of the designing process aims to translate terms belonging to the field of application of the underlying system into the TRS architecture proposed in Section 3.2.

If we achieve to do that, we can apply all the knowledge, tools, and benefits derived from the use of TRS into a completely different field of knowledge.

Although we will delve into this idea, we will give a short case of use to clarify this approach. Let's imagine an hypothetical scenario based on a backbone network routing traffic to enable some kind of information transmission service. This scenario do not know anything about *trust* and *reputation*, but about transmission rates, round-trip times, etc. However, we can create a direct association between this problem-specific concepts and the proposed TRS architecture. We can identify that the best routes are those managed by those routers of a specif model, or those handling with a specific amount of traffic, or those having short routing paths that yield to short round-trip times, etc.

Thus, we can assume that the *trust* of a router in its neighbors is related to those problem-specific terms. Therefore, we model *trust* as a new variable derived from processing those Trust Sources of Information by a specific Trust Calculation Algorithm. TRS do not know anything about transmission rates, round-trip times, or router's models, by they are very efficient at detecting and isolating trust anomalies.

Therefore, by detecting and isolating anomalies in trust values that are actually based on essential problem-specific features, the TRS now becomes a efficient tool to detect and isolate anomalies in the underlying system: over-utilized routers, not optimal routing paths, malfunctioning of routers, etc.

This simple approach can be extended as far as needed: routers communicate their trust information to other routers, we define different behaviors based on extreme situations (infra-utilized/over-utilized routers), the routers store previous interactions with other routers in order to use historical information in their routing decisions, a central or distributed service provider initial values of router's reliability to other entities within network, etc.

How we can systematically apply this *mapping process* to real-life underlying systems is described in the next sections.

4.4.1 Trust and Reputation

The first step of the mapping process is to define the concepts of *trust* and *reputation* by associating them to some specific feature or to some specific measurement of performance of the underlying system. This mapping from the application-problem domain to the TRS domain is essential.

As we have previously mentioned, we can propose analogies between *trust* and some subjective and local features/performance measurements of the entities in the underlying system. And in the same way, between *reputation* and some objective and global features/performance measurements.

The association between an *underlying system* feature and the concepts of *trust* and *reputation* has not to be strict. Trust and reputation can be associated to a complex feature (a feature resulting of processing several features).

An additional characteristic of this resulting values of trust and reputation is that they have to allow us to **sort** and **group** entities. The ability to sort entities will allow the TRS to use the concept of preference in its algorithms. The ability to group entities will allow the TRS to use the concept of difference in its algorithms.

Finally, it is important to remark that not every underlying system has relevant features/performance measurements for both local and global concepts. Therefore, it is not compulsory the mapping process yields to an architecture having always the concepts of both trust and reputation. An TRS dealing only with *trust* is usually known as *trust-pure TRS*, and a TRS dealing only with *reputation* is called *reputation-pure TRS*.

4.4.2 Architectural components

The goal of this step of the mapping process it to assign the roles of observers, disseminators, and reputation servers of the TRS to specific entities or types of entities within the underlying system.

Observers are the key architectural component regarding the concept of trust. Therefore, they usually have a local scope and are close or have an easy access to the selected sources of trust information.

They must count with the communication and computational resources required to implement the processes they are responsible of (trust information gathering, trust values calculation, etc.)

Obviously, if the concept of trust is not needed in this scenario, the observers as architectural component of the TRS will not be needed either.

Disseminators are the key architectural component regarding the transmission of trust and reputation information throughout the system. They must count with the communication and computational resources required to implement the selected dissemination protocol.

Reputation Servers are the key architectural component regarding the concept of reputation. Therefore, they usually have a global scope and have an easy access to the selected sources of reputation information (calculated trust values, etc.). They must count with the communication and computational resources required to implement the processes they are responsible of (reputation information gathering, reputation values calculation, etc.)

Obviously, if the concept of reputation is not needed in this scenario, the reputation servers as architectural component of the TRS will not be needed either.

Finally, it is important to review the main responsibilities and features of this architecture components as they were described in Section 3.3. A systematic analysis of these responsibilities and features will allow us to identify if the proposed architectural components comply with the expected responsibilities, and will allow us to characterize the their expected behavior.

4.4.3 Sources of Information

In the first step of the mapping process, we proposed a mapping between some feature or measurement of the performance of an entity providing a service, and the concepts of trust and reputation. This mapped feature or performance measurement is usually based on processing several basic features of the underlying system.

An important, and not simple task when mapping TRS, is to identify and choose the most appropriated features that will allow to calculate accurate and precise trust and reputation values.

This task is deeply application-problem dependent. Therefore, it is not possible to give a comprehensive list of sources of information for a generic scenario. However, based on the Trust Information Acquisition Process described in Section 3.3.2, we can systematize the process of identifying them by following the proposed taxonomy of sources of trust information. This taxonomy identifies perception, communication, memory, categorization, reputation, and reasoning. Based on this approach, the designer has to analyze the underlying system and try to identify if any of this sources apply to the studied scenario.

4.4.4 Architectural processes

Selecting and implementing an specific **Trust Calculation Algorithm** is one of the most complex tasks when designing a TRS. However, once we have chosen the *Trust Information Sources* and the meaning of the concept of trust for our specific problem, the process of generating a valid *Trust Calculation Algorithm* basically becomes and algorithmic problem. i.e: we have to find the best way to obtain a behavior in an output metric by processing a set of inputs.

Actually, most of the Trust Management Systems described in Chapter 2 are focused on this process. Therefore, it might be useful for a designer to know the field of application of those algorithms.

Anyway, as we described before, the goal of the *Trust Calculation Algorithm* is to provide a trust values that allow us to sort and group entities. Thus, we can try and evaluate any algorithm providing this two features.

A common approach when designing *Trust Calculation Algorithms* is based on previously define the Asymmetric Hysteresis Trust Loop (as it was described in 3.3.3), and then try to find a function that fits this loop. Concepts such as *trust update* and *trust revocation* are common in the literature regarding this designing *Trust Calculation Algorithm* approach.

In order to evaluate if an algorithm is suitable for the designed TRS, the specification of a performance test-bench is suggested. This test-bench has to identify the expected outputs associated a different scenarios. These scenarios can be specified through the definition of ranges of variation of their input values, changes in the architectural components, the presence of external elements that can modify the normal behavior of the underlying system, etc.

The same reasoning applies to the process of selecting and implementing a specific *Reputation Calculation Algorithm*.

Regarding the *dissemination process*, the steps to select and implement an specific protocol do not differ from those used in the process of implementing a generic communication protocol. It will be determined by the topology of the underlying system, the communication and computational resources of the entities chosen to be *disseminators*, etc.

4.5 Related Topics

In order to offer a comprehensive view of the process of designing, implementing, and deploying a TRS into a real-life scenario, we will present some topics regarding the last phases of this process.

4.5.1 Implementation and Deployment

The characterization and mapping phases are essential in the process of creating a conceptual framework for designing a TRS to improve the performance of an underlying system. However, this process is not complete until all the proposed components and processes are actually implemented, tested, and deployed.

To optimize the required time and the expected results of these phases we propose an iterative approach. In fact, we differentiate two kind of optimizations: fine-tuning optimizations, and a global T&R approach.

- *Fine-tuning optimizations*: these optimizations have to do with the implementation of the processes proposed in the TRS architecture. They are related to these tasks of the mapping stage: selection of information sources, and implementation of the calculation algorithms and the dissemination protocol.

  Some times the selection of the architectural components can be correct, but the implementation of the related processes can be sub-optimal. In this cases, an iterative approach is suggested. A performance test-bench or, at least, a set of minimum quantified performance results, has to be defined. Progressive iterations over the configuration, the implementation, and the utilized sources of information have to be carried out until the requirements are fitted. Restrictions on communication or computational resources, time of executions, etc. have to be taken into account.

- *Global T&R approach*: these optimizations have to do with the most general and fundamental topics of the mapping process: the mapping of the concepts of trust and reputations, and the assignation of the roles of the TRS components (i.e: observers, disseminators, and reputations servers). In some cases, a fine-tuning optimization is not enough to cope with the requirements of the underlying system because more deep and fundamental decision have been incorrectly made. In these situations, the relevant features of the underlying system are not correctly reflected into the concepts of trust and reputation. Therefore, trying to optimize algorithms based on wrong premises is useless.

This situations usually reflect a lack of knowledge about the specific field of application. To cope with these issues, a reformulation of the problem is suggested: recharacterizating the underlying system, and acquiring a deeper knowledge of its components and dynamics.

4.5.2 Security

Regarding TRS security, we cannot forget that the act of trusting is associate to the act of delegating a specific responsibility. When we deal with these concepts, the idea of risk is always present. This refers to the risk of generated expectations not being accomplished or being accomplished in a different way we anticipated. Therefore, any system using a TRS to improve its behavior is especially vulnerable if the premises it is based on are attacked.

Thus, one of the goals of a TRS is to ensure that trust and reputation values correctly reflect the actions taken by the entities in the system and cannot be manipulated or accessed by unauthorized entities.

All the components and processes of a TRS can be attacked, and those attacks can degrade the performance of the TRS, and even compromise the confidentiality, availability, integrity, etc. of both the TRS and the underlying system.

Due to the importance of these issues, we will dedicate the Chapter 5 to analyze the security of TRS.

4.6 Conclusions

The main goal of this chapter was to provide a methodology to effectively apply a TRS in order to improve the performance of an underlying system in a real-life environment. The methodology is actually simple and straightforward.

First, we have to identify if TRS are a suitable approach to cope with the goals or requirements of the underlying system, and identify its main limitations and restrictions. We have called *characterization phase* to these processes.

Then, we have to define the concepts of trust and reputation by associating them to some specific features of the underlying system; identify and select which elements in the underlying system will assume the roles of the components of the TRS architecture, identify data and information that will constitute the sources of information for the Trust Calculation and/or Reputation Calculation Algorithms. Finally, we have to propose and validate how the processes of the TRS architecture will be implemented over the restrictions of underlying system. We have called *mapping* to these processes.

Finally, we have presented some related topics such us an iterative process to implement, validate, and deploy the designed TRS, and some security considerations.

As main objection to this approach, we could mention that we do not offer a complete pattern-based design methodology.

Being able to identify a number of generic features or parameters which would allow us to characterize an application based on the requirements imposed on a TRS, would make it easier for us to identify patterns of standard-applications. Thanks to this pattern-based depiction and modeling we could create a knowledge foundation. This knowledge could allow us to face the resolution of new situations in an easier way and without the need of carrying out any initial work. Finally, having a model or a pattern for a specific type of problem would allow us to apply previously proposed solutions to solve similar problems in other fields of application.

However, design patterns are an advanced designing topic and we have considered that a complete analysis in this area would go beyond the limits of this Ph.D Thesis. Actually, it alone could be considered as a research line for a complete Ph.D Thesis work.

Finally, it is importance to remark that the methodology presented in this chapter stands as one of the main contributions of this Ph.D. Thesis.

5. TRS Attack Taxonomy

5.1 Introduction

One of the goals of a TRS is to ensure that trust and reputation values correctly reflect the actions taken by the entities in the system and cannot be manipulated or accessed by unauthorized entities.

For example, this is not achieved if entities can falsely improve their own reputation or degrade the reputations of others [140]: misbehaving entities might obtain unwarranted services or honest entities can be prevented from obtaining those services.

Regarding TRS security, we cannot forget that the act of trusting is invariably attached to the act of delegating a specific responsibility and, when we deal with these concepts, the idea of risk is always present. This refers to the risk of generated expectations not being accomplished or being accomplished in a different way we anticipated. Thus, we can see that any system using trust to improve or enable its behavior, because of its own nature, is especially vulnerable if the premises it is based on are attacked.

However, even though the importance of this matter, a taxonomy to identify TRS attacks has not been yet proposed. Different approaches in the research literature enumerate some of the most common attacks against TRS but none of them tries to provide a holistic analysis.

In this Chapter we will present the tools and the analysis framework that will allow us to define such taxonomy.

To achieve this goal, the rest of this Chapter is organized as follows: Section 5.2 explains the basis of security taxonomies and the most studied security topics regarding TRS. Section 5.3 presents a generic framework to analyze the security assets and potential attacks against any kind of system. Based on this framework, Section 5.4 presents the proposed taxonomy of TRS attacks. In Section 5.5 we present a case of use where the taxonomy is applied to a real-life scenario. Finally, in Section 5.6 we draw some conclusions.

5.2 Related Work

As we said before, we cannot find any holistic analysis of TRS security in the literature. Common approaches identify some features that allow us to classify TRS attacks and some of them offer lists of interesting attacks.

On the one hand, two of the most important feature vectors that allow us to classify the different kinds of attacks are:

- Passive attacks and active attacks. A passive attack occurs when an unauthorized agent gains access to a resource of the system but does not modify its content. Passive attacks include eavesdropping and traffic analysis, among others. An active attack occurs when an unauthorized entity modifies a resource of the system.

- Insider attacks and Outsider attacks. If an agent is authorized to access system resources but employs them in a malicious way, it is classified as an insider attack. On the other hand, an outsider attack is initiated by an unauthorized or illegitimate user. They usually acquire access to an authorized account or entity and try to carry out an insider attack.

Furthermore, the most popular attacks against TRS in literature are described bellow:

- Sybil attack: malicious agents can use multiple network identities [141].
- False information or false recommendation: malicious agents may provide false recommendations/information to isolate good agents while keeping malicious ones connected. It has three main variants: Bad-mouthing attack, where attackers manipulate reputation of surrounding agents by falsely decreasing it [90], [142]; ballot-stuffing, where attackers manipulate reputation of surrounding agents by falsely increasing it [143]; and self-promoting, where attackers manipulate their own reputation by falsely increasing it [142].
- Incomplete information: malicious agents may not cooperate in providing complete information.
- Initial Window: if agents rely only on their own experience in evaluating other agents, they are vulnerable until they find other trustworthy agents, because they do not have enough information to identify and avoid cheaters [144].
- Re-entry: if attackers can create new identities freely, this presents the opportunity to remove bad reputation by creating a new identity [144].
- Whitewashing: in some systems, attackers can repair their reputation completely by using some system vulnerability [145].
- Exit: attackers planning to leave the system have no further need for keeping their reputation. Thus, they can cheat freely without consequence.
- Value Imbalance: in some TRSs, all reviews are weighted equally, regardless of the importance of the action reviewed. This presents an opportunity of attack: an entity can honestly execute minor actions, then use the reputation gained to cheat on very important ones [144].
- Selective misbehaving attacks: agents can behave badly but selectively to other agents, or they can alternatively behave well and badly to try to stay undetected, or they can even behave differently to nodes in different groups to make the opinions from the different groups conflicting, and lead to non-trusted relationships between them.

Besides, some attacks, although they are not specific for this field, are usually analyzed when dealing with TRS: loop attacks, wormhole attacks, black-hole/gray-hole attacks, packet modification/insertion, replay attacks, DoS attacks, etc.

In this way, we can see that there are a diverse set of attacks but there does not exist a holistic framework to analyze them. Thus, to offer a holistic approach to TRS attack identification, we propose a new taxonomy in Subsection 5.4.

The purposes of any taxonomy are diverse. A taxonomy allows for previous knowledge to be applied to new attacks as well as provides structured tools to identify such attacks. Finally, a taxonomy also provides a holistic approach to classify attacks.

The features required to define a good taxonomy are [146]–[149]: **accepted**, it should be structured, so it can be generally approved; **comprehensible**: it has to be able to be understood; **completeness/exhaustive**: it should account for all possible attacks and provide categories accordingly; **determinism**: the procedure to classify attacks must be clearly defined; **repeatable**: classifications should be repeatable; **mutually exclusive**: it has to categorize each attack into one category; **terminology**: existing terminology has to be used to avoid confusion and to increase previous knowledge with it; **unambiguous**, based on the defined categories, there is no ambiguity with respect to an attack's classification;

Anyway, a taxonomy could not necessarily meet all the requirements identified above. It depends on its specific goals.

Related to the evolution of attack taxonomies, two of the first taxonomies in the security field were the Protection Analysis (PA) [150] taxonomy and the Research in Secured Operating Systems (RISOS) [151]. They were focused on vulnerabilities instead of attacks, but they provided a solid background used by later taxonomies.

Bishop made important contributions to the field of security taxonomies. In [152], he presents a taxonomy of Unix vulnerabilities in which the underlying flaws or vulnerabilities are used to create a classification scheme. They are classified based on six axes: the nature of the flaw, the time of introduction, the exploitation domain (what is gained through the exploitation), the effect domain (what can be affected by the vulnerability), the minimum number of components necessary to exploit the vulnerability, and the source of the vulnerability. Bishop and Bailey [153] also performed a complete analysis of other vulnerability taxonomies, such as PA or RISOS.

In [154], Howard presents a taxonomy of computer and network attacks based on factors such as the attacker motivation and objectives. In this way, it can be considered a process-driven taxonomy, rather than a classification taxonomy.

In 2001, Lough [155] proposed VERDICT (Validation Exposure Randomness De-allocation Improper Conditions Taxonomy). It is based upon the characteristics of attacks, namely: improper validation (insufficient or incorrect validation allows an unauthorized access); improper exposure (a system or information is improperly exposed to attack); improper randomness (not enough randomness in the system behavior); improper de-allocation (information is not properly deleted).

Hansman and Hunt [156] proposed a taxonomy based on four dimensions that can be applied to cover both network and computer attacks. The first dimension is the attack vector, the second dimension identify the target. The third one consists of the vulnerability classification based on Howard's taxonomy [147], and the fourth describes the payload or effects involved in the attack.

In conclusion, there exists a high number of identified attacks against TRSs and some attack taxonomies in the literature, but a generic security framework to identify all viable attacks against TRSs in a holistic way has not yet been proposed.

5.3 Proposed Security Framework

In this section, a security framework based on well-known security topics is presented. Together with the TRS architectural analysis performed in previous sections, it will be the other key element of our proposed attack taxonomy.

This framework will be presented in a historic way, beginning with the CIA triad. For over twenty years, information security has held the CIA triad (confidentiality, integrity and availability) to be the core principles of information security: confidentiality was addressed by LaPadula and Bell in 1976 in their mandatory access control model for Honeywell Multics [157]; integrity was addressed by Clark and Wilson work in 1987 [158]. Anyway, the CIA triad is a very simple model with narrow application, that cannot adequately describe many important security objectives. Thus, there have been attempts to augment the CIA triad with more fundamental concepts. The most relevant augmentation could be the one made by Parker [159], that he called the six atomic elements of information (commonly known as the Parkerian hexad). These elements are confidentiality, possession, integrity, authenticity, availability, and utility. However, this model is also limited and more topics have been considered later, such as accountability or non-repudiation.

First of all, to carry out a complete analysis of possible attacks to TRS, we will identify the relevant topics that we should take into account from the viewpoint of security. In order to do this, we propose a holistic security framework, based on the augmentation and redefinition of the CIA triad and the Parkerian hexad.

The framework is divided into basic entities and topics. The topics are divided into: wide scope topics, that affects to all the other topics; primary topics, that are necessary and sufficient to perform any security analysis; and derived topics, that can be defined as combination of

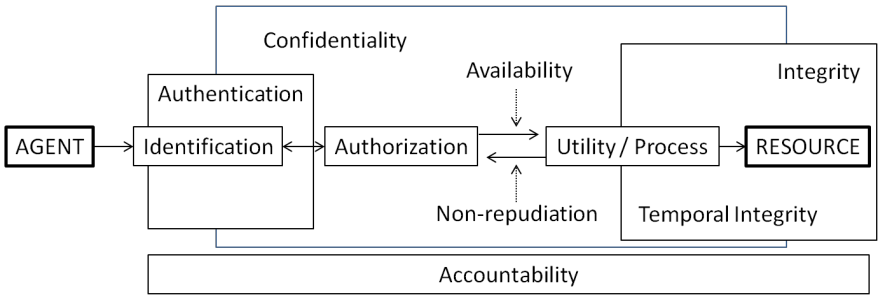


Figure 5.1: Security Framework

primary topics. Derived topics are included in this framework due to its importance in the literature and because they can be very useful to carry out a deeper and detailed analysis if these sub-topics are especially relevant in the different scenarios analyzed. All the topics are conceptualy organized as showed in Figure 5.1. The meaning of every element is shortly described bellow.

The basic entities of our framework are:

- **Agent**: agent means an entity trying to perform an action within a system.
- **Resource**: resource means the entity used to offer a feature within a system. Data, information, knowledge, any kind of physical resource, etc. are examples of resources.

The only wide-scope topic is:

- **Accountability/Activity logging**: Accountability is assurance in tracing all activities within the system. It can be applied in combination with any of the factors detailed bellow.

The primary topics of the framework are:

- **Authentication**: Authentication means the processes related to check if an agent can fulfill a challenge-response process. The challenge can consist in login/password/PIN prompts, bio-metric constant analysis, or any kind of requirement, or combination of requirements, that an agent has to fulfill within the system.

  The classical concept known as Identification could be expressed in this context as a specific type of authentication. Identification means the act of determining who someone or what something is. It should be based in specific features of the agent uniquely associated to it. Thus, from the viewpoint of the system, the agent and the set of selected features are exactly the same. In an optimal identification, the values of the features should be unique for each agent. Moreover, identification can be applied to identify resources as well.

  It is important to mention that identification and authentication are not exactly the same process. They are usually confused, though. Identification can be considered as an strict authentication method where the challenge consists in providing enough credentials to identify the agent uniquely within the system compared to other agents.

  Attacks to authentication are usually called credentials theft, and attacks to identification are referred to as impersonation, man in the middle, etc.

- **Authorization**: Authorization process shows what resources an agent is permitted to access and what processes it will be allowed to perform after it has successfully been and authenticated. The access control mechanisms are usually defined in policies. Attacks against authentication are usually called privilege escalation.

- **Availability**: Availability means having timely access to the utility/process (as described bellow) whenever it is required by an authorized agent. The Parkerian element possession can be expressed as a specific type of availability where physical access is required. Attacks to availability are usually referred to as Denial-of-Service (DoS), and attacks to possession are usually known as robbery

- **Utility/process**: Utility means usefulness. It means that the action an agent wants to perform over a resource could be carried out properly. Utility is often confused with availability, but it is important to understand that availability is focused on an agent (when it is trying to access a utility), and utility is focused on the process itself (that should be carried out correctly).

Finally, the derived topics are:

- **Confidentiality**: Confidentiality refers to limits on what agent can access what kind of information. This process is usually associated to cryptography, although it is valid for any method used to protect information from unauthorized access. It can be considered a combination of the authorization and utility (access) topics, dealing with the resource information. Attacks to confidentiality are usually called eavesdropping, sniffing, spying, etc..

- **Integrity**: Integrity means the act of being consistent with the intended state of information. Any unauthorized modification of data, whether deliberate or accidental, is a breach of data integrity. It can be considered a combination of the authorization and utility (modification) topics, dealing with the resource information. A special case of integrity is Temporal Integrity where the aspect of the information that is modified is time: when it has happened.

  Attacks to integrity are usually called manipulation, and attacks to temporal integrity, delay and re-transmission.

- **Non-repudiation**: It means that one party of a transaction (agent-resource) cannot deny having received a transaction nor can the other party deny having sent it. It can be considered a combination of the identification, utility/process, and accountability topics.

All these factors constitute a security framework proposed to analyze the security of any kind of process: services offered by a web server, access to a file system, operation of control and monitoring systems, etc.

In this work, they are applied to TRSs in order to identify an attack taxonomy composed of any possible security threat against their basic processes and resources. The description of this taxonomy is provided in the next section.

5.4 Trust and Reputation System Attack Taxonomy

Our taxonomy is based on the TRS architectural model proposed in Section 3.2 and the security framework proposed in Section 5.4.

In order to identify all the different attacks against TRS, an analysis of the four agents (entities, observers, disseminators, and reputation servers) and the basic processes related to TRS (trust and reputation information gathering, calculation, and dissemination) will be performed from the viewpoint of each topic considered within the security framework.

The results of this analysis are presented and described below. In the following tables viable attacks to the different processes of TRSs are presented, and a measure of attention received by the research community has been included (from five stars: ★★★★★, assigned to the most studied attacks, to zero stars ☆☆☆☆☆, assigned to attacks without any research work in the field of TRSs). No previously cited attacks in TRS literature have been marked with **(*)**.

5.4.1 Attacks against gathering T&R information

In terms of the security framework detailed in the previous section, the process of gathering T&R information could be described as follows: *observers* from the TRS architecture are the *agents*, the process they try to carry out is to gather information from the *resources*, in this case, *entities* or other *observers*. The results of this analysis are presented in Table 5.1.

Topic	Gathering T&R information	
Authentication	Credential Forgery	★☆☆☆☆
Identification	Man in the Middle, stolen identity, Clone, Sybil, Re-entry	★★★★☆
Non-repudiation	Entity/observer Misbehavior, Confusion	★★☆☆☆
Authorization	Gathering-based Privilege Escalation (*)	☆☆☆☆☆
Availability	DoS against entities and observers	★★★★★
Utility/process	Bad-Mouthing, Ballot-Stuffing, Incomplete information, Null information, Selective information, Entity Malfunction	★★★☆☆
Confidentiality	Information and Behavior Eavesdropping	☆☆☆☆☆
Integrity	Second-Hand Information Manipulation	★★★☆☆
Time Integrity	Second-Hand Information Delay/Re-transmission	★★☆☆☆

Table 5.1: TRS Attack Taxonomy. Gathering T&R Information

Attacks related to authentication, identification, and non-repudiation rely on the fact that the identity or the authentication tokens of the *entities* might be broken in some way. In TRS most attacks of this group are focused in identification. In this way, we can find popular attacks such as man in the middle attacks, stolen identity attacks, clone attacks, or Sybil attacks. But there are more subtle attacks included in this group, such as re-entry attack (if identity generation is a replicable process for an attacker). Pure authentication attacks are less common in TRS, but they could include the forgery of categorization credentials, so an attacker can pretend to belong to a specific category within the system. *e.g.*,a passport can pre-assign some rights to an agent based on its nationality. Thus, forgering a false credential (*i.e.*,the passport) could be considered a break into the security of the TRS system known as foreign affairs. Finally, attacks to non-repudiation in TRS include all kind of misbehavior/confusion attacks (*entities* that offer different information to different *observers*). These attack can be carried out only if the accountability of the system is inadequate.

Attacks against authorization rely on the fact that an *observer* could obtain more privileges in the *entity* than just observing specific T&R information. Sometimes, the work of the *observers* is completely passive, but if an action is needed to get this information from the entity, the *observer* can abuse this access. This kind of attacks regarding TRS has not been studied in detail in the literature.

In TRS, attacks against availability and possession are focused in this first topic: DoS attacks against the availability of *entities* and *observers*. However, there is no specific knowledge about DoS in the area of TRS. Regarding to possession, robbery attacks are included in this group. In TRS, they mean to physically steal the *entities* or *observers* that generate the trust or reputation information, but they have not been studied in detail in the TRS literature.

When an *observer* effectively accesses to the T&R information of an *entity* (i.e: the *entity* is available), the next step in our analysis lead us to study attacks against utility/process. In this process of gathering trust information, common attacks are; bad-mouthing and ballot stuffing attacks (*entities* provide false information), incomplete information attacks (*entities* do not provide all the information available), null information attacks (*entities* do not provide information at all) or even selective information attacks. This group also includes true *entities* malfunction.

Regarding confidentiality, the T&R information that *observers* obtain from *entities* can be the subject of eavesdropping attacks. Its important to mention that this analysis do not include only information, but behavior too. Both of them can be subject of unauthorized access.

Despite their importance, these attacks have not been studied in detail in the TRS literature.

Attacks against integrity or temporal integrity are uncommon when dealing with the process of gathering first-hand T&R information because *observers* usually access to it directly. Thus, attacks such as trust information manipulation or trust information re-transmission have not been studied in detail. The exception resides in attacks based on relayed information. In this case, second-hand T&R information can be manipulated. Actually, this kind of attacks are very common in different areas of knowledge such as social relationship or corporation environments.

5.4.2 Attacks against T&R calculation

In terms of the security framework detailed in the previous section, the process of T&R calculation could be described as follows: *observers* from the TRS architecture are the *agents*, the process they try to carry out is to calculate new T&R information, and to do that, the *resources* are the T&R information they previously gathered. The results of this analysis are presented in Table 5.2.

Topic	**T&R Calculation**	
Authentication	Forgery of credentials to access T&R information (*)	☆☆☆☆☆
Identification	Stolen/Clone Observer Identity (*)	☆☆☆☆☆
Non-repudiation	T&R information Authorship Rejection (*)	☆☆☆☆☆
Authorization	Calculation-based Privilege Escalation (*)	☆☆☆☆☆
Availability	DoS against observers	★★★★★
Utility/process	Initial Window, Whitewashing, Observer Malfunction	★★★☆☆
Confidentiality	T&R information Eavesdropping	☆☆☆☆☆
Integrity	Whitewashing	★☆☆☆☆
Time Integrity	T&R information delay/reuse (*)	☆☆☆☆☆

Table 5.2: TRS Attack Taxonomy. T&R Calculation

Attacks to authentication, identification, and non-repudiation are not very common dealing with T&R calculation, because this is often an internal process. *Observers* do not usually have to identify or authenticate against himself to access its own information (e.g: human beings do not need to prove their identities to access to their own memories or sensor devices to access its RAM memory, etc.). However, in some TRS, the algorithm to calculate the new values of T&R and the T&R information can reside in different *agents*. In these cases, dealing with authentication, identification and non-repudiation becomes a matter of great importance (i.g: cloud computing systems). In the same way, attacks to authorization are not common when dealing with T&R calculation. At the present time, all these topics have not been studied in detail in the TRS literature, but they are a promising field when dealing with TRS in cloud-computing or high-distributed systems.

Attacks against availability and possession are focused on DoS attacks. A DoS attack against the *observers* can frozen the T&R values, since they cannot calculate them. Attacks to possession are a threat only in the aforementioned high-distributed scenarios where T&R information and T&R calculators (i.e: observers) are in different agents.

Most of the attacks against T&R calculation are those against utility/process. For example, we can find initial window attacks or whitewashing attacks (based on exploiting the calculation algorithm). Any kind of malfunction or bug in T&R calculation, although they are not attacks, can be classified within this group.

Attacks to confidentiality consist in sniffing attacks against the T&R information used in the calculation process and inverse engineering against the calculation algorithms. Anyway, despite their importance, these attacks have not been studied in detail in the TRS literature.

Regarding attacks against integrity, the main threat is the manipulation of the T&R information used to calculate the new T&R values. This can result in a whitewashing attack (based on exploiting T&R information instead of exploiting T&R calculation). Finally, attacks

against temporal integrity can only appear in high-distributed systems, where delay or re-transmission attacks might be a real threat.

5.4.3 Attacks against T&R dissemination

In terms of the security framework detailed in the previous section, the process of disseminate T&R information could be described as follows: *disseminators* from the TRS architecture are the *agents*, the process they try to carry out is to distribute T&R information throughout the TRS, and the *resources* are the messages with that information. Due to the fact that T&R dissemination is basically a routing process, most of the attacks shown bellow are not specific of TRS, but they are important to identify all threats that can affect to a TRS. The results of this analysis are presented in Table 5.3.

Topic	**T&R Dissemination**	
Authentication	Routing credentials forgery (*)	☆☆☆☆☆
Identification	Man in the Middle, stolen identity, Clone, Sybil, Re-entry	★★★☆☆
Non-repudiation	T&R messages Routing Rejection (*)	☆☆☆☆☆
Authorization	Routing-based privilege escalation (*)	☆☆☆☆☆
Availability	DoS against disseminators	★★★★★
Utility/process	Loop, Black-Hole, Gray-Hole, Wormhole, Misrouting, Disseminator Malfunction	★★★★★
Confidentiality	T&R messages Eavesdropping	☆☆☆☆☆
Integrity	T&R messages manipulation	☆☆☆☆☆
Time Integrity	Pulse Delay, T&R message re-transmission	★★★☆☆

Table 5.3: TRS Attack Taxonomy. T&R Dissemination

Attacks related to identification rely on the fact that the identity tokens of the *disseminators* might be broken in some way. In fact, these attacks are rather similar to those regarding gathering T&R information but the subjects of the attacks are *disseminators* instead of *entities*. In this way, we can find attacks such as man in the middle attacks, stolen identity attacks, clone attacks, Sybil attacks, re-entry attack (if the *disseminator* identity generation is a replicable process for an attacker).

Authentication and non-repudiation attacks are very rare in the TRS dissemination process and we could not find any example in the literature. Anyway, these attacks could be carried out only if the accountability is inadequate.

Thus, authorization attacks are unusual because the only action that *disseminators* perform is relaying those messages. Attackers could try to carry out a privilege escalation attack against the *disseminators* through the sent messages but it is very unlikely that they could do this in most TRS. Anyway, we have to know that it can be a possible threat against our TRS.

Attacks against availability are basically DoS attacks. A DoS attack against *disseminators* can frozen the T&R values, since the calculated new T&R values are not propagated throughout the TRS. Attacks regarding possession are robbery attacks, where the T&R messages are stolen from the *disseminators*.

Most of the attacks against T&R dissemination are those against utility/process. In this group, we can find loop attacks, black hole attacks, gray-hole attacks, wormhole attacks or misrouting attacks, and, although they are not attacks, any kind of malfunction or bug in disseminators, can be classified within this group. These attacks are quite popular in TRS literature, even though they are not specific of this field.

Attacks to confidentiality are classical network eavesdropping/sniffing attacks. But, despite their importance, they have not been studied in detail in the TRS research works.

Regarding integrity, manipulation of the T&R messages is the most dangerous threat. It is a topic analyzed in depth in generic communication scenarios, but it has not attracted the attention of TRS researchers despite its importance. Finally, regarding temporal integrity, we can find two classical TRS attacks: pulse delay attacks and T&R message re-transmission attacks.

5.4.4 Taxonomy-based Analysis Conclusions

Based on this analysis, we can identify two main deficiencies of the state-of-the-art in TRS security literature. First of all, there are some potential threats that have not been previously identified, such as credential forgery to access T&R information, deny of T&R information authorship, privilege escalation attacks against all the processes of a TRS, and T&R information reuse attacks. And secondly, there are some identified attacks that have received few attention from the TRS community despite their importance, such as attacks to the confidentiality of the T&R information sources, the T&R information utilized by observers, and the T&R information disseminate throughout the TRS.

At the same time, all these deficiencies are new opportunities for TRS researchers to increase the available knowledge about security regarding TRS.

5.5 Case of study

In this section, the proposed taxonomy is applied to a real-life scenario. In this way, we can describe the benefits derived from its use, and we can validate the approach.

The selected real-life TRS environment is the Journal Citation Report, as a measure of the relevance of a scientific journal.

Journal Citation Report has been selected as our first example because it is a well-known TRS for all the research community. Besides, because of its simplicity, it allows us to clearly illustrate the use of the proposed taxonomy.

5.5.1 Journal Citation Report

As Thomson Reuters says in its website: Journal Citation Reports offers a systematic, objective means to critically evaluate the world's leading journals, with quantifiable, statistical information based on citation data. By compiling articles' cited references, JCR Web helps to measure research influence and impact at the journal and category levels, and shows the relationship between citing and cited journals.

From the viewpoint of TRS, and based on the architectural model presented in the Section 3.2, we can identify the following subjects:

- In order to simplify our model, the *entities* of the TRS will be **all** the published journals.
- JCR is a pure-reputation TRS (it does not deal with trust). Thus, there is only one *observer*, and it is Thomson ISI (Institute for Scientific Information). There is not any *disseminator*, and Thomson ISI is itself the reputation server.
- The reputation algorithm is the well-known Journal Impact Factor (IF). It is a measure of the frequency with which the average article in a journal has been cited in a given period of time. IF is calculated based on a three-year period, and can be considered to be the average number of times published papers are cited up to two years after publication. For our purpose, IF will mean reputation.

There are more agents involved in the scientific publishing market, such as publishing companies, authors, reviewers, chief editors, etc. But for our analysis we will focus only in the agent journal. There are a number of reasons for a journal to try to obtain a high IF. Anyway, we will not describe all those reasons and we will just work with the premise that this fact is true.

Now, we can analyze JCR from the viewpoint of the proposed taxonomy. In this way, we could identify all possible attacks against this specific TRS.

5.5.2 Gathering T&R information

What makes a journal different from another one is its name and its publishing company. Thus, we assume name and publishing company are the identity of a journal. In this way, we

have to check if this identification mechanism is prone to some of the aforementioned attacks. Firstly, we can see that Man in the middle attacks, confusion attacks or misbehaving attacks are unfeasible.

However, clone attacks (or semi-clone attacks) are viable: a journal can create an identity similar to the identity of an existing journal. With this technique, a journal can try to clone the reputation of the attacked journal by attracting authors to publish on it. Although it is not a fast and effective attack, it is easy to find some examples in the publishing market. A variant of this identity attack consist on creating appealing identities. This means to create journals that are susceptible to be referenced by other journals just based on their identity (i.e: their names). Journals devoted to tutorials and surveys are clear examples of these attacks.

Related to the fact that it is easy for an attacker to create new identities, we can find examples of re-entry attacks. If a journal is not well considered it can be re-created in order to clear its reputation.

Regarding to authorization and availability there is no viable attacks. Thomson can always access the required data (the references between articles) and always has enough resources to perform this process.

Regarding utility we can find the most popular and dangerous attack against JCR: it is a collaborative attack where some journals improve the IF of other journal. It is a classical ballot stuffing attack, and the only thing needed is that a set of journals artificially reference articles of a specific journal.

All information handled by Thomson to calculate the IF is public. So, we do not have to take care of confidentiality.

Regarding integrity, all the information used to calculate the reputation is eventually controlled by the editors-in-chief. This means that there is no way to ensure the integrity of the journals (i.e: integrity means that the papers published in every journal actually deserve to be published). Each editor-in-chief can decide what papers are or are not included in the journal. So, they control the information managed by Thomson to calculate the IF: they can add or remove references to others journals just by adding or removing specific articles in their editions. This is a big security hole for all the publishing system and the Thomson's IF.

Finally, regarding time integrity, we can find re-transmission attacks. An author can send the same article (or almost the same article), to different journals. If they are published, the IF is suffering a classical re-transmission attack, with duplicate information. Although this could be detrimental to the author's reputation, it is a viable attack against the JCR system.

5.5.3 T&R calculation

Regarding T&R calculation, the analysis is simpler. Due to the fact that Thomson is the only entity that calculates the IF based on public information and with its own calculation resources, authentication, identification, non-repudiation, availability, utility/process, confidentiality and temporal integrity are not subject of any of the identified attacks.

However, we are in presence of an insider attack, where the reputation server (Thomson) can decide what information will be include in the calculation of the IF and what will not. This decision can be expressed as an integrity attack: the information the reputation server handles can be deleted (not included). It might look like something that would never happen. But, due to the importance of the IF factor, Thomson can arbitrarily decide what journals are included in the JCR and what are not. And this has dramatic consequences in all the publishing market, affecting authors, publishing companies, etc.

5.5.4 Gathering T&R dissemination

Finally, attacks against T&R dissemination are not possible, because the information is distributed in a wide and public manner and subvert this information completely is an almost impossible task for any attacker.

5.5.5 JCR Analysis Conclusions

As we have demonstrated, the JCR system has many important deficiencies in the processes of gathering T&R information and calculating new reputation values (i.e: IF). In this way, we have identify re-entry attacks, clone attack, appealing-identities attacks, ballot stuffing attacks, editors-in-chief attacks, re-transmission attacks (i.e: multiple paper re-submission attacks), and excluding-journal-from-JCR attacks. Anyway, all of them have been easily detected with the proposed taxonomy.

5.6 Conclusions

Due to its nature, TRSs are especially vulnerable to attacks if the premises they are based on are subverted. There exists a high number of identified attacks against TRSs in the literature, but a generic security framework to identify all possible attacks against TRSs in a holistic way has not yet been proposed.

To achieve this goal a security framework has been presented. It is based on an augmentation of classical models such as the CIA triad and the Parkerian hexad. The topics identified in this framework are: accountability, authentication, identification, non-repudiation, authorization, availability, utility, confidentiality, integrity and time integrity.

Thus, the presented taxonomy is based on the TRS architectural model and on the security framework: in order to identify the different attacks against TRS, an analysis of all the agents and processes related to TRS is performed from the viewpoint of each topic considered within the security framework.

Based on this analysis, we can identify two main deficiencies of the state-of-the-art in TRS security literature. First of all, there are some potential threats that have not been previously identified, such as credential forgery or privilege escalation attacks. And secondly, there are some identified attacks that have received few attention from the TRS community despite their importance, such as confidentiality attacks. Anyway, these deficiencies constitute new opportunities for TRS researchers to increase the available knowledge about security regarding TRS.

Finally, the proposed taxonomy is applied to a real-life TRS: the Journal Citation Report (JCR). In this way, we can validate the benefits derived from its use. The result of this analysis is a complete list of viable attacks against this TRS.

Part II

Cases of study

6. Detection and isolation: Anomalies in Data Centers

Reliability is one of the key performance factors in Data Centers. The out-of-scale energy costs of these facilities lead Data Center operators to increase the ambient temperature of the data room to decrease cooling costs. However, increasing ambient temperature reduces the safety margins and can result in a higher number of anomalous events.

Anomalies in the Data Center need to be detected as soon as possible to optimize cooling efficiency and mitigate the harmful effects over servers. In this context, TRS can provide a significant improvement for these systems.

In order to take advantage of TRS we will follow the analysis and design methodologies proposed in Section 3.2: identify architectural entities, trust and reputation information sources, functional and non functional requirements, dissemination algorithms, etc.

This analysis allow us to choose the constitutive elements of a TRS. Therefore, we can **reduce the detection time and improve the isolation capacity** of anomalies in Data Centers.

6.1 Introduction

During the last few years, there has been a rapid increase in the number of Data Center facilities over the world. Data Centers provide the required infrastructure for a wide range of traditional applications (social and business networking, Webmail, Web search, etc.) as well as new-generation applications such as e-Health or Smart Cities. Advances in the underlying manufacturing process and hardware design technologies have continuously made possible the constant increase in computing capacities.

However, the increase in computational capabilities has not come for free. These facilities consume huge amounts of electrical power, accounting for 2% of the total USA energy budget [160]. They also generate a tremendous amount of heat that has to be extracted to ensure the reliable operation of server and other computational (IT) equipment. The energy consumption needed to cool down servers accounts for around 30% of the total energy cost of the infrastructure [161]. Even though increasing the Data Center room temperature has proven to be a way to save cooling energy, there are some important concerns regarding reliability, which is one of the key performance factors in Data Centers.

The American Association of Heating and Cooling (ASHRAE) [162] describes that the inlet temperature of servers should be kept below 30ºC to avoid CPU redlining [163]. Failures in either the room or the server cooling systems could lead to reliability issues that would reduce the Mean Time To Failure (MTTF) of IT equipment [164].

Temperature anomalies in the Data Center, as well as any other type of anomaly that might affect the reliable behavior of IT equipment, need to be detected as soon as possible to mitigate the harmful effects.

To this end, this chapter describes the usage of clustering-based outlier detection techniques coupled with a TRS to detect anomalies in Data Centers.

Clustering-based outlier detection approaches [89] offer numerous advantages for detecting insider attacks, such as high adaptability, flexibility, possibility to detect unknown attacks, no restrictions on training data, etc. Data center anomalies exhibit a similar behavior, making

clustering techniques a good candidate for their detection. Within the scope of clustering-based approaches, we encounter different deployment possibilities: *i)* k-means or k-Nearest Neighbor (k-NN) techniques, or *ii)* topology-preserving competitive methods, such as Self-organizing maps (SOM) or Growing Neural Gas (GNG). Topology preserving techniques are very convenient for our application scenario, since one of the main parameters that reveal the presence of outliers is the average distance of a cluster to its closest neighbors.

The remainder of this chapter is organized as follows: Section 6.2 describes the related work on the area of detecting anomalies in Data Centers. Section 6.3 analyzes in detail how TRS can improve the system, focusing in the typology of anomalies (Section 6.3.2), the available trust information sources (Section 6.3.3), and the algorithms proposed for this scenario (Section 6.3.4). Experimental results are shown in Section 6.4. Finally, the most important conclusions are drawn in Section 6.5.

6.2 Detecting Anomalies in Data Centers

Next-generation applications, such as the ones found in Smart Cities, e-Health, Ambient Intelligence or Weather analysis, require constantly increasing high computational demands that can only be provided in Data Centers [165], [166].

Several techniques to reduce energy consumption in Data Centers are based on increasing the supply temperature of air conditioning units to reduce cooling costs. However, increasing the inlet temperature of servers has some drawbacks. A report by the Uptime Institute [167] showed that for every 10ºC degrees of temperature in excess of 21ºC in the inlet temperature of servers, long-term reliability could be reduced by 50%.

Even though recent research [168] shows that the effect of high temperatures on reliability is smaller than what had been assumed, as the ambient temperature increases the safety margin for the server thermal shutdown is decreased. Moreover, the temperature distribution in a Data Center is not uniform and tends to have hot spots, which are areas significantly hotter than the average. To prevent server thermal shutdown, the highest CPU temperature limits the maximum Computer Room Air Conditioning (CRAC) air-supply temperature.

Thus, it is important to be able to detect and localize any anomaly taking place at the Data Center. Anomalies can be due to failures in the cooling system, in the servers, or misbehaviors in the workload assignment, that affect the thermal conditions of the server and room.

There is much research in the area of anomaly detection in Data Centers. Some approaches try to model and estimate the temperature conditions with Computational Fluid Dynamics (CFD) simulations [169]. CFD is time and cost expensive, and results are not robust to changes in the Data Center. Other works use regression models with historic data [170] or threshold-based anomaly detection [171].

All the previous techniques rely on considering static Data Center layouts. However, Data Center environments are subjected to constant changes in the placement of servers and racks. Learning and training techniques based on fuzzy control have been previously used by Sedano et.al.[172] for temperature control in buildings to maximize energy efficiency. For the particular case of Data Centers, machine learning approaches based on Neural Networks (NN) aim to find relationships between the thermal features. Other works use Self-Organizing Maps (SOM) [173] but only to discover network attacks in the Data Center, not as a methodology for anomaly detection.

6.3 TRS and Anomaly Detection in Data Centers

As we describe before, in order to improve the behavior of other anomaly detection techniques we only have to analyze this kind of systems from the point of view of the proposed TRS methodologies. Thus, we can identify the elements that are not being used by other approaches and propose the most suitable sources of trust information and trust algorithms to achieve the goals of the underlying system.

6.3.1 Underlying System Analysis

Following the structure showed in Section 3.3.7 we can identify these topics about the underlying system.

- *Description of the underlying system*. Without loss of generality and for the purposes of this work we describe a Data Center as system that is composed of a resource manager/workload allocator, a number of servers equipped with different kind of in-server sensors, and a parallel cooling system that is made up of air conditioning units and environmental sensors deployed through a WSN.
- *Requirements and goals*. Data Centers have to detect any kind of anomaly and isolate them as soon as possible to avoid any of the drawbacks described previously in this chapter. Based on section 4.3.3 this is a paradigmatic example of a **Minimization of the Response Time** and **Maximization of the Isolation Capacity**. Due to the importance of this topic, it will be discussed in 6.3.2.
- *Topology*. A number of different topologies can be implemented in a Data Center. However, due to the high connectivity between all the entities in the system it will not introduce any limitation or restriction to the design of the TRS. The only relevant issue regarding this topic is the fact that all the entities in the system are static and they will have a fixed position.
- *Timing*. There might be a global clock to trigger whole-system sensors and actuators behavior, but its presence is not compulsory. Thus, the system can be both event oriented or polling oriented.
- *Limitations*. Most, or all the entities have a permanent energy supply, and a permanent communication link to each other. The main limitations might be the computational and storage resources of some of the sensors or actuators.

6.3.2 Requirement and Goals: Taxonomy of Anomalies

In order to define our TRS architecture, therefore, it can cope with the previously defined goal of detecting and isolating anomalies in Data Center, we propose in this section an anomaly taxonomy according to their causes:

- Data room cooling: caused by failures in the cooling equipment of the data room. Their impact depends on the number of CRAC units failing and the nature of the failure.
- Server level: refers to failures in the electronic components of the servers. The effect is local to the server (i.e. thermal redlining in the CPUs). However, local effects can also have an impact on the room dynamics.
- Workload execution: workload is allocated to the computing nodes via a resource manager. Failures can be understood as tasks assigned to a certain computing node that aborted or did not complete properly. Their effect is local to a server but can be extended to the nodes absorbing the unattended demand, which might become potential hot spots.
- Information sources: caused by failures in the environmental or in-server sensors used to gather information to detect anomalies. Malfunction can come because of battery-powered sensors running out of power, environmental sensors being moved by data center operators, server sensors providing random incorrect values, etc.

A last category would be attacks on the information or networks of the data center. The scope of these attacks can be very broad, but they are generally related to gaining access to the computing nodes to retrieve sensitive information. The aim of this work is not to detect anomalies due to foreigner attacks on the data center, which falls under the area of security, but to discover anomalies inherent to the data center.

6.3.3 Trust and Reputation System Analysis

If we review the elements and processes of the proposed TRS architecture, we can identify the following ones:

- *Observers*. Every sensor, server, and the workload allocator can provide some useful information about the status of the system. Therefore, they can be an observer in the TRS.
- *Trust Information Sources*. Current Data Centers are constantly monitored by a large number of sensors to enable overall IT and cooling management. All the information gathered in the data center can be used as trust information source. Generally speaking, it can be classified as follows:
  - Environmental sensors retrieve relevant thermal characteristics of the data room. In a real-life scenario, these sensors are: *i)* temperature sensors to measure the inlet and outlet of servers, *ii)* data room relative humidity sensors, *iii)* differential pressure sensors for raised-floor air-cooled data centers and *iv)* CRAC air supply temperature sensors.
  - Integrated server sensors: these sensors are embedded in the electronics of the servers during their manufacture, and can be polled without performance overhead. The most relevant sensors are: *i)* CPU, memory and ambient temperature, *ii)* fan speed sensors, and *iii)* server power consumption sensors.
  - Server workload information: this information is obtained directly through the OS of the server (e.g. CPU and memory utilization, disk accesses, etc.).
  - Workload allocation: the resource manager provides information about the particular workload allocation to each node, i.e. number of tasks assigned, execution time, start and end time, etc.
- *Disseminators*. Every observer in the underlying system can act as a disseminator in the TRS. Due to the communication capabilities of every entity, we do not have to take any special consideration regarding the functionality or performance required by the disseminators.
- *Dissemination Protocol*. All communications in the system are sensor-to-sensor, sensor-to-server, server-to-server communications. They have place in a local network or even in a dedicated link.
- *Reputation Server*. For the purpose of this work, we do not need to specify where the Reputation Server logic has to be deployed. We just need to know that one of the high-computational-resources entity within the system will assume this role. However, the resource manager is the perfect candidate to assume this role.
- *Trust and Reputation Algorithms*. Because of the topology, complexity, and communication capabilities of the underlying system we will evaluate a combination of a local-area trust algorithm implemented by observers deployed throughout the system and a reputation algorithm implemented by the workload allocator. It will allow us to cope with both local anomalies and wide-area anomalies. Because of the special importance of this matter it will be discussed in detail in the next section.

6.3.4 Trust and Reputation Algorithms

Introduction

Most of the anomalies that take place at the Data Center have a direct impact on the thermal behavior of the data room.

Due to the fact that the anomalies demonstrate themselves as spatial and temporal inconsistencies, no matter what their source is, we find that SOM or GNG clustering techniques yield to high quality results in detecting and isolating anomalies. This theory will be validated in the section 6.4

The explanation on the next subsections applies both for SOM and GNG, as both algorithms follow the same standard steps. They only differ in the fact that the size of SOM is fixed from the start, whereas the size of GNG grows during the training. Fixed size can be a limitation, as it might not possible to know the optimal number of clusters from the start, leading GNG to perform better in some scenarios where SOM does not obtain adequate detection and isolation rates. Due to space reasons, the reader is referred to [174] and [175] for a deeper explanation on the SOM and GNG techniques used in this paper.

Feature Extraction and Model Formation

Following the idea of temporal inconsistency in the presence of anomalies, we provide the data model that captures these properties and allows us to deploy machine learning. For the case of sensed values, we follow the idea presented in [176] based on extracting n-grams and their frequencies within different time windows.

We give a short example for a boolean sensor. Let the sensor give the following output during the time window of size 20: 1 1 1 1 0 0 0 0 0 0 1 1 1 1 1 1 0 0 0 0. If we fix the n-gram size on 3, we extract all the sequences of size 3 each time moving one position forward. In this way we can observe the following sequences and the number of their occurrences within the time window: 111 - occurs 6 times, 110 - 2, 100 - 2, 000 - 6, 001 - 1, 011 - 1. Thus, we can assign them the following sequences: 111 - 0.33, 110 - 0.11, 100 - 0.11, 000 - 0.33, 001 - 0.05, 011 - 0.05.

In our model, the sequences are the features and their frequencies are the corresponding feature values. This characterization is performed in predefined time instants and takes an established amount of previous data, *e.g.*,we can perform the characterization every 20 time periods based on previous 40 values. As the extracted feature vectors are not of the same size, we calculate the distance function using the approach presented in [177], which calculates distance between sequences.

The same solution is applied to a continuous magnitude by normalizing the values to a fixed range (*e.g.*,from 0 to 5) and quantifying the sensor values to reduce the amount of n-grams without losing relevant information.

Anomaly Detection

Our goal is to detect unknown behaviors which have not been seen during the training phase, thus, we aim to detect outlying data that belongs to non-outlying clusters.

For this reason, we calculate the quantization error (QE) of each input as the distance from its group center. The deployed distance function [177] is equivalent to Manhattan distance after making the following assumption: a feature that does not exist in the first vector while exists in the second (and vice versa) actually exists but occurs with 0 frequency. In this way, we get two vectors of the same size and the distance between the center and an input is between 0 (when they are formed of the same features with the same feature values) and 2 (when the features with the values greater than 0 are completely different). Similarly, if the set of the features of one is the subset of the feature set of the other, the distance is between 0 and 1.

During the testing, n-grams not seen in the training appear when a sensor starts providing data significantly different than before. When this happens, the distance (*i.e.*,the QE value), between the n-gram and its corresponding center is greater than 1, showing evidence of abnormal behavior in the sensor or the data room.

Sensors are arranged in areas according to the events they report information about. All sensors providing information about the same observation (*e.g.*,a thermal anomaly in a certain rack or room area), are assigned to the same area. The sensors in each area are examined by one or more independent *observers*. *Observers* are trained separately and execute the clustering algorithms. The system is complemented with reputation server that assigns a value o reputation to each sensor.

For our purpose, the trust and reputation values of the sensors are used in two different ways: *i)* individual sensor trust reflects the level of confidence that other sensors have in this sensor, and is used to detect sensor malfunctioning. On the other hand, *ii)* area-wide reputation is calculated as the average trust value for a specific area, and reflects the global-spectrum anomalies occurring in the Data Center (*e.g.*,CRAC malfunctioning).

6.3.5 TRS mapping

After analyzing the components and processes involved in the design of a TRS to cope with detection of thermal anomalies in Data Center, we present a complete specification of all the decisions taken in Table 6.1.

Component/Process	**Feature**	
Underlying System		
	Goals/Requirements	Detection time and isolation capacity
	Functionality provided	Thermal Anomaly Detection System
	Timing	NR
	Topology	NR - fixed
	Limitations	Sensors: computational and storage capacity
Entities		Enviromental Sensors Server Sensors Workload Allocator
	Observed Service	Environmental Sensors: Sensed values Server Sensors: sensed values Workload Allocator: allocated workload
	Area of influence	Local
Observer	Deployed in...	Environmental Sensors, servers, workload-allocator
	Observed entities	Environmental sensors: 1 Server Sensors: n (1-5) Workload Allocator: 1
	Observation time	NR
	Range of observation	NR
	Internal vs. external	Internal
Trust Gathering		
Information	Perception	Environmental Sensors: sensed values (temperature, humidity, differential pressure) Server Sensors: CPU, memory, and ambient temperature. Fan speed. Power consumption. Workload allocator: task manager information.
	Communication	Yes
	Memory	Yes
	Categorization	No
	Reputation	No
	Nature of information	Quantitative
	Reliability	1
	Redundancy	Server Sensors: Yes
	Scope	Situational
Trust Calculation	Base algorithm	SOM, GNG
	Calculation Time	NR
	Computational Resources	Environmental Sensors: limited Server Sensors: NR Workload allocator: NR
	Nature of information	Quantitative
	Required information	Last polling

	Information consumption	No
	Scope	Sensed values
	Dynamism	Yes
	No-transitivity	No transitivity
	Asymmetry	Yes
	Histeresis Loop	Logaritmic update function
Disseminator	Deployed in...	NR
	Disseminated observers	NR
	Dissemination Range	NR
	Dissemination Time	NR
	Confidentiality	NR
	Filtering	NR
	Reliability	NR
Dissemination protocol	Base algorithm	NR
	Connection/connectionless	NR
	Point to point/broadcast	NR
	Confidentiality	NR
	Integrity	NR
Reputation Server	Deployed in...	Workload Allocator
	Nr.of reputation servers	1
	Topology	Central server
	Internal vs. External	Internal
Reputation Gathering		
Information	Trust	Yes
	Reputation	Yes
	Other sources	No
	Public vs. Private Information	Private
Reputation Calculation	Base Algorithm	SOM, GNG
	Calculation Time	NR
	Computational Resources	NR
	Observed entities	Global
	Nature of information	Quantitative
	Required information	Trust, Reputation
	Information consumption	No

(NR) Not relevant.

Table 6.1: TRS and Anomaly Detection in Data Centers: system specification.

6.4 Experimental results

In this section we show the experimental methodology used for the experiments performed to validate the approach proposed in this chapter.

All data has been collected from a data room belonging to the research group. For the purpose of this chapter, we restrict our experiments to the enterprise servers in one rack. The rack contains two types of servers, different in terms of architecture and power consumption: *i)* SunFire V20z with 2 Dual-Core AMD Opteron CPU and 4GB of RAM and *ii)* Fujitsu RX300-S6 servers with 1 Quad-Core Intel Xeon processor and 16GB of RAM. The servers are arranged in three different partitions: *i)* one containing all Intel servers, *ii)* one containing one half of the AMD servers and *iii)* a last one containing the other half of AMD servers.

All servers execute a controllable workload consisting on different tasks of the SPEC CPU 2006 benchmark [178], each requiring a different amount of CPU cores, arriving with a Poisson statistical distribution. The workload is assigned via the SLURM resource manager [179]

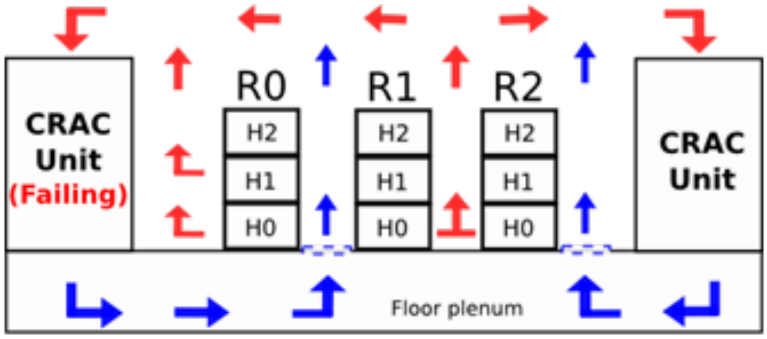


Figure 6.1: Simulated environment

that distributes workload across partitions. Thus, each partition exhibits its own workload profile. A WSN developed by the research group is deployed in the Data Center to measure the inlet and outlet temperature of all servers as well as per-server power consumption. Internal server sensors are collected via the Intelligent Platform Management Interface (IPMI) tool that enables us to obtain, for each server: CPU, memory and server ambient temperature, and average fan speed.

Our experimental setup allows full controllability on the data room environmental conditions, as well as on the workload execution, enabling the generation of normal and abnormal training and test sets, in a fully controlled way. In particular, we generate different conditions in the Data centers that lead to two different anomalies:

- Anomalies in the data room cooling due to a CRAC fan failures
- Anomalies in the workload execution.

Moreover, these anomalies take place together with anomalies in the sensing infrastructure of the Data Center, i.e. malfunctioning sensors. Anomalies are detected with the TRS simulator described in Appendix A.

The next subsections describe how each type of anomaly is generated, which are the information sources needed to detect and isolate them, and how random sensor failures can be detected within this scope. To systematize this analysis, we provide results on detection ratios, detection time, and isolation time.

6.4.1 Anomalies in the data room cooling

In our experimental setup, during the normal operation of the air conditioner, the inlet temperature of the servers varies between 16ºC to 23ºC. CRAC anomalies can be generated by suddenly turning off the air conditioning unit for a certain time.

For these experiments, we simulate a CRAC failure in a real raised-floor air-cooled real Data Center environment composed of three racks (R0, R1, R2) with servers at three heights (H0, H1, H2) that are cooled via 2 CRAC units. Figure 6.1 shows the simulated rack and CRAC distribution in the data room, and the failing CRAC unit, whereas Figure 6.2 shows the inlet and ambient temperature sensor for a server in the middle height (H1) in all three racks.

The information provided by inlet and ambient temperature sensors of servers at the same rack and height is highly correlated, comes from two different information sources (WSN and internal server sensors) and is sufficient to detect and isolate CRAC failures. We arrange data in areas according to their physical position in the data center and run TRS simulator to test the anomaly detection with SOM and GNG algorithms, both when all sensors are working properly and when some sensor malfunction exists during the testing phase.

The best results for both cases are obtained with SOM, using a training set of 300 ticks (each tick representing 1 minute) and an n-gram size of 3. Usually, n-gram size varies from 2 to 5.

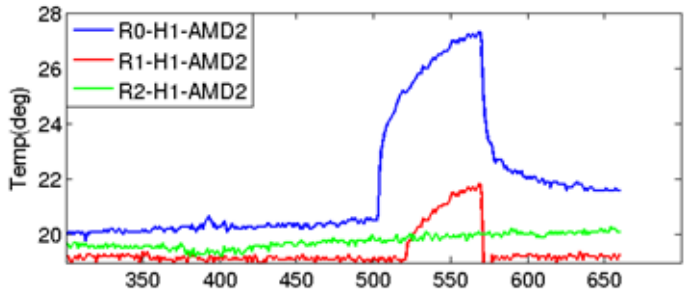
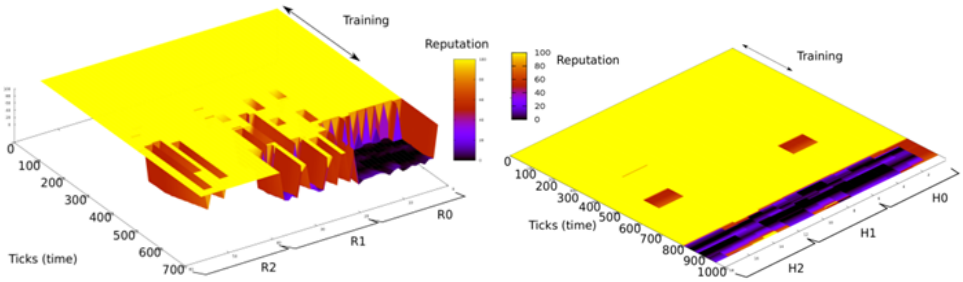


Figure 6.2: Server inlet temperature with time under CRAC failure.



(a) CRAC failure detection in three different racks (b) Sensor malfunction around ticks 550 and 600 for rack (R0)

Figure 6.3: CRAC fan failure detection and isolation with individual anomalies in sensors.

Higher n-gram sizes give more sensibility to anomaly detection but, at the same time, increase the false positive rate [84]. An n-gram size of 3, provides the best trade-off between detection and false positive rate in our setup.

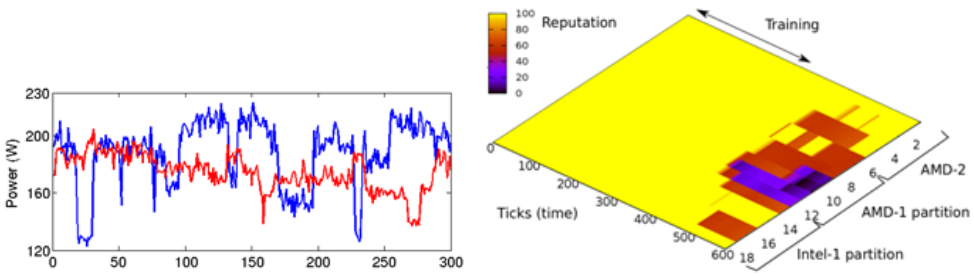
Figure 6.3(a) shows the results provided by TRS-SIM (see Appendix A) for SOM with a CRAC failure starting around tick 500 that highly affects rack 0 (R0), moderately affects rack 1 (R1) and does not affect rack 2 (R2) at all. Red and purple colors represent low reputation values and yellow color represents reputation values near 100 percent. In the horizontal axis, information source IDs are represented for the different racks are presented. CRAC-failures are calculated by averaging the reputation of sensors in the same area. If reputation is below 40, we consider that an anomaly takes place.

Figure 6.3(b) shows the malfunction of two sensors in Rack 0 (one in H2 and another in H0) around time instant 550. Regarding individual sensors, we consider that a sensor is malfunctioning when its reputation drops below 60 whereas the reputation of its neighbors if stable. Around tick 800 all sensors have a drop in their reputation. Because all sensors provide the same values, our system detects a CRAC anomaly around tick 800, instead of a sensor malfunction.

For our experiments, we obtain a CRAC failure detection rate of 100%, with a false positive rate of 0%, a very low detection and isolation time of 2 and 5 ticks respectively.

6.4.2 Anomalies in the workload execution

Detecting anomalies in the workload execution in a heterogeneous Data Center is not an easy task mainly because of the temporal variation usually exhibited by the workload. Power consumption gathered via the WSN shows different profiles depending on the workload under execution and the server architecture (AMD vs Intel, see Figure 6.4(a)). CPU temperature is correlated with power consumption and gathered via the internal server sensors, making these two metrics good candidates to detect anomalies. Because the SLURM resource manager as-



(a) Power consumption for AMD (blue) and Intel (red) nodes with time

(b) Workload misconfiguration detection

Figure 6.4: Power profile in two different architectures and workload misconfiguration detection with individual anomalies in sensors.

signs the incoming workload to three different partitions, to detect and isolate anomalies, we arrange the sensors depending on the partition they refer to.

In this case, GNG techniques with a training set of 300 ticks and an n-gram size of 2, outperform SOM in terms of false positive rate. Figure 6.4(b) shows the detection and isolation of workload anomalies in a rack composed of 9 servers belonging to the three previously described partitions. Around tick 400 servers in AMD2 partition start having an abnormal behavior that extends to more servers around tick 500. When the behavior of the server workload changes partially its reputation drops. To avoid false positives, however, we only consider that an anomaly exists when the area-wide reputation drops below 40.

For our experiments, we obtain a workload misconfiguration detection rate of 100% and again immediate detection and isolation times, as in the previous case.

6.5 Conclusions

In this chapter we have presented a clustering-based detection methodology based on SOM and GNG coupled with a TRS to detect and isolate cooling and workload anomalies.

Following the methodology proposed in Chapter 3 and Chapter 4, we selected the most suitable algorithms and trust information sources for the designed TRS in order to improve the detection and isolation times.

By making use of sensor topological information and arranging data in different areas we differentiate between individual sensor *trust* and area-wide *reputation*, splitting CRAC and workload data center anomalies from anomalies due to the malfunction of information gathering sensors.

We show how SOM provides better results for CRAC anomaly detection, yielding detection rates of 100%, in training data with malfunctioning sensors. We also show that GNG yields better detection and isolation rates for workload anomaly detection, reducing the false positive rate when compared to SOM. It is important to note the very low detection and isolation rate, that allows rapid actuation upon a Data Center anomaly.

7. Throughput Maximization: Social Odometry

The improvement of odometry systems in collective robotics remains an important challenge for several applications. Social odometry is an online social dynamic which confers the robots the possibility to learn from the others. In this context, TRS can provide a significant improvement for the coordination capabilities of this robot networks.

In order to take advantage of TRS we will follow the analysis and design methodologies proposed in Section 3.2: identify architectural entities, trust and reputation information sources, functional and non functional requirements, dissemination algorithms, etc.

This analysis allow us to choose the constitutive elements of a TRS so we can **maximize the throughput** of the underlying robot network in adverse, unsupervised and complex environments.

7.1 Introduction

Robots are individual sensors highly efficient, equipped with sufficient abilities, that can be exploited jointly. The collaborative swarm is a group of entities that work together to achieve a common objective. They make intelligent decisions to achieve a foraging goal which requires some mechanism of collaboration by means of social odometry. In social odometry, each robot is a sensor for the other robots of the swarm. The importance of social odometry lies on the fact that the swarm (the collectivity) allows the robots to collaborate to achieve a common objective because the individuals are working together.

Many robotics applications require the robots to be localized to achieve different tasks. Different solutions to the localization problem have been implemented. Among these, odometry is probably the most used as it provides easy and cheap real time position information by the integration of incremental motion information over time. Unfortunately, this integration causes an accumulation of errors during the movement of the robot, and this can be a great drawback in some robotic applications, such as foraging, where the robots have to find, select and exploit resources from unknown locations.

Different approaches have been implemented to deal with this complexity; however, those solutions have a number of different limitations: *i)* they are power consuming in terms of computation [180], [181], *ii)* some robots are not allowed to move or they have its mobility limited [182], *iii)* robots must maintain visual contact at all times with the rest of the group [183], and *iv)* in some cases robots have to communicate with a central device to update or download maps of their environment, synchronize movements, or update positions [184], [185].

Social odometry [186], [187] is a novel solution that exploits self-organized cooperation in a group of robots to reduce each individual location error.

Each robot location knowledge consists of an estimate of its own location and an associated confidence level that decreases with the distance traveled since the last know location. In order to maximize its confidence about its estimate, each individual tries to update it by using the information available in its neighborhood. Estimated locations, confidence levels and actual locations of the robots co-evolve in parallel in order to guide each robot to the correct objective.

Without loss of generality, in this chapter, we will work with a classical swarm foraging scenario: a number of resource items (usually called prey) are randomly scattered in the arena. In this context, robots search and retrieve those resource-items back to a specific place (usually

called nest). The performance of the robot network in this kind of foraging systems can be measured as either the resources-items collected by unit of time, or the time robots need to exhaust the resources.

Actually, as we said before, social odometry already uses a simple TRS based on the distance travelled. However, from the point of view of TRS techniques, foraging robot network scenarios have more valuable trust information sources that have not been used at all in previous works.

With the use of the systematic analysis, design and deployment TRS methodology proposed in previous chapters we will identify all these valuable resources and we will evaluate the performance improvements we can achieve in complex and unsupervised scenarios.

The rest of this chapter is organized as follows: section 7.2 explains how social odometry works in detail. Section 7.3 analyzes in detail how TRS can improve social odometry robot networks, and in section 7.4 we present the experimental results. Finally, in section 7.6 we draw some conclusions.

7.2 Social Odometry

7.2.1 The odometry problem

Odometry is probably the most used localization method. It provides easy and cheap real time position information through the integration of incremental motion information over time without the need for any other device.

In all the odometry techniques a travel path is derived from sensors computing the movement of the robot. However, the accuracy of odometry measurements strongly depends on the kinematics of the robot. Typical sensors for robots with a differential drive system are incremental encoders. Incremental encoders are mounted into the drive motors to count the wheel revolutions. A robot can perform odometry using simple geometry equations.

Odometry errors can be classified as either systematic or non-systematic errors [188]. Systematic errors can be modeled and corrected, while the non-systematic ones cannot be corrected and many classical techniques have been implemented to cope with them.

7.2.2 Learning from others

Social odometry is a previously defined technique [187], [189] which is not based on any map-like algorithm, and despite being inspired by the Kalman Filter [186], [190], it does not require any explicit model of the movement errors. On the contrary, a relationship between the distance traveled and a confidence level allows the robots to select the closest resource site on a foraging-like scenario.

The key aspect of social odometry is that robots within the swarm act as virtual landmarks to the others and exchange their knowledge about the position of goal areas. Nonetheless, they have to deal with two main issues: *i)* the robots only know estimated locations, not the real locations, and *ii)* the more the robots travel the worse those estimates are.

Figure 7.1 shows how information about the estimated location of area Y is transmitted from robot i to robot j. In a first step, robot i transmits its estimate of the distance dy_i and direction ϕ_i of area Y to robot j. For the direction, the value transmitted is the angle α, obtained from ϕ_i using the communication beam as reference axis: $\alpha = \phi_i - \gamma_i$. In a second step, robot j transforms the received data into its own coordinates system using simple trigonometric equations.

At this stage, robot j has the opportunity to adopt the estimate of the neighbor, to keep its own or to produce an updated location based on both. Given that estimates get worse with distance travelled, the robots use the inverse of the distance travelled as a confidence level of their estimated location. This confidence level, denoted by ϵ_i for robot i, respectively ϵ_j for robot j, is part of any communicated location and informs about the reliability, or quality, of the information.

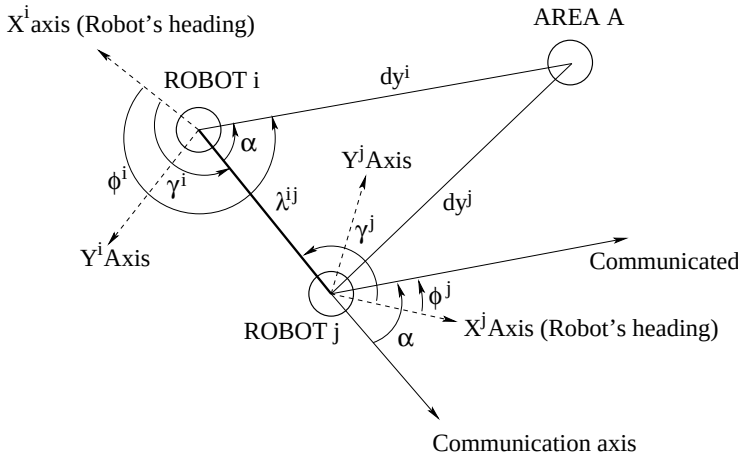


Figure 7.1: Robots sharing information about the estimated location of area Y.

7.2.3 Social Odometry equations

In social odometry, we define the state vector of the robot i at time k as:

$$\boldsymbol{x}_k^i = \begin{bmatrix} x_k^i & y_k^i & \theta_k^i \end{bmatrix}^T \tag{7.1}$$

where x_k^i and y_k^i are the robot's Cartesian coordinates and θ_k^i its orientation.

Moreover, the inverse of the confidence level (p_k^i) is defined as distance travelled by the robot (d_k^i).

Every robot keeps track of its movements and updates its *a priori* estimated location and confidence level about the different goals (i.e.: nest and prey) as:

$$\begin{aligned} \widehat{\boldsymbol{x}}_{k|k-1}^{goal,i} &= \widehat{\boldsymbol{x}}_{k-1|k-1}^{goal,i} + \Delta\widehat{\boldsymbol{x}}_k^i \\ p_{k|k-1}^{goal,i} &= p_{k-1|k-1}^{goal,i} + \Delta d_k^i \end{aligned} \tag{7.2}$$

where $\Delta\widehat{\boldsymbol{x}}_k^i$ is the state vector displacement in the time step duration and Δd_k^i is the distance travelled in the time step duration.

If there is no encounter between the robots, the *a posteriori* values are matched to the *a priori* values ($\widehat{\boldsymbol{x}}_{k|k}^{goal,i} = \widehat{\boldsymbol{x}}_{k|k-1}^{goal,i}$, $p_{k|k}^{goal,i} = p_{k|k-1}^{goal,i}$). Therefore, the confidence level decreases indefinitely. On the other hand, if two robots meet, the robots exchange information about their position and confidence level. In order to produce an *a posteriori* estimated location, each robot takes into account all information available, but weighs its sources in a different way:

$$\widehat{\boldsymbol{x}}_{k|k}^{goal,i} = \left(1 - g_k^{goal,i}\right) \widehat{\boldsymbol{x}}_{k|k-1}^{goal,i} + g_k^{goal,i} \left(\widehat{\boldsymbol{x}}_{k|k-1}^{goal,j} + \boldsymbol{x}_k^{ij}\right) \tag{7.3}$$

$$p_{k|k}^{goal,i} = \left(1 - g_k^{goal,i}\right) p_{k|k-1}^{goal,i} + g_k^{goal,i} p_{k|k-1}^{goal,j} \tag{7.4}$$

where $\boldsymbol{x}_k^{ij}$ is the vector from one robot i to robot j and g_k represents the so called pairwise comparison rule often adopted in evolutionary/social dynamics studies [191], to code the social learning dynamics, which makes use of the Fermi distribution:

$$g_k^{goal,i} = \frac{1}{1 + e^{-\beta\left(\Delta p_{k|k-1}^{goal,ij}\right)}} \tag{7.5}$$

where $\Delta p_{k|k-1}^{goal,ij} = p_{k|k-1}^{goal,i} - p_{k|k-1}^{goal,j}$ and β measures the importance of the relative confidence levels in the decision making.

Social odometry has been applied successfully and, despite the simplicity of its model, shows results comparable to more complex odometry techniques, which are more difficult to implement in real environments because of the resource and computational limitations of the robots.

7.3 TRS in a Social Odometry context

Social odometry exploits self-organized cooperation in a group of robots to reduce each individual location error using a simple and low-resources-consumption model. This allow us to use this localization technique in a wide range of real-life scenarios. If we minimize this location error without increasing the complexity order of the solution, we will be able to both improve the performance of social-odometry applications and broaden even more the range of the systems where we can apply social odometry techniques.

As we describe before, in order to improve the behavior of the basic social odometry techniques we only have to analyze this kind of systems from the point of view of the TRS methodologies. Thus, we can identify the elements are not being used by social odometry and propose the most suitable algorithms to achieve the goals of the underlying system.

7.3.1 Underlying System Analysis

Following the structure showed in Section 3.3.7 and Section 4.3.4, we can identify these topics about the underlying system.

- **Description of the underlying system**. Based on a classical ant colony behavior [187] we propose a richer and more complex scenario so we can analyze the viability of this solution in a more generic environment. The additional features are: **there are different models of robots** and is well known that they have different location performances (some models are better than others); within a specific model, **individual robots have different location performances** (but this specific performance it is not known by the other robots).

- **Requirements and goals**. Robots have to go to the source of resources (prey) and go back to the nest as many times as they can. Based on section 4.3.3 this is a paradigmatic example of a *Minimization of the System Degradation* (equivalent to maximization of the system performance).

- **Topology**. There aren't central services. The robots have full freedom of movements and all P2P communications between them are allowed if they are near enough.

- **Timing**. There isn't a global clock to trigger whole-system behaviors. So the system is event oriented.

- **Limitations**. The main limitations are based on the communication, computational and storage resources of the robots. Power consumption might be a limitation too but we won't take it into account in this work.

7.3.2 Trust and Reputation System Analysis

If we review the elements and processes of the TRS architecture proposed in Section 3.2, we can identify the following ones:

- **Observers**. Every robot in the underlying system is a sensor in the network, so it can be an observer in the TRS.

- **Trust Information Sources**. The main disadvantage of the previous social odometry approach is that it misses some of the traditional trust information sources. They use information from the *real world* (obtained by their sensors) and information from other

observers in a simple way (in the P2P robot-to-robot *communications*), however they lack for an accurate use of *memory* and *categorization*. On the one hand, *memory* is a key factor in the system. In the basic social odometry scenario robots only *remember* how long they have been walking since they found a known location. However, a model with more historical information could improve the precision of any trust algorithm. We will see how simple concepts like the global performance of the robot (total distance/number of locations found or number of round-trips done) can significantly increase the throughput of the system. On the other hand, the use of *categorization* can help us to improve the behavior of the system in the early stages. Therefore, robots can have a more accurate knowledge of the confidence level of the positions transmitted by other robots. Even when they have not already had a minimum amount of historical information (*memory*).

- **Trust Algorithm**. Because of the special importance of this matter it will be discussed in detail in the next subsection.
- **Disseminators**. Every robot in the underlying system can act as a disseminator in the TRS. *Communication* is essential in the social odometry and we will take advantage of it.
- **Dissemination Protocol**. All communications in the system are robot-to-robot communications so we do not need a complex protocol. We only have to deal with physical and link layer issues. Network layer features are not needed.
- **Reputation Server**. Because of the topology of the underlying system and its limitations there are not any global services. Thus, we will not have a reputation server for the whole system. We can evaluate if all robots or some of them can act as reputation servers, however the concept of reputation is not realistic in the defined scenario because in this kind of swarms there is not any kind of *a-priori* individual knowledge. Besides, we do not have an efficient mechanism to propagate information throughout the network. Therefore, we could not disseminate the reputation values. Anyway, future works could deal with this idea of introducing reputation servers within the system and analyzing advantages and drawbacks of this proposal.
- **Reputation Algorithm**. Based on the previous point, a reputation algorithm is not needed in this scenario.

7.3.3 The Trust Algorithm

Based on previous works in social odometry and reputation systems we will try to define the main requirements of our trust algorithm.

- In a system with entities which have different performance levels, the possibility of having an *a-priori* knowledge of this performance or a knowledge of the predictable behavior of these entities can help us improve the global performance. Kalman filters are a classical approach to this topic but they are computational expensive compared to the resources available in the robots. However, in the TRS world, this kind of knowledge is often modeled with the concept of *category*.
- We have identified the number of round trips divided by the distance travelled can be a good estimator of the individual performance of every robot in the system.
- The information exchange carried out by the social odometry approach has proved to be valid in this kind of environments. However it's limited to the transmission of *personal* information. As we explained before, one of the main trust information sources is carried out by the *disseminator*, and in a social odometry environment they do not have almost any responsibility. Besides of transmitting their own location information they can transmit trust information about previous known robots based on its individual performances. This way trust information can be disseminated faster and the whole system performance might be improved as well.

- In a foraging environment minimizing system degradation (or maximizing system performance) is the main goal. So we have to take this into account in the design and selection of our trust algorithm.

Based on all this ideas our trust algorithm will be defined in these terms:

- The inputs for our algorithm will be: the category or type of the robots in the system, so we can introduce an *a-priori* knowledge but in a simpler way than using Kalman filters; the ratio total round-trips divided by total distance, so we will have an estimate of the general individual performance; the distance travelled since the last known location, so we can keep the advantages of the social odometry approach.
- We will store these inputs so we can use this historical information.
- We will promote the trust dissemination between robots.

In order to implement the algorithm we could have used some standard trust algorithm, such as beta algorithm [192], genetic algorithms [193], self-organize maps [194], etc. However, in this environment none of them suits our requirements, tough. They are computational expensive so we decided to adapt the Fermi distribution used in social odometry so it take into account the new trust information sources: the value previously called [187] confidence level is now a function of the category information, the individual performance ratio, the inverse of the distance travelled and the historical values of this ratios.

Based on the next equations, we are going to introduce the main improvements we commented before.

First of all, we will introduce the idea of *category*. In our system there will be three kinds of robots based on the accuracy of their location sensors. Respectively, tolerance will be 2%, 5% and 10%. To introduce this concept in the algorithm we will model this tolerance as maximum errors. Therefore, the new confidence level will be weighted by this error estimation:

$$E_{category,j} = \begin{cases} 0.02, & \text{if tolerance is } \pm 2\% \\ 0.05, & \text{if tolerance is } \pm 5\% \\ 0.10, & \text{if tolerance is } \pm 10\% \end{cases} \tag{7.6}$$

$$\varepsilon'_j = \varepsilon_j * E_{category,j} = \frac{1}{d_j\,(loc)}\,(1 - E_{category,j}) \tag{7.7}$$

The next step is to introduce the idea of *memory* in the form of an estimated error. We will use the aforementioned simple ratio: total distance divided by number of round-trips.

Firstly, we define the estimated distance from nest to prey for an entity i as follows:

$$D_{NP,i} = \frac{T_{length,i}}{N_{rounds,i}} \tag{7.8}$$

Thus, the better the performance the shorter the distance.

Then, we define the estimated error of the entity j (observee) from the point of view of the entity i (observer) as given by the next equation:

$$E_{memory,ji} = \begin{cases} \frac{D_{NP,j} - D_{NP,i}}{D_{NP,i}}, & \text{if } D_{NP,j} - D_{NP,i} > 0 \\ 0, & \text{if } D_{NP,j} - D_{NP,i} <= 0 \end{cases} \tag{7.9}$$

There are two important ideas we should clarify. Firstly, we have introduced the idea of subjectivity. We remarked trust is a subjective concept but we had not yet used this fact: the confidence level now depends on the observer. Secondly, related to the equation 7.9, we only define $E_{memory,ji} \neq 0$ when the observer has a better performance than the observee. In this way, robots with worse performance cannot say that robots with better individual performances are wrong.

Finally, we can introduce this memory error ratio in our confidence level as follows:

$$\varepsilon_{j,i} = \varepsilon'_j * E_{memory,ji} = \frac{1}{d_j\,(loc)}\,(1 - E_{category,j})\,(1 - E_{memory,ji}) \tag{7.10}$$

Finally, the dissemination process does not need to be introduced in the algorithm, but in the exchanged information. If robots exchange their D_{NP} tables and their estimates of different locations, an entity can use those estimates even when it has not had a previous direct communication with other entities. However, this can introduce a significant overload both in storage and computational resources. We will analyze the effects of the trust dissemination in the next section.

7.3.4 TRS mapping

After analyzing the components and processes involved in the design of a TRS to improve the performance of social odometry foraging techniques, we present a complete specification of all the decisions taken in Table 7.1.

Component/Process	**Feature**	
Underlying System		
	Goals/Requirements	Throughput Maximization
	Functionality provided	Odometry information improvement
	Timing	Event oriented
	Topology	Ad-hoc
	Limitations	Robots: computational, communication, storage
Entities		Robots
	Observed Service	Odometry information
	Area of influence	Local
Observer	Deployed in...	Robots
	Observed entities	$[1..n)$
	Observation time	NR
	Range of observation	Local
	Internal vs. external	Internal
Trust Gathering		
Information	Perception	Odometry sensors: estimated position and angle
	Communication	Yes
	Memory	Scenario 2,3: Yes
	Categorization	Yes
	Reputation	No
	Nature of information	Quantitative
	Reliability	<1
	Redundancy	Yes
	Scope	Situational
Trust Calculation	Base algorithm	Fermi Distribution
	Calculation Time	NR
	Computational Resources	Robots: limited computational and storage resources
	Nature of information	Quantitative
	Required information	Direct perceived information
	Information consumption	No
	Scope	Odometry information (position)
	Dynamism	Yes
	No-transitivity	No transitivity
	Asymmetry	Yes

	Histeresis Loop	Fermi Distribution
Disseminator	Deployed in...	Robots
	Disseminated observers	$[1..n)$
	Dissemination Range	Local
	Dissemination Time	NR
	Confidentiality	NR
	Filtering	NR
	Reliability	NR
Dissemination protocol	Base algorithm	NR
	Connection/connectionless	NR
	Point to point/broadcast	NR
	Confidentiality	NR
	Integrity	NR
Reputation Server	Deployed in...	Pure-Trust TRS
	Nr.of reputation servers	-
	Topology	-
	Internal vs. External	-
Reputation Gathering Information	Trust	-
	Reputation	-
	Other sources	-
	Public vs. Private	-
	Information	-
Reputation Calculation	Base Algorithm	-
	Calculation Time	-
	Computational Resources	-
	Observed entities	-
	Nature of information	-
	Required information	-
	Information consumption	-

(NR) Not relevant.

Table 7.1: TRS and Social Odometry: system specification.

7.4 Experimental results

7.4.1 Simulation Tools

The proposed algorithms have been tested in simulation. We used a simulator of robot networks developed by the IRIDIA research group from Université Libre de Bruxelles. This simulation platform is a fast multi-robot simulator for the e-puck robot ([195], [196]). It has a custom rigid body physics engine, specialized to simulate only the dynamics in environments containing flat terrain, walls and holes. This restriction allows for certain optimization in the computation of the physics and, thereby, reduces the computational resources needed for running simulations (see [197] for more details).

This simulator has been combined with a high level abstraction layer based on the TRS simulator described in Appendix A.

The robot network simulator is responsible for cinematic, sensing, decision making and communication tasks, and the TRS simulator is responsible for the trust generation and management logic and provides high level information for the decision-making module of the robot network simulator.

The combination of these two specific simulators allow us to derive novel results in this area of knowledge.

In our simulations, a robot is modelled as a cylindrical body of 3.5 cm in radius that holds

8 infrared proximity sensors distributed around the body, 3 ground sensors on the lower-front part of the body and a range and bearing communication sensor. IR proximity sensors have a range of 5 cm, while the range and bearing sensor used for the communication has a range of 15 cm.

For the three types of sensors, we have sampled real robot measurements and mapped the data into the simulator. Furthermore, we added uniformly distributed noise to the samples in order to simulate effectively the different sensors. Up to $\pm 20\,\%$ noise is added to the infrared sensors and up to $\pm 30\,\%$ to the ground sensors. In the range and bearing sensor, noise is added to the range (up to ± 2.5 cm) and bearing (up to $\pm 20\,^{\circ}$) values. Moreover, each message emitted can be lost with a probability that varies linearly from 1 % when the sender-receiver distance is less than 1 cm, to 50 % when the two robots are 15 cm from each other. A differential drive system made up of two wheels is fixed to the body of the simulated robot. Errors have also been introduced into the encoder sensors chosen uniformly random in $\pm 20\%$ of the maximum movement at each time step for each wheel.

7.4.2 Simulation experiment

In this section, we compare results obtained for different social odometry experiments, with the ones obtained for the proposed TRS architecture based on all the analysis and design decisions followed in the previous sections.

Experiments have been tested in a typical foraging scenario. The selection of this scenario has been made in order to allow for comparison with previous social odometry experiments.

Based on the previous assumptions the following scenarios have been analyzed:

- *no odometry error*: robots communicate and they are not affected by odometry errors. Therefore, they navigate with a precise knowledge about the goals location ($\widehat{\boldsymbol{x}}_k^{goal,i} = \boldsymbol{x}_k^{goal,i}$, $p_k^i = 0\,;\, \forall k, i$)

- *co-variance knowledge*: robots implement a Kalman Filter to fuse their own information and the one provided by their neighbor. In these experiments, the robots need to calculate the Kalman gain every time step. Because of the comparison with previous works, all the robots assume they have the same noise on both the kinematic and communication for the Kalman Filter equations. Moreover, each robot transmits its estimated location and its own a posteriori covariance matrix when it meets with other neighbors.

- *social odometry*: robots communicate using the social odometry filter presented in Section 7.2.2. In these experiments the robots only transmit their estimated location and confidence level (inverse to the distance traveled).

- *advanced reputation system - category*: robots use the proposed TRS architecture. The trust algorithm only uses the *category* as a new trust information source as described before (based on the equation 7.7). They must transmit their estimated location, the confidence level, and a value based on the quality of their fabrication process.

- *advanced reputation system - memory*: robots use the proposed TRS architecture. The trust algorithm uses both *categorization* and *memory* as new trust information sources. Moreover, they transmit their estimated location, the confidence level, a value based on the quality of their fabrication process, and an average value of reliability based on their previous performance.

- *advanced reputation system - dissemination*: robots use the proposed TRS architecture and try to disseminate trust information to other robots. Therefore, they transmit their estimated location, the confidence level, a value based on the quality of their fabrication process, an average value of reliability based on their previous performance, and a set of average values based on previous communications with other robots.

In order to carry out detailed and realistic experiments, in all those scenarios we assume:

- There are three categories of robots with sensors with different reliability degrees. It is important to notice that typical social odometry experiments assume all the robots in the swarm are homogeneous. We have already defined in Section 7.3 that reputation systems are able to improve the swarm behavior even if the robots are heterogeneous (e.g. differences in the fabrication process). Therefore, as we said before, all the experiments presented in this section assume the swarm is made up of three categories of robots related to the fabrication process.
- Both stored data and trust dissemination messages are limited, so they do not go beyond the computational resources of the robots.

Finally, the simulations were carried out in a 3x3 m^2 and a 5x5 m^2 arenas with two marked areas (prey and nest), and 30 robots were involved in every experiment. To obtain significant statistical data, the simulations sets were performed one thousand times each.

7.4.3 Computation and communication complexity

Computation complexity

As aforementioned, covariance knowledge experiments make use of Kalman Filters. The covariance matrix $\boldsymbol{P}_{k|k-1}$ is updated based on the previous *a posteriori* estimated covariance matrix ($\boldsymbol{P}_{k-1|k-1}$) and the noise $\boldsymbol{v}_{k-1}$ through its covariance matrix $\boldsymbol{Q}_{k-1}$:

$$\widehat{\boldsymbol{x}}_{k|k-1} = f\left(\widehat{\boldsymbol{x}}_{k-1|k-1}, \boldsymbol{u}_{k-1}, 0\right) \tag{7.11}$$

$$\boldsymbol{P}_{k|k-1} = \boldsymbol{A}_k \boldsymbol{P}_{k-1|k-1} \boldsymbol{A}_k^T + \boldsymbol{V}_k \boldsymbol{Q}_{k-1} \boldsymbol{V}_k^T \tag{7.12}$$

where, $\boldsymbol{A}_k$ and $\boldsymbol{V}_k$ are the Jacobians of $f(\cdot)$ with regard to $\boldsymbol{x}_k$ and $\boldsymbol{v}_k$ respectively, and $\boldsymbol{P}_0 = 0$.

On the other hand, in the social odometry, the prediction stage is directly related to the confidence level. Since the spectral norm of the covariance matrix $\boldsymbol{P}$ grows endlessly until a communication is established or the robots arrive at one of the goals, we define the inverse of the *a priori* confidence level ($p^i_{k|k-1}$) of robot i as the distance travelled (d^i_k) since the robot left a specific area. Therefore the prediction stage for the induced covariance matrix is defined as:

$$p^i_{k|k-1} = d^i_k \tag{7.13}$$

This implementation allows the robot not to calculate the covariance matrix at each time step, and therefore to save computational time.

Moreover, in the covariance knowledge experiments, the correction stage transforms the *a priori* estimated state ($\widehat{\boldsymbol{x}}_{k|k-1}$) into the *a posteriori* estimated state $\widehat{\boldsymbol{x}}_{k|k}$. The *a posteriori* estimated state ($\widehat{\boldsymbol{x}}_{k|k}$) is adjusted in proportion to the Kalman gain ($\boldsymbol{K}_k$), which specifies the degree to which the *a priori* estimation and the measurement $\boldsymbol{z}_k$ are incorporated into the *a posteriori* state. Finally, the *a posteriori* covariance matrix $\boldsymbol{P}_{k|k}$ is also adjusted based on the Kalman gain.

$$\boldsymbol{K}_k = \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T \left(\boldsymbol{H}_k \boldsymbol{P}_{k|k-1} \boldsymbol{H}_k^T + \boldsymbol{W}_k \boldsymbol{R}_k \boldsymbol{W}_k^T\right)^{-1} \tag{7.14}$$

$$\widehat{\boldsymbol{x}}_{k|k} = \widehat{\boldsymbol{x}}_{k|k-1} + \boldsymbol{K}_k \left(\boldsymbol{z}_k - h\left(\widehat{\boldsymbol{x}}_{k|k-1}, 0\right)\right) \tag{7.15}$$

$$\boldsymbol{P}_{k|k} = \left(\boldsymbol{I} - \boldsymbol{K}_k \boldsymbol{H}_k\right) \boldsymbol{P}_{k|k-1} \tag{7.16}$$

where, $\boldsymbol{H}_k$ and $\boldsymbol{W}_k$ are the Jacobians of $h(\cdot)$ with regard to $\boldsymbol{x}_k$ and $\boldsymbol{w}_k$ respectively.

Once again, because of the simplification of the covariance knowledge on the social odometry experiments we define g as the scalar value representative to the Kalman gain:

$$g^i_k = \frac{1}{1 + e^{-\beta\left(\Delta p_{k|k-1}\right)}} \tag{7.17}$$

Experiment	Information transmitted
Covariance knowledge	$\widehat{\boldsymbol{x}}^i_{k\|k-1}, \boldsymbol{P}^i_{k\|k-1}$
Social odometry	$\widehat{\boldsymbol{x}}^i_{k\|k-1}, d^i_k$
RS category	$\widehat{\boldsymbol{x}}^i_{k\|k-1}, d^i_k, q^i_k$
RS memory	$\widehat{\boldsymbol{x}}^i_{k\|k-1}, d^i_k, q^i_k, \bar{r}^i_k$
RS dissemination	$\widehat{\boldsymbol{x}}^i_{k\|k-1}, d^i_k, q^i_k, r^s_k$

Table 7.2: Information transmitted between the robots when encounter occurs.

Hence, we use a weighed average to obtain the new location $\widehat{\boldsymbol{x}}^i_{k|k}$ and the inverse of the confidence level $p^i_{k|k}$ using the Fermi function:

$$\widehat{\boldsymbol{x}}^i_{k|k} = \left(1 - g^i_k\right) \widehat{\boldsymbol{x}}^i_{k|k-1} + g^i_k \left(\widehat{\boldsymbol{x}}^j_{k|k-1} + \boldsymbol{x}^{ij}_k\right) \tag{7.18}$$

$$p^i_{k|k} = \left(1 - g^i_k\right) p^i_{k|k-1} + g^i_k p^j_{k|k-1} \tag{7.19}$$

Therefore, it is observed that social odometry implementations are based on scalar values calculations, while covariance knowledge experiments make use of matrices.

Communication complexity

Because robots in our experiments are used as the measurement $\boldsymbol{z}_k$ to correct the estimates, the estimated state and error needs to be transferred between the robots. In all experiments, robots transmit the *a priori* estimated state ($\widehat{\boldsymbol{x}}_{k|k-1}$), but differences come up with the estimated error communication. In the covariance knowledge experiments robots need to transmit the *a priori* covariance matrix ($\boldsymbol{P}_{k|k-1}$) while in the social odometry robots only transmit scalar values. Table 7.2 shows a comparison about the information transmitted between the individuals.

A maximum of three scalar values is transmitted in all social odometry experiments, with the exception of the dissemination experiment, which depends on the size of the set which must be transmitted. However, as aforementioned, this increase in the communication load is balanced thanks to the reduction on the computation complexity.
$\widehat{\boldsymbol{x}}^i_{k|k-1}$ is the *a priori* estimated state, d^i_k is the inverse of the confidence level (distance traveled), q^i_k is the associated quality to the fabrication process, $\bar{r}^i_k$ is the average value of reliability based on their previous performance and r^s_k represents the set of average values based on previous communications with other robots.

7.5 Results and Discussion

As mentioned before, we carried out two sets of simulations based on the size of the arena (3x3 m^2 and 5x5 m^2). We have implemented the same metric used previously in social odometry experiments, time to elapse the prey, in order to allow comparison with previous works. Results are compiled in Figure 7.2 and Figure 7.3.

In the vertical axis we can see a value of performance, meaning by performance the time robots need to exhaust the resources in the prey. In order to visualize this ratio, we show it in percentage terms compared with the time robots, having no odometry errors, need to exhaust the prey.

On the other hand, in the horizontal axis, we will display a box-plot for each of the studied odometry techniques (no odometry errors, homogeneous covariance knowledge, basic social odometry, heterogeneous covariance knowledge, improved reputation model based on categorization, improved reputation model based on categorization and memory, and the complete proposed reputation model).

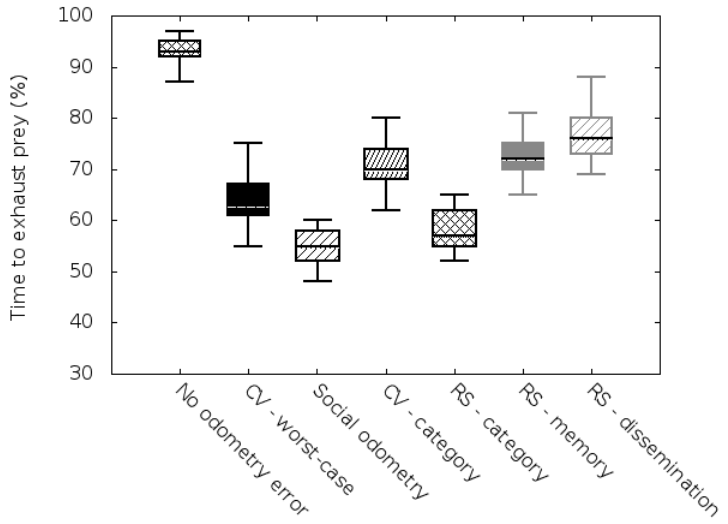


Figure 7.2: Simulation results for 3x3m^2 arena

Results of the 3x3 m^2 arena are shown in Figure 7.2. In this case, we can see the results obtained for the basic odometry scenario (no odometry errors, homogeneous covariance knowledge and social odometry) are similar to the results previously obtained in related works [187]. If we analyze the results with *category*-based-reputation system scenario (algorithm based in the formula 7.7), we can observe that the performance obtained in the basic social odometry experiment has been overcome. This difference is because category information helps robots to improve its coordination capabilities in the early stages of the simulation when the swarm is heterogeneous. However, we can see that the heterogeneous covariance knowledge performance has not been overcome by the *category*-based-reputation experiment. We should not forget that the social odometry approach is a simplification of the covariance knowledge methods.

Anyway, we can find the most important improvement when *memory* is considered and utilized as a trust information source (algorithm based in the formula 7.9). The main difference is because individual performance prevails over local situations (distance traveled since the last know location) and over general statements (categorization). This allow robots to trust more capable entities in the system and follow them as if they were leaders. In this case, the TRS *memory* experiment shows a similar performance to the heterogeneous covariance knowledge (Wilcoxon test outputs $p \approx 0.5$).

It is important to say that this is because robots use more information than in the covariance approach but the improvement is compensated with the model simplification.

Finally, if we take advantage of the trust *dissemination* feature we notice that the results are better than in the heterogeneous covariance knowledge ($p < 0.001$ in the Wilcoxon test). This is because trust information is spread faster and the effect is similar to the use of categorization but with individual information: robots obtain an *a-priori* information about the expected individual performance of other robots. Therefore, they can easily trust in the more capable individuals even without previous interactions. However, we have to remember that *dissemination* introduces a significant storage and computational resources overload. So we should evaluate robot's resources in order to know if we can incorporate this technique to our robots.

If we compare these results with the results of the 5x5 m^2 arena scenario (Figure 7.3), we can see that the reputation system approach offers even better performances. This is because the a priori knowledge (categorization) that the robots have helps them to improve their behavior in early stages and this effect is more important in wider scenarios. Without this *a-priori* knowledge robots tend to randomly walk around longer throughout the arena and the global

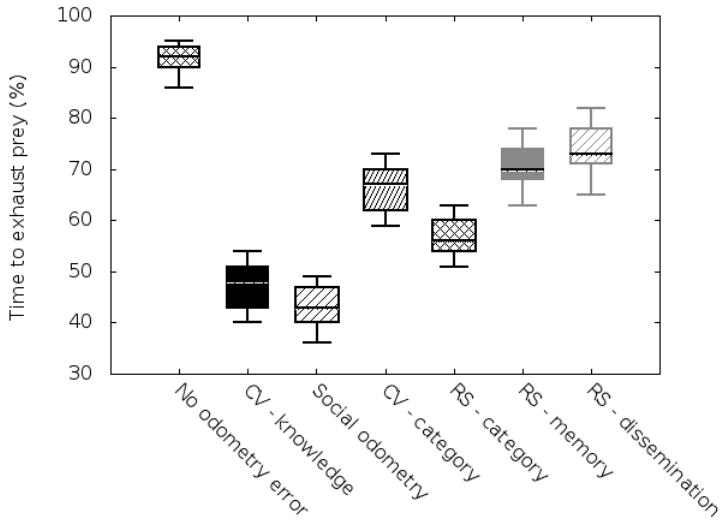


Figure 7.3: Simulation results for 5x5m^2 arena

performance gets reduced.

Notice that all the experiments, making use of the proposed TRSs, improve previous experiments done with social odometry. The main factor for this improvement is that the robots in the swarm have at hand more information than in standard social odometry algorithms. Therefore, the robots are able to generate a confidence level based, not only on their own movement as in standard social odometry, but based on the information provided by the other robots in the swarm integrated in time.

7.6 Conclusions

In this chapter we have described how a TRS can improve the performance of a complex and unsupervised scenario. In order to show it, we reviewed a novel odometry technique, social odometry, and we improved the coordination capabilities of this kind of robot networks designing a TRS that takes advantage of all the significant information sources we can find in the system.

We selected the most suitable trust algorithm and dissemination policies in order to minimize the throughput degradation that less capable robots can induce in the global behavior of the system.

To take advantage of the TRS features, we followed the TRS analysis and design methodologies previously proposed in Chapter 3 and Chapter 4. This methodologies are based on the identification of architectural entities, trust and reputation information sources, dissemination algorithms, functional and non functional requirements.

This analysis allowed us to choose the constitutive elements and the more suitable trust algorithms in order to improve the global behavior of a social odometry scenario. Simulation results quantitatively showed that the benefits of this approach were based on the use of *categorization*, *dissemination* and especially *memory*. Therefore, all of them allowed us to achieve better performances than classical odometry approaches. However, an important drawback appears with the use of *dissemination*. It requires a significant computational and storage overload in the robots, and this fact can limit its utilization in some real-life scenarios where robots have very few resources. Nonetheless, the resources required during simulation are computationally comparable to the one of the heterogeneous covariance knowledge.

8. Improving Overall Security: Wireless Mesh Networks

One of the most important problems of Wireless Mesh Networks, that is even preventing them from being used in many sensitive applications, is the lack of security. To ensure security of WMNs, two strategies need to be adopted: embedding security mechanisms into the network protocols, and developing efficient intrusion detection and reaction systems.

To date, many secure protocols have been proposed, but their role of defending attacks is very limited. In this context, TRS can provide a significant improvement for the overall security of this kind of networks. An additional advantage of this approach is that it is quite independent on the attacks, and therefore it can detect and confine new, previously unknown, attacks.

In order to take advantage of TRS we will follow the analysis, design, and securing methodologies proposed in Part I: identify architectural entities, trust and reputation information sources, security assets and threats, etc.

This analysis allow us to choose the constitutive elements of a TRS so we can **improve the security** of the underlying Mesh Network.

8.1 Introduction to WMNs

Wireless Mesh Networks (WMNs) are dynamically self-organized and self-configured, with the nodes in the network automatically establishing an ad-hoc network and maintaining the mesh connectivity. WMNs are comprised of two types of nodes: mesh routers and mesh clients. Other than the routing capability for gateway/bridge functions as in a conventional wireless router, a mesh router contains additional routing functions to support mesh networking. Through multi-hop communications, the same coverage can be achieved by a mesh router with much lower transmission power.

Mesh routers have minimal mobility and form the mesh backbone for mesh clients. Thus, although mesh clients can also work as a router for mesh networking, the hardware platform and software for them can be much simpler than those for mesh routers. For example, communication protocols for mesh clients can be light-weight, gateway or bridge functions do not exist in mesh clients, only a single wireless interface is needed in a mesh client, and so on.

In addition to mesh networking among mesh routers and mesh clients, the gateway/bridge functionality in mesh routers enables the integration of WMNs with various other networks. Consequently, instead of being another type of ad-hoc networking, WMNs diversify the capabilities of ad-hoc networks.

This feature brings many advantages to WMNs, such as low up-front cost, easy network maintenance, robustness, reliable service coverage, etc.

The main characteristics of WMNs are outlined below:

- WMNs support ad-hoc networking, and have the capability of self-forming, self-healing, and self-organization.
- WMNs are multi-hop wireless networks, but with a wireless infrastructure/backbone provided by mesh routers.

- Mesh routers have minimal mobility and perform dedicated routing and configuration, which significantly decreases the load of mesh clients and other end nodes.
- Mobility of end nodes is supported easily through the wireless infrastructure.
- Mesh routers integrate heterogeneous networks, including both wired and wireless. Thus, multiple types of network access exist in WMNs.
- Power-consumption constraints are different for mesh routers and mesh clients.
- WMNs are not stand-alone and need to be compatible and inter-operable with other networks.

Therefore, WMNs diversify the capabilities of ad-hoc networks instead of simply being another type of ad-hoc network.

These additional capabilities make them suitable for a broad range of scenarios, including security surveillance systems, spontaneous networking for fast deployment of communication facilities in case of emergencies, disasters, or military operations, community and neighborhood networking, building automation, etc.

This chapter focuses on one of the most important problems of current WMNs: security. And we'll try to offer a way to improve this security through the use of a TRS.

With the use of the systematic TRS security analysis proposed in Chapter 5 and the analysis and design methodologies presented in Chapters 3 and 4, we will identify most common attacks against WMN and describe how TRS can cope whit this issues.

The rest of this chapter is organized as follows: Section 8.2 explains the main attacks and countermeasure in WMNs. Section 8.3 analyzes in detail how TRS can improve WMN security. In section 8.4 we present the experimental results. Finally, in section 8.5 we draw some conclusions.

8.2 Attacks and countermeasures in WMNs

In this section, we will follow the taxonomy defined in Section 5.4 to analyze the different kinds of attacks that a WMN is exposed to, and to present the main countermeasures found in the literature.

8.2.1 Authentication/Identity attacks

Malicious nodes can pretend to be other nodes. In this area we can find four main different types of attacks:

- *Clone* It consists in duplicating a legal node. Both nodes, simultaneously, communicate with the same identity.
- *Thief* A malicious node steals the identity from an operating node and replaces it in the network. The malicious node stops original node's operation.
- *Mole* A mole is a malicious node that behaves as a well-operating node. Once inside, it can attack the system from a privileged position. A variation is the *on-off* attack, where the malicious node behaves well and badly alternatively.
- *Sybil* It occurs when a malicious device presents multiple identities, as if it were multiple nodes, in order to control a substantial fraction of the system. The attacks can be performed at any layer of the protocol stack, but they are more profitable in the upper layers, like network or application.

The clone, thief and mole attacks are carried out by individual malicious nodes, and they can be considered special cases of the Sybil attack. The Sybil attack was first introduced in [198]. [199], [200] and [201] make thorough descriptions of the taxonomy, threats and countermeasures of identity attacks, focusing on the Sybil attack.

In the literature we can find three main types of solutions to the identity attacks against WMN: resource testing, cryptography and location-based.

- *Resource testing solutions*: they assume that devices are limited in some resource [198]. The solutions consist in testing a limited resource and checking that each identity has no less capability than a physical node. The resource tested in wireless networks, according to [199], is the radio communication capability, considering that a device can access only to one radio channel at a time. Each identity has a channel assigned and they must send a message through it simultaneously. The system detects an identity of a Sybil attack when it receives no message in its channel. Accurate synchronization between the monitoring devices is needed and, if we have more identities than channels, we can't perform the test to every identity at the same time, so the detection rate decreases.

- *Cryptography schemes*: they base their efficiency in secure communications, and the different solutions differ in how to establish the keys: the key agreement process. They can have a key server with the public key of all nodes, and only establish a key through the key server. Another scheme uses the self-enforcing scheme approach, based on asymmetric cryptography with public key. Efficient implementations of Elliptic Curve Cryptography (ECC) Cipher Suites can be used in WMNs to establish secure links, but it is not enough to avoid the Sybil attack, because a malicious device may have more resources than the normal nodes. The third key agreement mechanism is key pre-distribution scheme [202]–[204]. In these systems each node has a subset of the system keys and a secure link is established between nodes which have at least one key in common. If a node is compromised, several keys are known by the malicious device. If more nodes are compromised, the attackers can obtain a substantial fraction of the system keys.

- *Location based solutions* [205], [206]: they check that no identities are at the same position. The solutions assume that the nodes are static, but real WMN applications have heterogeneous networks, with static and mobile nodes.

8.2.2 Availability attacks

Availability attacks try to alter the normal behavior of the system by interrupting, disrupting, or destroying services and operations in a system.

The main kind of attacks in this area are **jamming, collision, and flooding attacks**. These attacks consist in interfering in communication by sending messages through several protocol layers. The immediate effect of these attacks is the loss of part of the messages from the nodes of the affected area. The affected area depends on the layer in which it occurs. The upper the attack occurs on the protocol stack, the more it spreads. [207] propose several countermeasures for these attacks: they suggest confinement, small frames, error-correcting codes and client puzzles.

8.2.3 Utility attacks

Utility attacks try to alter the normal behavior of the system by modifying the behavior hf mesh clients and routers. As we described in Section 5.3, some topics derive from this type of attack: process attacks, confidentiality attacks, and integrity attacks.

Process attacks

In this area we can find three main different types of attacks related to WMNs:

- *Neglect and greed* This simple form of DoS attack focus on a router vulnerability by arbitrarily ignoring all or some messages. It is especially dangerous in environments using hierarchical routes and static routing protocols. A possible solution could be a routing protocol with several paths available [207].
- *Blackhole* [208] While receiving routing requests, the attacker claims to have a link to the destination node, forces the source to send packets through it without forwarding them to the next hop.
- *Wormholes* [208] Two distant points in the network are connected by a malicious connection using a low-latency link called the wormhole link. Once the wormhole link is established, the attacker captures wireless transmissions on one end, and replays them on the other end. It can be used to control the routing behavior at the attacker's will.

These attacks are very difficult to avoid, detect and confine. Authorization and monitoring have been proposed to avoid them. However, it is not possible to deploy a secure WMN based exclusively on ciphering and authorization. It is necessary to supply additional techniques to reinforce the system. There exist some countermeasures consisting on enhanced protocols [209], however they require too many resources to be used in low-end nodes.

Confidentiality attacks

Confidentiality attacks attempt to access to the information stored in the network. In the case of WMN they can be further classified attending to the target of the attack into attacks on the confidentiality of communications, and attacks on the confidentiality of node information.

The network can use well-suited cipher algorithms [210] to provide security against attacks to communications. But WMN nodes are vulnerable to confidentiality attacks due to their characteristics:

1. Nodes have limited resources (both mesh clients and mesh routers).
2. Potential intruders may physically access to them.

It is difficult to achieve a high degree of confidentiality in low-end devices when physical security can not be guaranteed, and therefore it is better to minimize the amount of confidential information that these nodes store and process. Some approaches suggest ciphering stored data [211]. Nevertheless, a combination of logical (cryptography weakness and Trojan horses), and physical (DPA, SPA, micro-probing, reverse engineering) attacks could break the ciphering and access the information.

Integrity attacks

- *Tampering* Even if data is encrypted, mesh routers can modify any specific field in the packets while forwarding them, resulting in wrong routing decisions like redirections or route loops, which degrade the network performance. The lack of integrity checks is the root of most of these vulnerabilities.
- *Forging or misdirections* [208] An attacker can forge and broadcast wrong routing information, such as declaring some certain link is broken, or replying with a non-existing route. This might cause serious problems like loops or isolated networks.

8.2.4 WMN countermeasures: Secure routing protocols

Most of the countermeasures against attacks are based on more-or-less secure extensions of current MANET routing protocols, such as DSR, AODV, and DSDV.

SRP [212] extends current on-demand routing protocols with the ability of identifying and discarding false routing information, and avoids tampering, wormholes and forging attacks. But it depends on a shared key for verification and communication. Ariadne [213] is another

routing protocol based on DSR, using the TESLA technology. TESLA is a broadcast verification mechanism based on time synchronization and delayed key exchanging. It also depends on a shared key.

ARAN [214] uses public key certificates and a trusted CA to verify the routing information. SAODV [215] extends AODV with digital signatures and one-way hash chains to ensure packet integrity. SLSP [216] avoids tampering attacks by means of asymmetric cryptography, and it also avoids flooding by not processing packets coming from a node whose message frequency is now much higher than usual. Both, SAODV and SLSP require a lot of resources due to their usage of asymmetric cryptography.

8.2.5 WMN Security Challenges

Despite the usual resource constraints of WMN nodes and their physical accessibility, most of the countermeasures presented to date are based on secrets, shared or not. And there are many low-cost techniques to attack these systems in order to reveal the secret keys.

To ensure security of WMNs, two strategies need to be adopted. Either to embed security mechanisms into network protocols such as those presented in the previous section, or to develop security monitoring and response systems to detect attacks, monitor service disruption, and respond quickly to attacks, by isolating the compromised nodes as much as possible. To date, many secure protocols have been proposed, but their role of defending attacks is very limited, because schemes located in a single protocol layer cannot solve problems in other layers. However, security attacks in a network may come simultaneously from different protocol layers.

In this chapter we present a framework for improving the overall security of any WMN based on TRS and that is orthogonal to the network protocols. This approach will allow us to **detect and isolate ill-behaved nodes** by rating their *trust* as low based on unsupervised and distributed algorithms. An additional advantage of this approach is that it is quite independent on the attacks, and therefore it can detect and confine even new, previously unknown, attacks.

8.3 Improving WMN security with TRS

As we describe before, in order to improve the security of WMN by using TRS we only have to apply the methodologies described in 3.3. So, we can propose the most suitable architecture, the sources of trust information, and the trust algorithms to achieve the goal of preventing the system from the attacks described in Section 8.2

The first step of the proposed methodology is based on describing the underlying system (*i.e.*,WMN), identifying their main goals and requirements (*i.e.*,improve the overall security), and analyzing its topology, timing and limitations. In this case, all these topics have been already detailed.

So, we will focus on the analysis of the TRS elements and processes related to apply them to a WMN environment.

8.3.1 Trust and Reputation System Analysis

As shown in [217], we envisage a TRS architecture where each node participates in the TRS by assigning low trust to the nodes that behave suspiciously and vice versa.

Our approach to improve the security of WMN is based on avoiding any contact, either information exchange or usage as a routing hop, with the nodes that have low trust. In that way, the suspicious nodes will remain isolated from the network.

The proposed TRS architecture is based on the node-to-node interaction. On one hand, an optimal next-hope choice can lead to an improvement of both performance and security. On the other hand, a global behavior optimization is not viable because the dynamism and variability of WMN. So, we will focus on defining a *pure-trust TRS*, were all improvements

provided by the TRS to the underlying system come from the definition and optimal utilization of the local and subjective concept of *trust*, avoiding the use of the global-scope concept of *reputation*.

- *Observers*. Every node in the underlying system needs to exchange information and because of that, they can evaluate the performance of the overall network when exchanging this information based on the routing decisions taken. Therefore, all the nodes can be an *observers* in the TRS.

  To deal with the energy consumption limitations of the underlying system, even though each node will become an *observer*, most of them will be inactive. The active ones will execute the algorithm explained bellow. Considering that it takes constant retraining, and that the process consumes lots of resources, we propose that this process executes only when connected to the supply. In this way, our TRS does not affect significantly on power consumption. Anyway, the inactive *observers* can change to an active state in order to ensure the service in the system.

- *Trust Information Sources and Trust Algorithm*. Because of the special importance of this matters, they will be discussed in detail in the next subsections.

- *Disseminators*. Every node in the underlying system can act as a disseminator in the TRS. However, because we're dealing only with *trust* information, and this has a local meaning, the dissemination process do not have a relevant role in the system.

- *Dissemination Protocol*. We do not implement any new protocol to exchange trust information. All nodes in the system can use their default protocols to exchange this kind of information.

- *Reputation Server*. Because of the proposed architecture based on the topology of the underlying system and its limitations there are not any global services, so we will not have a reputation server for the whole system.

- *Reputation Algorithm*. Based on the previous point, a reputation algorithm is not needed in this scenario.

8.3.2 The Trust Information Sources

One of the main goals when designing an efficient TRS for improving the security of any underlying system is to define a set of significant features that accurately capture and distinguish the representative behaviors of both normal and intrusive activities.

Given that there are groups of attacks that target different assets of the network, we believe that the proper way to proceed is to establish different models that will address different vulnerabilities.

Therefore, according to the attack types presented in 8.2, we can distinguish two main different behavior models that cover all given attack scenarios where our integrated architecture can improve WMN security.

- *Routing Behavior*: within this behavior model we take account of the most significant routing parameters which will allow us to analyze normal routes and detect when unusual ones occurs (against wormhole or blackhole attacks, etc.). We also inspect the behavior of neighbor nodes (average packet arrival rate, number of messages dropped by the node etc.), the variance of the routes, etc.

- *Resource Utilization*: within this behavior model we take account of the values of communication channels utilization and node parameters such us memory or CPU utilization. This will allow us detect denial of service and distributed denial of service (DoS/DDoS) attacks and resource depletion attacks in general.

With these two models we can train and characterize our TRS in order to improve the detection and response against these kinds of attacks as we are going to analyze in the next section.

It's worth mentioning that, within these models, the concept trust turns into new concepts such as normal routing behavior and normal resource utilization but they are completely equivalent, in a computational sense, to the original trust concept.

Anyway, our models are based on two important assumptions:

1. The adversary can capture only a limited number of nodes in the WMN, which means that most of the resource utilization and routing behavior produced by the nodes is normal.

2. Resource utilization and routing behavior produced under the influence of an adversary are statistically different from the output produced during the normal operation of the network. For this reason, we establish the detection of anomalies in data/behavior as outlier detection[1]

8.3.3 The Trust Algorithm

Based on the model described in the previous section, we propose two different SOM algorithm setups that capture the routing behavior and resource utilization of every node in the WMN.

They are based on the definition of the following vectors for capturing the behavior of different aspects of WMNs:

- *Routing Behavior*: this vector will depend on the deployed routing protocol. In general, we can express it in the following way:

$$rout\_beh\_vect = [rout\_par_1, rout\_par_2, ...rout\_par_n]$$

  where $rout\_par_i$ are the significant parameters of the routing protocol.

  Moreover, for each node we add two more characteristics: average packet arrival rate and number of dropped messages by the node.

- *Resource Utilization*

  In order to capture normal behavior of each node in the terms of resource utilization, we establish the following vector:

$$res\_beh\_vect = [mem\_utilCPU\_utilI/O\_util]$$

  where the characteristics are the percentage of memory utilization, percentage of CPU utilization and percentage of I/O utilization correspondingly.

In addition, based on the previous definitions presented is Section 6.3.4, we calculate the average distance of each cluster to the rest of the clusters (or its closest neighborhood) (MD). Finally, we calculate quantization error (QE) of each input as the distance from its corresponding cluster center.

Then, we calculate the average distance of each cluster to the rest of the clusters (or its closest neighborhood) (MD). And we calculate quantization error (QE) of each input as the distance from its corresponding cluster center.

Based on the previous definitions of anomaly index (Section 6.3.4), we define the *trust* of every node in the following way:

[1]The definition of outliers is rather fuzzy, but it is considered that an outlier is an observation that lies an abnormal distance from other values in a random sample from a population, in other words extreme points in data cloud.

1. We limit the *trust* values to the range $[0, 1]$, where 0 is the lowest possible, meaning that there is no confidence in the node, and 1 the highest possible, meaning the absolute confidence in the node.

2. We define two *trust* values, $trustQE$ and $trustMD$ based on previously defined QE and MD values:
$$trustMD = \frac{(maxMD_{value} - anoScMed)}{maxMD_{value}}$$
where $maxMD_{value}$ is the maximum median distance for the current lattice and $anoScMed$ is the MD value for the best matching unit of the current input. In this way, $trustMD$ takes values between 0 and 1, where the nodes that are close to the rest (or its proximate vicinity, depending on the definition) have higher *trust* and vice versa.

Regarding QE value, during the training we calculate the median QE for all the nodes in the corresponding SOM lattice. In the testing process, we calculate QE value for the corresponding input and calculate $trustQE$ as the ratio of current QE and the median QE for its corresponding best matching unit node.

If the data produced by the presence of an intruder form their own group, it will be significantly distant from the rest. On the other hand, if this data is too sparse and it is not able to form their own group, it will end up belonging to the normal nodes.

Based on these premises, we establish the following manner to calculate what we call t_{ti} as *temporal current trust* for the node i, at the time t:

$$t_{ti} = \begin{cases} trustMD_{ti}, & if\ trustMD_{ti} < k \\ trustQE_{ti}, & if\ trustMD_{ti} \geqslant k \end{cases} \tag{8.1}$$

where k takes the value of 0.5 as threshold. This value has been validated through experimental results.

Finally, we update the *trust* of the entity i in the following way:

$$T_{ti} = T_{t_{-1}i} + t_{ti} + \log(m * t_{ti})$$

If the final value is greater than 1, we truncate it to 1, and in a similar fashion, if it is lower than 0, we truncate it to 0. The function $x + log(m * x)$ is presented in figure 8.1 with $m = 0.99$.

It provides exactly what we want to achieve: falling of the cumulative *trust* if we have small current *trust* values and vice versa, and also small changes in the *trust* if we are around 0.5. As it can be observed, for the values lower than 0.3 the *trust* will fall down quickly, while for the values higher than 0.65 the function rises significantly. Finally, for the values between 0.5 and 0.65 the *trust* changes in small amounts.

8.3.4 TRS mapping

After analyzing the components and processes involved in the design of a TRS to improve the security of WMN against routing and behavior atttacks, we present a complete specification of all the decisions taken in Table 8.1.

Component/Process	Feature	
Underlying System		
	Goals/Requirements	Detection time and isolation capacity
	Functionality provided	Detection and isolation of routing and resources availability attacks
	Timing	Event oriented
	Topology	Mesh Network (ad-hoc)
	Limitations	Mesh clients: computational resources, power consumption

Entities		Mesh clients, mesh routers
	Observed Service	Mesh clients: use of resources Mesh routers: use of resources and routing behavior
	Area of influence	Local
Observer	Deployed in...	Mesh Clients, Mesh Routers
	Observed entities	Mesh clients close to the Mesh Router
	Observation time	NR
	Range of observation	Mesh communication range
	Internal vs. external	Internal
Trust Gathering Information	Perception	Mesh Clients:<mem_util,CPU_util,I/O util> Mesh Routers:<mem_util,CPU_util,I/O util> and <route_path,dropped msg,avg.pkt.arrival>
	Communication	Yes
	Memory	Yes
	Categorization	No
	Reputation	No
	Nature of information	Quantitative - feature extraction
	Reliability	1
	Redundancy	Yes
	Scope	Situational
Trust Calculation	Base algorithm	SOM
	Calculation Time	NR
	Computational Resources	Mesh Clients: limited computational resources
	Nature of information	Quantitative
	Required information	Perceived and communicated information
	Information consumption	No
	Scope	Mesh clients: use of resources Mesh routers: use of resources and routing behavior
	Dynamism	Yes
	No-transitivity	No transitivity
	Asymmetry	Yes
	Histeresis Loop	Logaritmic update function
Disseminator	Deployed in...	Mesh routers
	Disseminated observers	m
	Dissemination Range	WMN deployment dependent
	Dissemination Time	NR
	Confidentiality	Not guaranteed
	Filtering	Yes
	Reliability	<1
Dissemination protocol	Base algorithm	WMN communication protocol
	Connection/connectionless	NR
	Point to point/broadcast	NR
	Confidentiality	Not guaranteed
	Integrity	Not guaranteed
Reputation Server	Deployed in...	Pure-Trust TRS
	Nr.of reputation servers	-
	Topology	-
	Internal vs. External	-
Reputation Gathering Information	Trust	-

	Reputation	-
	Other sources	-
	Public vs. Private information	-
Reputation Calculation	Base Algorithm	-
	Calculation Time	-
	Computational Resources	-
	Observed entities	-
	Nature of information	-
	Required information	-
	Information consumption	-

(NR) Not relevant.

Table 8.1: TRS and security in WMN: system specification.

8.4 Experimental results

The proposed architecture has been simulated extensively to evaluate its behavior in presence of attacks of very different nature.

In order to systematically analyze the attacks in the proposed scenarios, we will use the following characteristics which let us measure the performance and the effectiveness of our approach:

- *Detection time*. It is the elapsed time since the attack started until it is detected, *i.e.*,the ill-behave node trust begin decreasing.
- *Isolation time*. It is the elapsed time since the attack is detected until the *trust* of every attacker node gets below a threshold.
- *Isolation capacity*. It is the portion of ill-behaved nodes that are detected as attackers.
- *System degradation*. It is the portion of well-behaved nodes detected as attackers.

All the figures in the next sections show the evolution of the concept of *trust* previously defined of every node in the system after and before introducing the attack.

We use a 2D representation where the Y-axis represents time whereas X-axis indicates space. In order to clarify the representation of the results, we choose a representative one-dimension-space from the whole two-dimensions-space where the system is deployed. The color gradation associated to every 2D-point shows the *trust* values under these conditions of time and space.

8.4.1 The Redundancy problem

When we try to apply TRS techniques in a WMN scenario, the main difficulty might be the density of nodes in these scenarios.

Other scenarios previously presented such as those detailed in Chapter 6 and Chapter 7 are node-massive systems but WMN are not. So, it is really important for us to know if this difference can be an insurmountable obstacle in the application of TRS techniques to WMN systems.

Before starting to analyze the response of our proposed architecture against different kinds of attacks, we have to evaluate the importance of the node redundancy in the behavior of the TRS.

In order to solve this question we present a comparative analysis which shows the impact of a standard attack based on the trust algorithms used and different levels of redundancy.

As we can see in figure 8.2, the SOM algorithm needs much less redundancy to work properly compared to linear or beta algorithms. The x-axis represents the number of nodes per

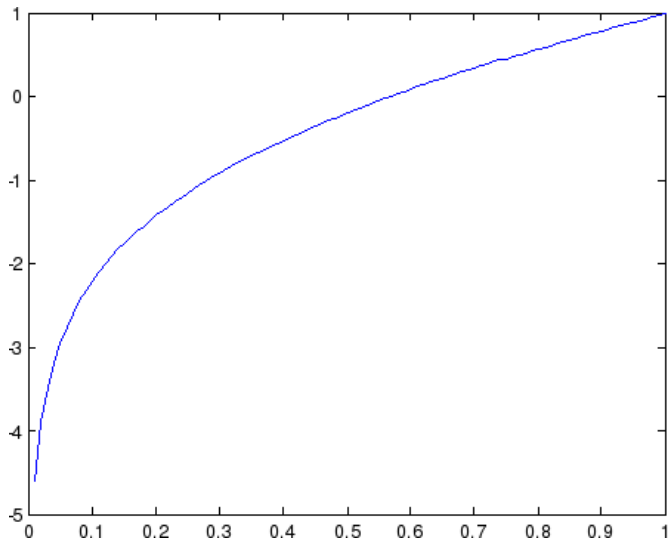


Figure 8.1: Function for updating trust values

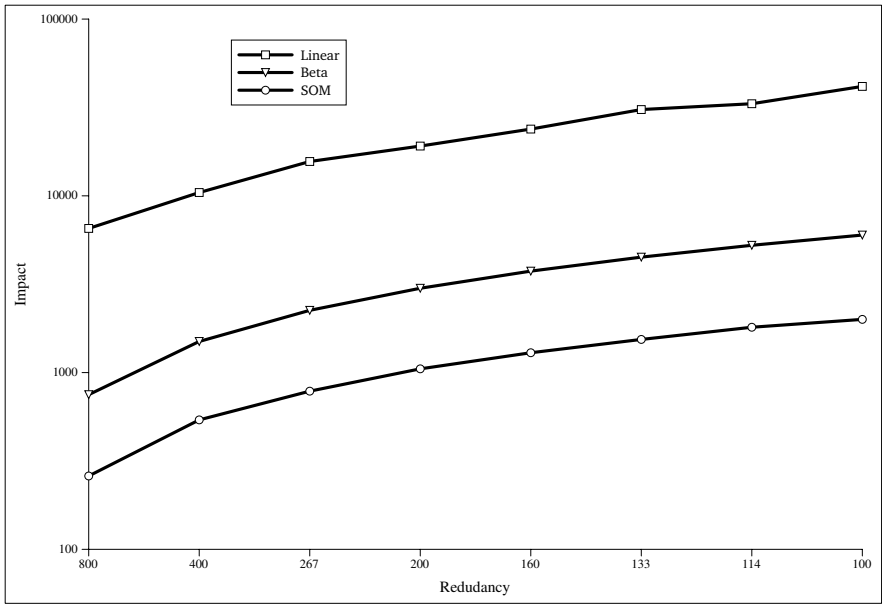


Figure 8.2: Evolution of the impact of the attack and the based on the node redundancy.

each one hundred attacker (n). The y-axis indicates the impact, calculated as the sum of the false positives (good nodes identified as malicious) given by the formula 8.2.

$$I(n) = \sum_{t=0}^{T_s} P_{fp_t}(n) \tag{8.2}$$

where T_s is the simulation time and P_{fp_t} is the percentage of false positives at the instant t.

These results have a number of very clear implications:

- Even if we are using a TRS in a WSN scenario, the selection of the trust algorithm makes a basic difference in terms of the impact of the attack. SOM becomes the best option based on these parameters.
- The dependency of the impact with the redundancy of the system is lower if we use SOM algorithms instead of linear or beta algorithms. This allows the system to scale better and be used in scattered deployments without suffering a severe degradation.
- We can obtain an adequate performance with a lower number of nodes in the system, which allow us to deploy the system faster and with a lower cost. This a key feature in order to widely use WMN in real scenarios.

8.4.2 Routing-behavior attacks

In this section we are going to analyze the behavior of our proposed architecture against several attacks related to modification of routing information: routing paths, routing behavior, etc.

Wormhole attack

In this experiment a wormhole has been introduced in the system in order to evaluate the response of the TRS in presence of data with inconsistent paths in its routing information. Some data from a node is stolen and seems to be generated in a different location.

In the experiment a routing node is attacked. Thus, the attack not only affects this specific node but the attack can reduce the *trust* of other nodes being routed by this one.

The results shown in figure 9.5 have been obtained by simulating a scenario of 100 nodes where one routing node is attacked, so that the TRS receives data with this attacked node in its path table from two different areas. The system is working normally until the 100th iteration, when the wormhole attack is launched. The routing algorithm is a custom multi-path AODV based on [218]. The network traffic due to trust information being sent to the nodes is about one tenth of the total traffic.

The system quickly detects an inconsistent behavior related to the attacked node and the nodes being routed through it, but initially, it cannot exactly determine the source of the anomaly. Therefore, the system reduces the *trust* of all the involved nodes. This reduction and the following data allow it to identify the source of the attack, and then it isolates the damaged routing node.

There are two areas with a reduced *trust* because the TRS cannot identify which one is the original node. Anyway, the system learns that this node id is not a good routing node and their neighbors shouldn't use them. This fact is pointed out by the reduction of *trust*.

8.4.3 Resources availability attacks

In this section we are going to analyze the behavior of our proposed architecture against several attacks related to a excessive use of the resources of the devices within the system and related to a excessive utilization of the communication resources of the system.

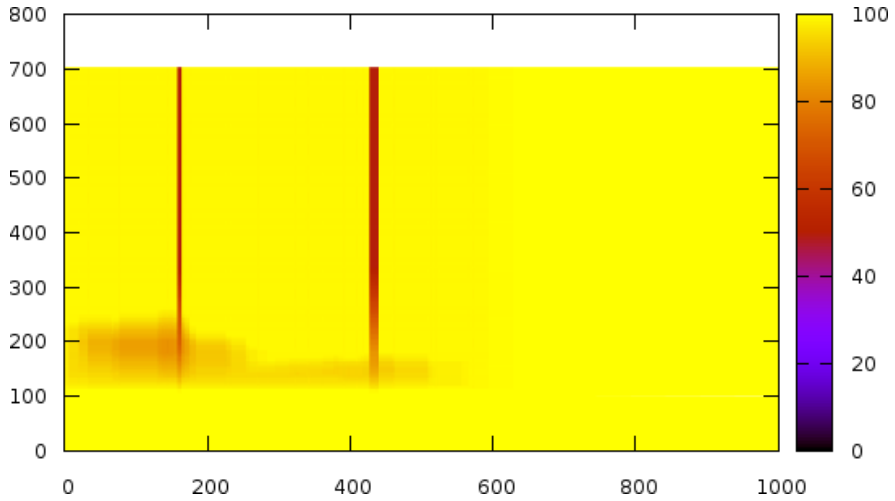


Figure 8.3: Trust evolution for a wormhole attack

DoS attack

In this experiment a DoS has been introduced in the system in order to evaluate the response of the TRS in presence of a excessive use of the communication resources.

This attack covers a number of situations such as an improper operation of a node, a overused routing node or, of course, a deliberate DoS attack.

The results shown in figure 8.4 have been obtained by simulating a scenario of 100 nodes where one node is attacked so that it receives 20 times more traffic than in its normal operation. At this moment, the attacked node starts to overuse its resources, flooding the system with non-useful data. The system is working normally until the 80th iteration, when the DoS attack is launched.

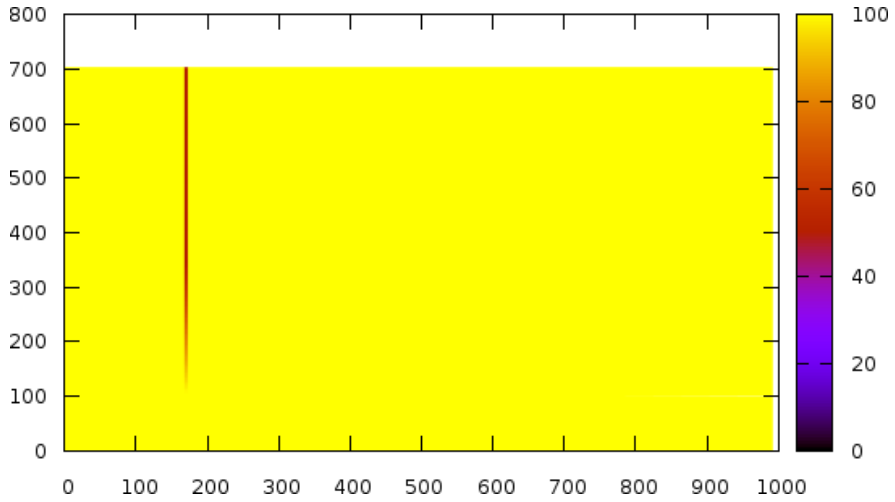


Figure 8.4: Trust evolution for a DoS attack.

As we can see, the TRS identifies immediately the attacked node and informs the neighbor nodes so they do not have to accept traffic from the attacked node. The detection and isolation times are short enough so that other nodes in the system are almost not affected by this excess of traffic.

In order to analyze the TRS in a more severe scenario we have simulated a new DoS where the traffic injected is much higher than before.

The results shown in figure 8.5 have been obtained by simulating a scenario of 100 nodes where one node is attacked so that it receives 100 times more traffic than in its normal operation. We assume that this node have enough computational and communication resources to process this traffic. In this moment the attacked node starts to overuse its resources flooding the system with not useful data. The system is working normally until the 100th iteration, when the DoS attack is launched.

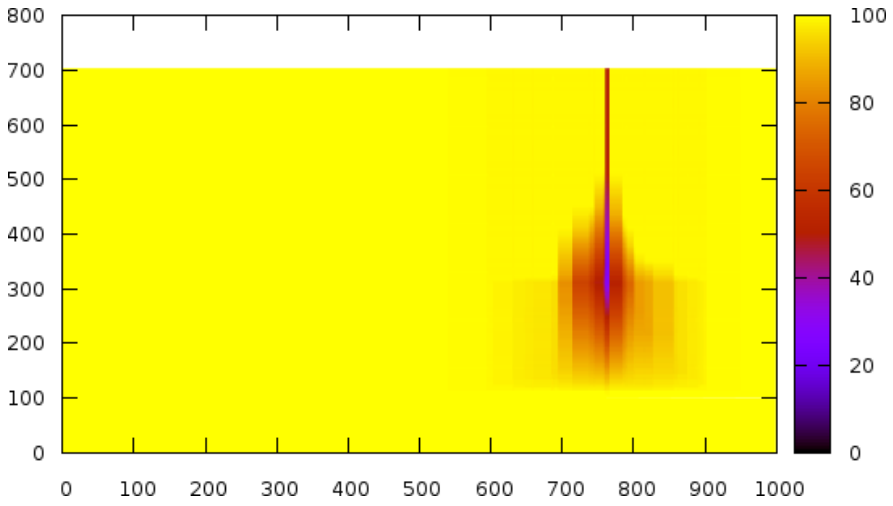


Figure 8.5: Trust evolution for a severe DoS attack.

As we can see, the TRS identifies immediately the attacked node but it is not fast enough to prevent the neighbor nodes from being affected by the attack. The TRS identifies an unusual behavior in these nodes and reduces their *trust* so other nodes reduce the traffic accepted and sent to these affected nodes. This allows the trust algorithms to delimit the source of the attack and in the long term the attack is confined as before.

DDoS attack

In this experiment a distributed denial of service (DDoS) has been introduced in the system in order to evaluate the response of the TRS in presence of a multiple and massive DoS attack. Due to the distributed nature of the TRS, it might be really interesting to analyze its response against a distributed attack.

The results shown in figure 8.6 have been obtained by simulating a scenario of 200 nodes where ten nodes are simultaneously attacked so that they receive 20 times more traffic than in their normal operation. In this moment the attacked nodes start to overuse their resources, flooding the system with not useful data. The system is working normally until the 100th iteration, when the DoS attack is launched.

Even though the attack is distributed, the TRS reacts fast enough and it isolates the attack in the same way than in the previous scenario.

This shows that the information originated by the concept of trust created in the neighborhood of the attacked nodes is enough to isolate some attacks when it is handled by the trust algorithm. Anyway, we need to analyze the response of the system when the attack exceeds the isolation capacities of the close neighbors and it is needed a global response based on all the information handled by the TRS.

In order to do that we have simulated a new DDoS where the traffic injected is higher than before. We use the same characteristics of traffic and computational and communication resources than in the previous severe attack but now the number of attacked nodes is 10 instead of 1.

The combined effect of the 10 simultaneous attacks is enough to reduce the *trust* of most of the nodes within the system. In this situation the maximum diminish is worse than in

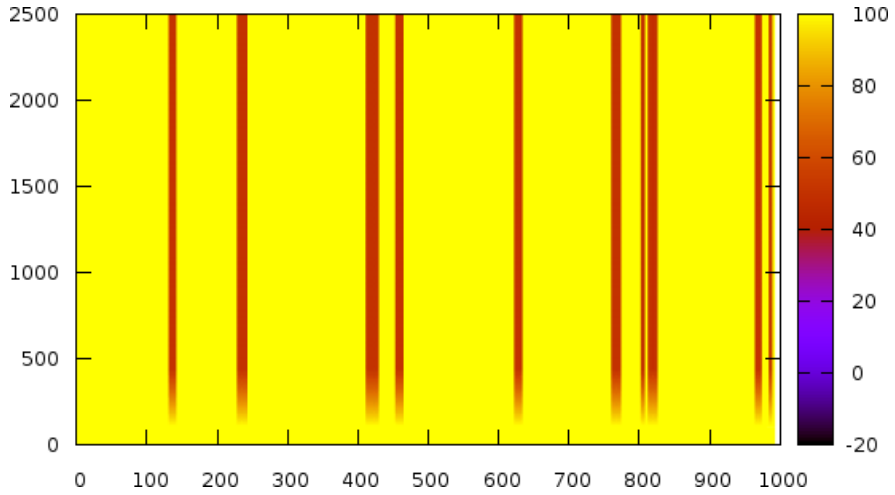


Figure 8.6: Trust evolution for a DDoS attack.

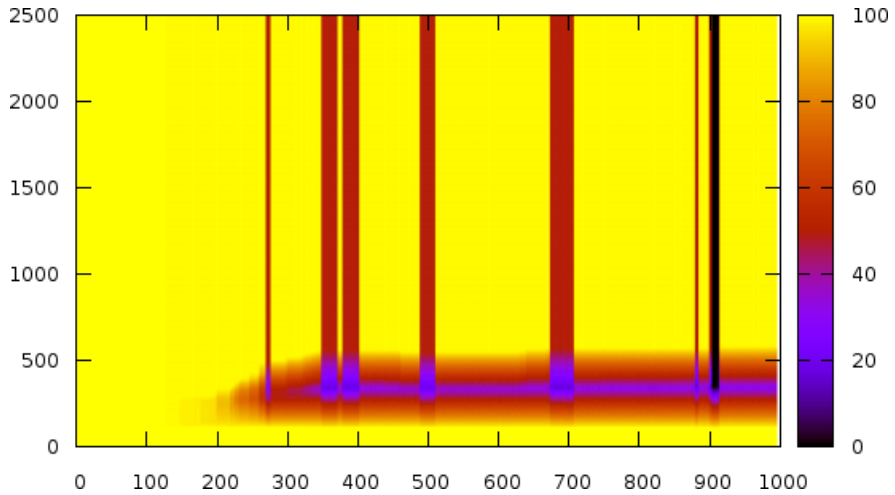


Figure 8.7: Trust evolution for severe a DDoS attack.

the previous scenarios, showing that the combined effect of the attacks can be really harmful. Anyway, the TRS is able to reduce selectively the *trust* of the sources of the DDoS attack and restore the *trust* of their neighbor nodes until they reach a normal behavior.

These results show that the TRS, combined with the SOM algorithm, is a good approach to detect and react against severe distributed attacks.

8.5 Conclusions

Due to the inherent insecurity of WMN nodes, we assume that confidentiality and integrity cannot be preserved for any single node. Based on this premise most of the conventional countermeasures are inadequate to cope with attacks against to WMN.

We have presented a framework for improving security of WMNs based on TRS that is orthogonal to the network protocols. To take advantage of the TRS features, we followed the TRS analysis and design methodologies previously proposed in Chapter 4. The described TRS detects and isolates ill-behaved nodes by rating their *trust* as low. Good ratios of detection times and isolation capacity can be achieved even in a scenario with low redundancy and severe attacks, such as DDoS.

An additional advantage of this approach is that it is quite independent on the attacks. Therefore, it could detect and confine new previously unknown attacks.

9. Advanced Topics: Insider Attacks in Wireless Sensor Networks

In this chapter we will present a TRS to improve the security of WSN against insider attacks. Although all these results mean a significant contribution to the topic of security of WSN, the main goal we pursue with this chapter is **to describe a more in-deep analysis of the implications of developing a TRS** in a real-life scenario.

Thus, this chapter will serve as an example of a complete process of analyzing, designing and optimizing a TRS in order to achieve the goals of the underlying system. Key aspects as mathematical modeling, experimental optimization of some features, or quantitative and qualitative analysis of the results and implications of design and implementation decisions will be presented.

On the other hand, regarding the specific topic of WSN security we have to take into account that the most serious obstacle in further proliferation of wireless sensor networks is their low level of security, where the insider attacks are one of the most challenging issues.

In this work we propose a holistic solution for detecting and confining insider attacks that couples reputation systems with clustering techniques, namely unsupervised genetic algorithm and self-organizing maps, trained for detecting outliers in data. The novelty of this work is the redundancy in detecting agents, their evaluation based on the majority voting and the calculation of the reputation as the average value, which makes it more robust to different attack scenarios and their parameter variations. The algorithms use the feature space based on sequences of sensor outputs (both temporal and spatial), as well as the routing paths used to forward the data to the base station, and designed with the idea of introducing the ability to detect a wide range of attacks. The solution performs both attack detection and recovery from attacks, and it offers many benefits: scalable solution, fast response to adversarial activities, ability to detect unknown attacks, high adaptability and high ability in detecting and confining attacks.

9.1 Introduction to Insider Attacks in Wireless Sensor Networks

Technological advances achieved in the previous two decades have paved the way for the development and deployment of Wireless Sensor Networks (WSN). Their development was mainly motivated by military applications, such as control and surveillance in battlefields, but over the years their deployment has been introduced to other areas, *i.e.*,industrial control and monitoring, etc. In all the applications, it is mandatory to maintain the integrity and the correct operation of the deployed network.

The operation of WSNs relies on a huge number of nodes, so they have to be very cheap, for which they exhibit very limited power and computational resources, small memory size and low bandwidth usage and usually no tamper-resistant hardware is incorporated with any of them. The nodes within a WSN are densely deployed in the area or the phenomenon to be observed, providing in this way high level of redundancy, which can serve as a way to discriminate the erroneous nodes.

The most common approach to deal with the security issue is to add an authentication

system and encryption to communications [219], [220]. However, as was the case with the mesh clients described in Chapter 8, limited resources of the nodes are not able to support the execution of powerful encryption algorithms. The nodes are also vulnerable to side-channel attacks [221] that can be used in order to discover the secret keys. Furthermore, encryption and authentication cannot help in the case of compromised mobile nodes, which often carry private keys that can come into possession of an attacker.

Once the attacker has obtained the secret keys, he can present himself as a legitimate participant in the network and he is able to launch insider attacks. In essence, insider attacks are all the attacks launched by an adversary that is considered to be legitimate participant in the network and that can exploit all the information encountered on the compromised node(s). In this way, the adversary can make much more damage to the network than he was able to do before entering the system, including the possibility of altering the network functioning. Thus, in order to secure WSNs, it is of highest importance to develop security mechanisms which are able to detect and confine insider attacks [222].

9.1.1 Overview of the Proposed Scenario

In this chapter we propose to couple the WSN with a TRS. The TRS assigns a lower reputation to the nodes where it detects adversarial activities and vice versa. Every node is being examined by at least one *observer* that resides on a node in its vicinity and listens to its communication in a promiscuous manner, and executes an algorithms for detecting attacks or temporal and spatial inconsistencies. The trust calculation algorithms include clustering algorithms, namely self-organizing map (SOM) and unsupervised genetic algorithm (GA), but can also include standard algorithms for calculating reputation such as beta-reputation [223].

We further advocate avoiding any contact with the nodes that have low reputation (or which reputation is below certain threshold). In this way, the compromised node remains isolated from the network and has no role in its further operation.

Presence of attackers should not compromise the integrity of the network, *i.e.*,the network should be able to continue working properly. For this to be true, the core network protocols should be protected: aggregation, time synchronization and routing.

In order to be able to significantly affect on these protocols (and in that way to compromise the network), the attacker has to be recognized as a part of the network, *i.e.*,he has to be an insider. In the following text we will see in more detail the possibilities the attacker has for compromising each of these protocols.

Bearing in mind the above-mentioned sensor redundancy, we believe that spatial and temporal characterization of the data, as well as the characterization of the paths used in routing, can be of great importance in discovering manipulated data and/or compromised nodes. Any major data inconsistency can be connected to malicious data manipulation. Furthermore, if any kind of delay to data transmission is introduced, it can be detected by spatial inconsistency. On the other hand, routing paths significantly different from the rest can be the evidence of attacks on routing protocols.

Temporal model is defined for each sensor, while spatial model considers groups of close sensors.

More details on the implementation of the proposed approach are given in the rest of the chapter, which is organized as follows. Section 9.2 gives an overview of the exiting solutions for detecting insider attacks. Sections 9.3 and 9.4 details the proposed solution, while its evaluation is given in Section 9.5. Finally, Section 9.7 draws the most important conclusions.

9.2 Detecting and Confining Insider Attacks in WSN

Sensor networks exhibit some salient features, *e.g.*,redundancy, that helps them to preserve the integrity in the presence of an attacker. However, in most of the cases this is not enough. Techniques for coping with insider attacks can be divided into prevention, detection and recovery

techniques. Since our work concerns mostly detection and the first step of prevention, we will concentrate on these techniques.

The idea of prevention techniques is to stop the attacks from entering the network. However, attack prevention strategy just increases the necessary effort of the attacker [224], but without any support it is not able to entirely protect the network.

Moving on to detection techniques, they can be divided into the techniques for detecting manipulated data and the techniques for detecting compromised nodes. However, the first one are not sufficient by themselves. Therefore, additional techniques have to be deployed for detecting compromised nodes in order to confine the attack. Thus, we concentrate on detecting compromised nodes. On the other hand, attacks can perform two types of compromise: read-only and read-and-write compromise. Read-only are harder to detect, but do not make any damage except violate data confidentiality. These attacks are usually deployed for collecting data for inspection that can discover possibilities for launching more harmful attacks.

The detection of read-and-write compromise can be performed simply by checking if the nodes have been tampered with. However, in most of the cases this is not viable since the areas where the nodes are deployed often cannot be reached, or the network is too big. On the other hand, a number of custom intrusion detection systems (IDS) for sensor networks have been proposed. Some of the representative solutions are given in [225]–[227]. However, they are mainly focused on misbehaving detection, hence are capable of detecting only limited number of attacks, *i.e.,*known attacks and their variations. In order to detect new attacks, they need to be adjusted by human.

Recently few solutions that deploy machine learning techniques appeared [228]–[230]. These solutions uphold the idea that machine learning techniques offer higher level of flexibility and adaptability to the changes of the environment. Furthermore, we often have to deal with incomplete information and noise, and the security requirements themselves are often fuzzy and incomplete. Machine learning techniques are known to cope well with these sorts of problems, which is the main reason they are becoming part of the security solutions, even the commercial ones [231]. However, these techniques consume significant resources. To the best of our knowledge, nobody has proposed any solution for this issue. Moreover, the feature sets the above-mentioned techniques deploy mostly include those features whose values are known to change under the influence of an attacker, or are known to be weak spots. This is their major deficiency, as relying on these features only the known attacks or their variations can be detected. In addition, it assumes that an attacker can exploit only the known vulnerabilities, but general experience is that vulnerability is detected after being exploited by an adversary.

After reviewing these approaches, we can see that there are many proposed solutions for coping with insider attacks in WSN, none of them is general enough to be able to handle greater variety of attacks. For this reason, most of the solutions should work aside with few more that address different aspects of security breaches. However, this can introduce high overhead and consume significant resources.

Another issue is that most of them are able to detect known attacks, but the experience from the network security tells us that the attackers always manage to find possibilities to launch their attacks. For these reasons, our goal with this work is to provide a holistic solution capable of coping with different groups of attacks, both known and unknown. Furthermore, we propose various possibilities for integrating the approach in WSN, bearing in mind the limited resources of the nodes, which is something that does not exist in the state of the art.

9.3 Improving WSN security with TRS

As we describe before, in order to improve the security of WSN by using TRS we only have to apply the methodologies described in Section 3.3. Therefore, we can propose the most suitable architecture, the sources of trust information, and the trust algorithms to achieve the goal of improving its resilience of the system against insider attacks.

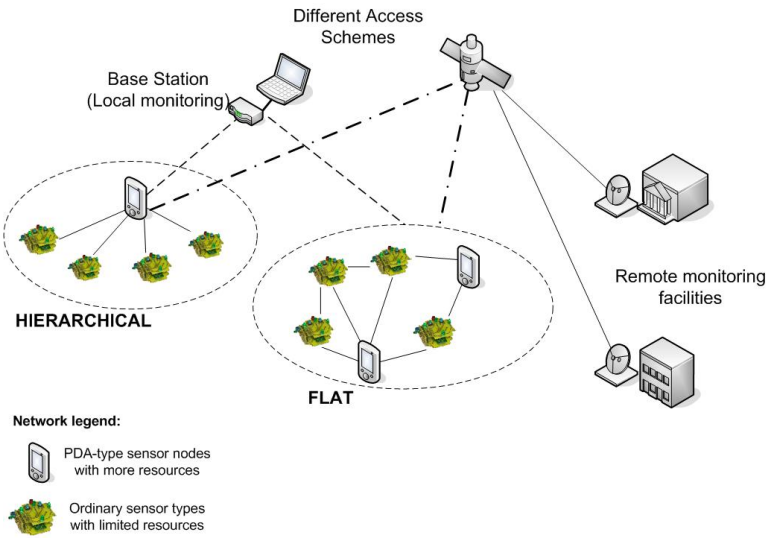


Figure 9.1: Envisioned WSN model

9.3.1 Underlying System Analysis

The first step of the proposed methodology is based on describing the underlying system (*i.e.*,WSN), identifying their main goals and requirements, and analyzing its topology, timing and limitations. Based on this approach, we can identify these topics about the underlying system.

- *Description of the underlying system*. We envision WSNs (Figure 9.1) where most of the sensor nodes exhibit limited resources, but there are also a number of smartphone-like sensors with more computational resources, memory and battery capacity. There is at least one base station as well. The base station is the only entity in the system with enough resources to execute complex encryption algorithms, and software to minimize the likelihood of attacks. For the purpose of this work, we will consider the base station as secure. The number of smartphone-like sensors is significantly smaller than the number of the normal sensors, usually few orders of magnitude smaller. The nodes can organize themselves either in a hierarchical or flat manner. Nodes can be fixed or mobile, although it is assumed that the majority of the nodes are fixed. No constrains regarding routing protocol are assumed.

- *Requirements and goals*. In order to provide uninterrupted network operation, core network protocols (aggregation, routing and time synchronization) have to be secured.

  Regarding the attacks on the aggregation protocol [222], we assume that they demonstrate themselves in skewed aggregated values, which can be the result of either a number of skewed sensed values, or a compromised aggregated node. The assumption is very reasonable, having in mind that the main objective of these attacks is to provide wrong picture of the observed phenomenon, or wrong context information in context aware systems. On the other hand, in time critical systems it is mandatory to receive information within a certain time window. If the attacker manages to introduce delays or desynchronize clock signal in various nodes, the received critical information will not be up to date, which can destabilize the system. Also, if the received information is not up to date, the aggregated value will be skewed, as it will also be out of date. For these reasons, and given the existing redundancy in WSNs, we believe that **these attacks can be detected as temporal and/or spatial inconsistencies of sensed values**.

  Regarding attacks on routing protocols [222], we assume that they will introduce new and different paths than those that have been seen before. Here we have attacks whose

main objective is to compromise the routing protocol, and they usually do it by spoofing or altering the data stored in the routing tables of the nodes. Thus, the resulting routing paths will be different from those used in a normal situation. In the case of wormhole for example, two nodes that are not within each others' radio range result in consecutive routing hops in routing paths, which is not possible in a normal situation. From these examples we can see that the assumption about the attacks resulting in routing paths different from those that appear in normal situation is reasonable. Thus, in this case **these attacks can be detected as temporal inconsistencies in paths used by each node**.

The attack can be either mote-based or laptop-based, but being insider, it possesses valid secret keys (so it is able to authenticate itself as a legitimate participant). Another assumption is that the attack always starts after the initialization of the network, *i.e.*,the network functions normally for some time, which is very reasonable to assume.

- *Topology, timing and limitations*. Regarding this work, the main implications of the topology, timing and limitations of WNS have already introduced in previous sections.

9.3.2 Trust and Reputation System Analysis

If we review the elements and processes of the TRS architecture proposed in Section 3.2, we can identify the following ones:

- *Observers* and *Disseminators*. To deal with the energy consumption and computation resources limitations of the underlying system, only node-sensor with high resources, smartphone-like sensors, and laptops/PC (if there are any in the system) will become *observers* and *disseminators*.

- *Trust Information Sources*. The main sources of trust information will be values sensed by the nodes and routing behavior. Based on this information and analyzing temporal and spatial inconsistencies we can calculate trust values that reflect anomalies in such magnitudes.

- *Dissemination Protocol*. We do not implement any new protocol to exchange trust and reputation information. All nodes in the system can use their default protocols to exchange this kind of data.

- *Reputation Server*. For the purpose of this work, the base station will assume this role, due to the fact it is the only secure entity in the WSN.

- *Trust and Reputation Algorithms*. Because of the topology, complexity, and communication capabilities of the underlying system we will evaluate a combination of a local-area trust algorithm implemented by *observers* deployed throughout the system and a reputation algorithm implemented by the base station. It will allow us to cope with both local anomalies and network-wide anomalies. Because of the special importance of this matter it will be discussed in detail in the next section.

9.4 Trust and Reputation algorithms

In this section we will describe an in-deep analysis of the trust and reputation algorithms proposed for this WSN scenario. Due to the importance of this analysis we will present the detailed structure of this section.

First of all the process of feature extraction are described in Section 9.4.1). Therefore, we will get processed information that will allow us to cope easily with temporal and spatial inconsistencies. Then, a distance function that will enable the quantification of the concept of anomaly is presented in Section 9.4.2. After that, a mathematical analysis of the scope of possible attacks detected and a recovery policy are described in Section 9.4.3 and Section 9.4.4. Finally, some deployment specific issues are presented in Section 9.4.5 and Section 9.4.6.

9.4.1 Feature extraction

As we detailed before, our goal is to find temporal and/or spatial inconsistency in sensed data and in routing data in order to detect manipulated data and/or compromised nodes. For this reason, we follow the idea presented in [80], [176], [232] based on extracted n-grams and their frequencies within different time windows. Thus, the vectors used for characterization that allow the deployment of machine learning are composed of the extracted n-grams.

We already presented a feature extraction for temporal characterization in Section 6.3.4. Therefore, in this section we will only focus on the spatial characterization and the routing paths characterization.

Regarding spatial characterization, the first step is to establish vicinities of nodes that historically have been giving consistent information. Furthermore, since an agent is supposed to reside on a node, vicinities are established using the nodes whose information can reach the agent. In this way, an n-gram for spatial characterization in a moment of time is made of the sensor outputs from that very moment. For example, if sensors S1, S2, S3 that belong to the same group each give the following output: 1 1 1 0 during four time epochs, we characterize them with the following set of n-grams (each n-gram contains at the first position the value of S1, the value of S2 at the second and the value of S3 at the third at a certain time epoch): 111 - occurs 3 times, 000 - occurs once, thus the feature value of each n-gram is: 111 - 0.75, 000 - 0.25, *i.e.*,the frequencies within the observed period of time.

In this work we develop the same principle for characterizing routes that a node has been using to send its sensed data to the base-station. Each routing hop adds its ID to the message that is further forwarded, so the base-station has the information about the routing path together with the message. However, this is not performed with each message in order to avoid the overhead in the communication channel. Yet, having in mind that one routing path is usually used more than once, it is reasonable to assume that the base-station will have all the paths used for routing the data from a certain sensor. As previously mentioned, each sensor has its own model and each feature, *i.e.*,n-gram in the model consists of a predefined number of successive hops used in routing information coming from the node. For example, if during the characterization time, the node has used the following paths for routing its data to the base-station: A-B-C-S - 3 times, A-D-E-F-S - 2 times, A-B-E-F-S - 1 time (A - the node that is sending the data, B, C, ... - other nodes in the network, S- base-Station), we can characterize the routing with the following n-grams (n=3): ABC, BCS, ADE, DEF, EFS, ABE and BEF. In all of the routes, the n-gram ABC occurs 3 times, BCS - 3, ADE - 2, DEF - 2, EFS - 3, ABE - 1, BEF - 1. The total number of n-grams is 15, so dividing the values given above with 15, we get the frequencies of each n-gram which are the values that we assign to our features, *i.e.*,n-grams.

9.4.2 Deployed Distance Function

Since some of the n-grams can appear more than once, it is obvious that the extracted vectors will not be of constant size. Thus, we cannot use standard distance functions. The distance between the instances of the presented model is taken from [233]. It is designed to calculate the distance between two sequences. We have elected this one (among all given in [233]) since it is proven to be the most efficient in the terms of the absolute execution time.

9.4.3 Scope of Attacks Covered With the Approach

As previously mentioned, due to the fact that the anomalies demonstrate themselves as spatial and temporal inconsistencies, no matter what their source is, we will treat attacks as data outliers and deploy clustering techniques, namely SOM and unsupervised GA. Further details on the algorithm implementation can be found in [80], [232].

In the following we will explain the principles of the approach. It is important to mention here that the algorithms can be trained with both clean and unclean data (contains traces of attacks). Furthermore, the algorithms are constantly retrained in order to decrease time lags between model training and model application. The retraining frequency depends on the dynamics of the underlying sensor network.

As we have already described, there are two approaches for detecting outliers using clustering techniques [234] depending on the following two possibilities: detecting outlying clusters or detecting outlying data that belong to non-outlying clusters. For the first case, we calculate the average distance of each cluster to the rest of the clusters (or its closest neighborhood) (MD). In the latter case, we calculate quantization error (QE) of each input as the distance from its corresponding cluster center.

The attacks that can be detected with the proposed approach are those that introduce changes into either the sensed value that is forwarded to the base station or the routing paths. These changes will result in different distribution of the extracted n-grams. However, if we take frequencies as feature values, the sum of the feature values remain the same, *i.e.*,1, so we can write the following equation:

$$\sum_{i=0}^{N} \Delta f_i = 0 \tag{9.1}$$

where N is the total number of the extracted n-grams and Δfi is the change of the feature value of the n-gram i. On the other hand, according to the distance function [233], the introduced change in distance between the attacked instance and any other is:

$$\Delta D = \sum_{i=1}^{N} |\Delta f_i| \tag{9.2}$$

In essence, this is the change introduced in the above defined QE or/and MD values. Thus, the following inequality defines the changes introduced by the attacks:

$$\sum_{i=1}^{N} |\Delta f_i| > f_{th} \tag{9.3}$$

where f_{th} is the threshold value used to distinguish attacks from normal situations.

Now we will see how the changes introduced by the attacker affect on the feature values. Having in mind that each sensed value or a routing hop participates in n features, where n is the size of the n-gram, if the attacker changes one value, the values of $2n$ (at most) features will be changed (the values of newly created n-grams (n at most) with the change will increase, while the values of those that existed before the change (again n at most) will decrease). For example, the third element in the sequence ..1 0 0 1 1.. for $n = 3$ participates in 3 n-grams: 100, 001 and 011. However, if the attacker changes this value into 1, the sequence becomes ..1 0 1 1 1.., in which case the third element participates in these n-grams: 101, 011 and 111. This results in decreased occurrences of the n-grams 100 and 001, while the occurrences of the 101 and 011 become increased (011 appears in both cases, so its total occurrence remains the same). In total, the occurrence of 4 n-grams is changed.

For these reasons, if the attacker introduces N_{err} change in the sample of the size N_{sample}, the value of ΔD will range between 0 (in the case the changes are symmetric, so the effect of one change cancels the effect of another and the distribution does not change at the end), and the value that corresponds to the case when the effects of each change are completely uncorrelated, so they sum together, which is given with the following formula:

$$D_{max} = 2nf_{err} = \frac{2nN_{err}}{N_{sample}} \tag{9.4}$$

Thus, having in mind the correlation of the n-grams, in order to model this change that ranges from 0 to D_{max} we use the next formula:

$$F(\rho) = \beta + (1 - \beta) e^{k\alpha} \tag{9.5}$$

where $\alpha = 1 - \frac{1}{\rho}$, $\beta (< 1$, since the function should grow with ρ) and k are constants defined in the design process (the specific meaning of both will be explained later in this section) and

ρ is the coefficient of total correlation between the n-grams. The value of $F(\rho)$ is β for $\rho = 0$ (the reason for this will be explained in the following), and 1 for $\rho = 1$.

The coefficient of total correlation [235] expresses the amount of dependency that exists among a set of variables. For a given set of k random variables $X_1, X_2, \ldots, X_k$, the total correlation $C(X_1, X_2, \ldots, X_k)$ is given by the following formula:

$$C(X_1, X_2, \ldots, X_k) = \sum_{i=1}^{k} H(X_i) - H(X_1, X_2, \ldots, X_k) \tag{9.6}$$

where $H(X_i)$ is the information entropy of variable X_i, while $H(X_1, X_2, \ldots, X_k)$ is the joint entropy of the variable set $X_1, X_2, \ldots, X_k$. In our case, the variables are the extracted n-grams. For the sake of calculating the above formula, their distribution can be approximated either with a common distribution depending on the purpose of the deployed sensor network, or using the historical data sensed by the network.

Regarding the value of β , we have to take into account that the higher the value of β is, the function becomes closer to its asymptotic function $F(\rho)$ = 1. Thus, the effect of ρ becomes smaller. Similar stands for the value of k. As $k \to 0$, the function becomes closer to the same asymptotic function. In the opposite case, as $k \to \infty$, the function reaches its asymptote: $F(\rho)$ = 0 for $\rho < 1$, $F(\rho)$ = 1 for $\rho = 1$. In both cases the effect of ρ becomes less significant.

Finally, we get the following formula:

$$F(\rho)\,\frac{2nN_{err}}{N_{sample}} > f_{th} \tag{9.7}$$

which gives us the minimal number of changes the attacker has to introduce in order to be detected by the approach:

$$N_{errmin} = \frac{N_{sample}}{2nF(\rho)} f_{th} \tag{9.8}$$

In the previous equation we have the following degrees of freedom: N_{sample}, n and f_{th}. Lower characterization periods (N_{sample}) and the threshold on one side and higher n on the other give us the opportunity to detect the attacker even if he introduces very few changes. However, this can also result in higher false positive rate. Therefore, a trade-off between higher detection and lower false positive rate has to be established. This trade-off decision depends on many factors, such as the application of the deployed WSN or the existing redundancy. Also, the values of both β and k indirectly affect on this value through $F(\rho)$. As the value of β increases or the value of k decreases, the value of $F(\rho)$ for the same ρ increases, which decreases the value of N_{errmin}. In opposite cases, as the value of β decreases or the value of k increases, the value N_{errmin} will increase.

The previous formula also helps us to define the minimal value of β. It derives form the constraint that the maximal possible value of N_{errmin} is equal to N_{sample}. For the same reason, $F(\rho)$ has to be different than 0 for $\rho = 0$ (in the opposite case, $N_{errmin} \to \infty$). This results in following:

$$\beta > \frac{N_{sample}}{2nN_{errmin}} f_{th} \tag{9.9}$$

9.4.4 Trust Calculation and Recovery from Attacks

Every sensor node is being examined by *observers* that execute one of the algorithms for detecting attacks, which reside on nodes in its vicinity and listen to its communication. The agents are trained separately. The system of agents is coupled with a reputation system where each node has its reputation value that basically reflects the level of confidence that others have in it based on its previous behavior.

In our proposal, the output of an agent affects on the reputation system in the way that it assigns lower reputation to the nodes where it detects abnormal activities and vice versa. We

further advocate avoiding any kind of interaction with the low-reputation nodes: to discard any data or request coming from these nodes or to avoid taking them as a routing hop. In this way, compromised nodes remain isolated from the network and have no role in its further performance. After this, additional actions can be performed by the base station, *e.g.*,it can revoke the keys from the compromised nodes, reprogram them, etc.

In this work the reputation is calculated in the following way. f_{th} is taken to be 1 for the following reasons. Having in mind that the attacks will often result in creating new n-grams, it is reasonable to assume that the extracted vector in the presence of attackers will not be a subset of any vector extracted in normal situation, thus the distance will never be lower than 1. We further define two reputation values, $repQE$ and $repMD$ based on the previously defined QE and MD values and afterwards a temporal reputation value r used for updating overall reputation R based on these two values:

$$repQE_{ti} = \begin{cases} 1, & if\ QE_{ti} < 1 \\ 1 - QE_{ti}/2, & if\ QE_{ti} \geqslant 1 \end{cases} \tag{9.10}$$

$$repMD_{ti} = \begin{cases} 1, & if\ MD_{ti} < 1 \\ 1 - MD_{ti}/2, & if\ MD_{ti} \geqslant 1 \end{cases} \tag{9.11}$$

The temporal value (r) for updating overall reputation is calculated in the following way:

$$r_{ti} = \begin{cases} repMD_{ti}, & if\ QE_{ti} < 1 \\ repQE_{ti}, & if\ QE_{ti} \geqslant 1 \end{cases} \tag{9.12}$$

There are two functions for updating the overall reputation of the node, depending whether the current reputation is below or above the established threshold (H) that distinguishes normal and anomalous behavior. If the current reputation is above the threshold and the node starts behaving suspiciously, its reputation will fall quickly. On the other hand, if the reputation is lower than the established threshold, and the node starts behaving properly, it will need to behave properly for some time until it reaches the threshold in order to redeem itself. The first objective is provided by the function $x + log(1.2x)$. Finally, the reputation is updated in the following way:

$$R_{ti} = \begin{cases} R_{t_{-1}i} + 0.05 * (r_{ti} + log(1.2 * r_{ti})), & if R_{t_{-1}i} < H \\ R_{t_{-1}i} + 1.00 * (r_{ti} + log(1.2 * r_{ti})), & if R_{t_{-1}i} \geqslant H \end{cases} \tag{9.13}$$

The second objective is provided by the coefficient c_limit, which takes values lower than 1 and its purpose is to limit selective behavior of a node by decreasing the reputation growth if the reputation value is below the threshold. Very low values of this coefficient obligate nodes to behave properly most of time. If the final reputation value falls out from the $[0, 1]$ range, it is rounded to 0 if it is lower than 0 or to 1 in the opposite case.

However, if during the testing of temporal coherence, we get normal data different from those that the clustering algorithms saw during the training, it is possible to get a high QE value as well. On the other hand, the spatial coherence should not detect any anomalies. Thus, the final reputation will fall only if both spatial and temporal algorithms detect anomalies. In the opposite case, its reputation will not change significantly.

On the other hand, as mentioned in the previous text, in the situations such as the data coming from a node exhibits large variations, temporal inconsistencies are not likely to be detected. However, spatial inconsistencies are very likely to be detected. Thus, spatial inconsistency is sufficient in order to raise an alarm.

Concerning the detection of routing protocol anomalies, the explained approach can tell us if there is something suspicious in routing paths of a certain node. Yet, in order to find out the nodes that are the origin of the attack, we need to add one more step. In this second step, if the reputation of the routes calculated in the previous step is lower then the established threshold, the hops that participated in the bad routes will be added to the global list of bad nodes, or if they already exist, the number of their appearance in bad routes is increased. The similar

principle is performed for the correct nodes. For each node, let the number of its appearances in bad routes be $nBad$ and the number of its appearances in good routes be $nGood$. Finally, if $nGood$ is greater than $nBad$, the node keeps its reputation value, and in the opposite case, it is assigned the following reputation value:

$$\frac{nGood}{nGood + nBad} \tag{9.14}$$

In this way, as the bad node spreads its malicious behavior, its reputation will gradually decrease.

9.4.5 Distributed Organization of Observers

The distributed system is organized as a group of detectors, *i.e.*,intelligent agents that execute one of the detection algorithms (GA or SOM) and assign reputation to sensors. Based on our TRS architecture, this agents are assuming the role of *observers*.

The possibilities of their positioning will be explained in the following section. Considering that there is a possibility that the attacker that has taken over a node can disable or compromise the *observer* that resides on that node, we introduce *observer* redundancy: at least three different *observer* will examine the behavior of each node and all will affect on its reputation. The final trust and the final decision on a node can be implemented in various ways, such as majority voting, average trust, average weighted trust, etc.

Additionally, we assign reputation value to each detector. We have opted for beta reputation [223], since it has strong background in the theory of statistics. It is calculated according to the following formula:

$$R = E(Beta(\alpha + 1, \beta + 1)) = \frac{\alpha + 1}{\alpha + \beta + 2} \tag{9.15}$$

where α stands for the number of correct decisions made by the detector, while β stands for the number of the incorrect ones. We will call this the validation value. The voting system decides whether a response is right or wrong based on the majority voting. The algorithm for calculating reputation of the detectors together with the voting systems is executed in the base station. Each *observer* has to pass through a period of validation: if after a certain period of time its reputation is above the established threshold value, the agent can participate in the detection process.

A potential attacker on this detection process has to be very skillful and powerful. The detection system can be compromised if the majority of the *observers* that perform the same task get compromised at the same time. In the opposite case, if the *observers* get compromised one by one and express their faulty behavior the moment they become compromised, their decision will simply be discarded.

Furthermore, the information about the tasks the agents perform exists only in the base station. Thus, in order to compromise the proposed detection system, the attacker first has to discover with tasks the *observers* are performing and has to be aware that the compromised *observers* can express their flawed behavior only after having compromised the majority of them. Hence, detector redundancy provides high level of robustness against attacks launched on the detection system itself.

On the other hand, learning algorithms have many parameters that should be set from the start, *e.g.*,number of clusters, duration of training, etc. In our case, it is not easy to guess the optimal parameters in the beginning, but with more specimens it is probable to achieve optimal training in a smaller amount of time. The cost of detector redundancy, however, is a higher communication overhead.

9.4.6 Deployment issues

The learning process consists in two parts: training and testing, *i.e.*,detection in our case, which does not necessarily have to be executed in the same device. The training is the part that

consumes much more resources, so it has to be executed in devices that have enough resources, while the detection can be executed even in devices with limited resources.

Thus, we can say that it is possible to distinguish two types of nodes: training nodes and detection nodes or *observers*. Bearing this in mind, there are various possibilities of incorporating our TRS-based detection system in the proposed WSN model (Section 9.3.1):

Training of agents can be performed either in the base station or the Smartphone-like sensors and already trained agents are further distributed to all the nodes. Hence, SmartPhones and the base station can be training nodes, but they can also serve as *observers*. On the other hand, since the detection process does not consume many resources, trained *observers* can be executed even in the sensors with limited resources. Thus, even sensor nodes can serve as *observers*.

Both training and detection of intrusions could be performed in Smartphone-like sensors that are supposed to have enough resources to carry out these operations. In this way the rest of the sensors would not be affected by the incorporation of our system.

Although the distributed organization has many advantages, it has one limitation. Namely, if we assume that the base-station is always the destination of all sensed data, Smartphone sensors will only have partial information about the routing paths and will be able to detect the attack only if it has occurred before the data has reached the sensor that performs the detection. Thus, the detectors should be organized in a way that they cover most of the routing paths from sources to the base-station.

9.4.7 TRS mapping

After analyzing the components and processes involved in the design of a TRS to improve the security of WSN against insider and unknown attacks, we present a complete specification of all the decisions taken in Table 9.1.

Component/Process	**Feature**	
Underlying System		
	Goals/Requirements	Detection time, isolation capacity, throughput maximization
	Functionality provided	Detection and isolation of attacks against core WSN protocols.
	Timing	Event oriented
	Topology	WSN
	Limitations	Nodes: computational, communication and storage resources, power consumption
Entities		Sensors Smartphone-like nodes TRS-observers
	Observed Service	Sensors: Sensed values Smartphone-like nodes: Routing behavior TRS-Observers: rating behavior
	Area of influence	Local
Observer	Deployed in...	Sensors, smartphone-like nodes
	Observed entities	Cluster of the associated smartphone like node
	Observation time	NR
	Range of observation	Cluster of the associated smartphone like node
	Internal vs. external	Internal
Trust Gathering Information	Perception	Sensors: sensed values (time and spatial coherence). Smartphone-like nodes: sensed values, routing behavior

		TRS-observers: ratings
	Communication	Yes
	Memory	Yes
	Categorization	No
	Reputation	Sensors and smartphone-like nodes: No TRS-observers: Yes
	Nature of information	Sensors: Quantitative-feature extraction Smartphone-like nodes: Quantitative - feature extraction TRS-Observer: Quantitative
	Reliability	1
	Redundancy	Yes
	Scope	Situational
Trust Calculation	Base algorithm	SOM,GA
	Calculation Time	NR
	Computational Resources	Sensor: limited computational resources
	Nature of information	Quantitative
	Required information	Perceived and communicated information
	Information consumption	No
	Scope	Sensors: sensed values Smartphone-like sensors: routing behavior TRS-observers: ratings routing behavior
	Dynamism	Yes
	No-transitivity	No transitivity
	Asymmetry	Yes
	Histeresis Loop	Logaritmic update function
Disseminator	Deployed in...	Smartphone-like nodes
	Disseminated observers	m
	Dissemination Range	WSN deployment dependent
	Dissemination Time	NR
	Confidentiality	Not guaranteed
	Filtering	Yes
	Reliability	<1
Dissemination protocol	Base algorithm	WSN communication protocol
	Connection/connectionless	NR
	Point to point/broadcast	NR
	Confidentiality	Not guaranteed
	Integrity	Not guaranteed
Reputation Server	Deployed in...	Base Station
	Nr.of reputation servers	1
	Topology	Central Server
	Internal vs. External	Internal
Reputation Gathering Information	Trust	Yes
	Reputation	Yes
	Other sources	Observers' ratings
	Public vs. Private Information	Private
Reputation Calculation	Base Algorithm	Sensor nodes: logaritmic function Smartphone-like nodes: presence in suspicious routes TRS-Observers: Beta
	Calculation Time	NR
	Computational Resources	NR

Observed entities	Global
Nature of information	Qualitative
Required information	Sensor nodes and smartphone-like nodes: Trust, Reputation TRS-Observers: Observer ratings
Information consumption	No

(NR) Not relevant.

Table 9.1: TRS and security in WSN: system specification.

9.5 Experimental results

9.5.1 Simulation Environment

The proposed approach has been tested on TRS-sim, a simulator of sensor networks developed by our research group and designed using the C++ programming language described in Appendix A.

For the purpose of this experiments the network is organized as clusters of close sensors where each group has its cluster head, as often done in real networks in order to reduce computational overhead and energy consumption. Cluster heads are the only sensors that can participate in the communication between different clusters and also in routing.

We have implemented various attacks in order to test performances of the TRS. In this chapter we will present the results based on the most popular WSN insider attacks [222]:

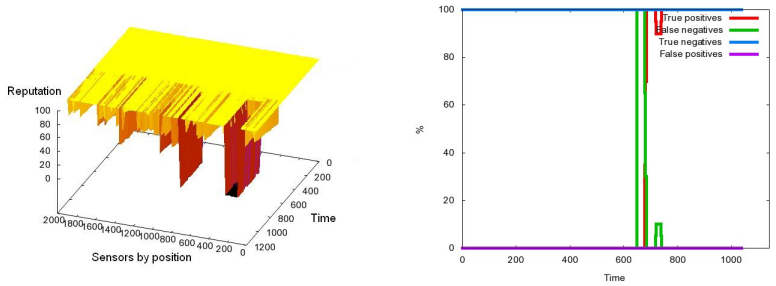
- Sybil. In this scenario, the compromised node pretends to have multiple IDs, either false, *i.e.,*fabricated, or impersonated from other legitimate nodes, *i.e.,*stolen IDs.
- Pulse-delay. This scenario assumes that the data from attacked node(s) have much higher latencies than in the normal case.
- Wormhole. In this scenario the compromised node starts sending data to a node that is not in its vicinity, but to another area that surpasses the range of its radio signal.

In the case of the Sybil, added nodes send random values that may or may not coincide with the values sent by the original good nodes.

The proposed algorithm has been tested on the presented simulated sensor network that contains 200 sensor nodes that can be placed in 2000 different positions. The network simulates a sensor network for detecting presence in the area of application. The groups for spatial characterization are formed in the following way: close sensors that should give the same output are placed in the same group.

The duration of the experiment is 1000 time ticks. One time tick in simulator is the period of time required to perform the necessary operations in the network, and it is equivalent to a sampling period, or time epoch in sensor networks. In the following we will present results in different scenarios regarding the presence of attacks in training data. The final reputation is calculated as the average reputation of the algorithm set that contains both SOM and GA with the following configurations (n-gram size is 3 in all of them):

- Training Ends at tick 500, test every 20 ticks based on previous 40 values.
- Training Ends at tick 400, test every 15 ticks based on previous 40 values.
- Training Ends at tick 300, test every 15 ticks based on previous 40 values.
- Training Ends at tick 200, test every 10 ticks based on previous 40 values.
- Training Ends at tick 150, test every 10 ticks based on previous 40 values.
- Training Ends at tick 50, test every 10 ticks based on previous 20 values.



(a) Reputation Evolution (b) Detection Evolution (TP, FN, TN, FP)

Figure 9.2: The Sybil attack - start at 650

After the tick 500, in the following 100 ticks we perform the process of detector evaluation described in Section 9.4.5, so each algorithm has its reputation. In the process of assigning reputation to the nodes, we only include those detectors which reputation is greater than the established threshold. We will present the results based on different threshold values. In all the following experiments the validation value of the algorithm given by Equation 9.15 is 0.9.

9.5.2 Insider Attack Analysis

In this section we will present the results of the proposed solution based on a set of representative insider attacks explained above. These results represent average cases. In all of the simulation the compromised nodes will be stationed in the same area for the purpose of clearer presentation of the results.

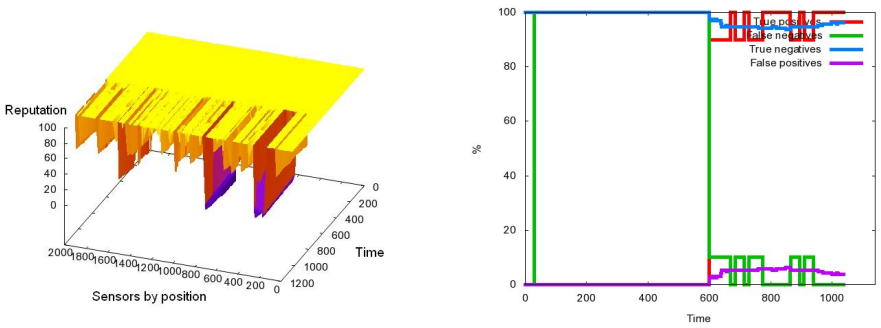
The Sybil Attack

In the following we will see the performances of the proposed solution under the Sybil attack. Again, we will have two different scenarios: the traces of the Sybil attack in the training data, and the case when the Sybil attack starts after the end of the training.

In the first scenario a new node added at the position 800 by the adversary impersonates 10 existing nodes with the IDs from 23 to 32 that take positions 195, 201, 219, 228, 258, 273, 275, 304, 307 and 323. In Figure 9.2 we can observe the reputation and the detection evolution. The threshold for distinguishing normal and compromised nodes for depicting Figure 9.2(b) is taken to be 20, but all the compromised nodes have their reputation lowered to 0, as it can be observed from the Figure 9.2(a). Therefore, any threshold higher than 0 will completely confine the attack. Thus, we can conclude that in this case all the compromised nodes have been detected, *i.e.*,detection rate is 100% with 0% of false positives, and the attack is completely confined. The attack has been detected 30 ticks after the start, *i.e.*,1-2 testing cycles, and completely confined 35 ticks after the start.

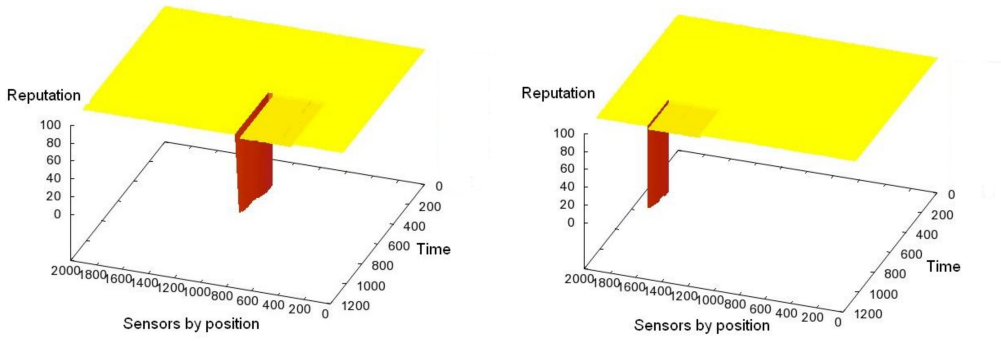
We will show now an example when the Sybil attack starts at the tick 30 (Figure 9.3). In this case a node is inserted at the position 800 and it takes IDs 27-36 from the nodes that are situated at the following positions: 258, 273, 275, 304, 307, 323, 334, 339, 345 and 349.

After experimenting with different threshold values, we have concluded that for the value equal to 60 (Figure 9.3(b)) all the malicious nodes are confined. However, the false positive rate also rises, up to 5.5% at most. In general, we assume that the majority of the data used for training is normal, and in this case we have an algorithm trained with the data obtained in the first 50 ticks (for this algorithm the majority of the data is normal), which is the most important (and responsible) one to detect this attack. Thus, in the case when it is possible that the attacks start in the early stages of network operation and it is necessary to detect them rapidly, more algorithms that finish their training soon after the network initiation can be added to the algorithm set.



(a) Reputation Evolution (b) Detection Evolution (TP, FN, TN, FP) – Th.60

Figure 9.3: The Sybil attack - start at 30



(a) Start at 400 (b) Start at 650

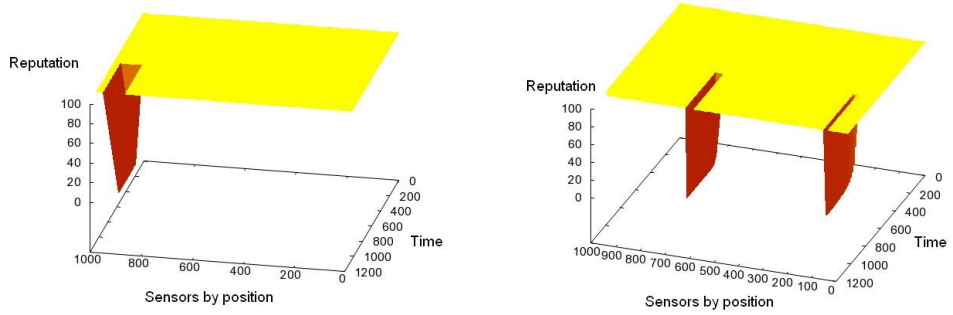
Figure 9.4: The Pulse-delay attack

The Pulse-Delay Attack

The following experiments have been performed on the pulse-delay attack (Figure 9.4). One more time we present results of the scenario when the attack is present in the training data and when it starts after the end of training. In the Figure 9.4(a) we can observe the reputation evolution when the attack starts at the tick 400 and the affected node is situated at the position 793, while in the Figure 9.4(b) we have a situation when the attack starts at the tick 650 and the affected node is situated at the position 1598. In both cases, the attack introduces random delay between 20 and 50 time ticks. Again, we can observe than in both cases the reputation of the attacked node is significantly lowered (15 in the first and 10 in the second), while the rest of the nodes in the same group have their reputation slightly lowered (to 97), but not enough to be reported as compromised. Thus, we can say that in this case the compromised node has been detected and the attack has been confined, with no false positives.

The Wormhole Attack

In the following we will present the results of the wormhole attack detection. These experiments are carried out in the scenario where 100 nodes can take 1000 different positions due to lower simulation time. In the first case the origin of the attack is the node at the position 914 (the link node is not considered to be malicious) and the attack starts at the tick 650. As we can observe in the Figure 9.5(a) this is the only node with the lowered reputation (0), so we can say that in this case we have 100% detection rate with 0% false positives. In the second case the attack starts at tick 250 (Figure 9.5(b)). The origin of the attack is the node at the position 655. As we can see, its reputation is lowered to 0, so it is completely detected and confined. The time of detection and confinement is 40 ticks. However, in this case the reputation of the link node is also lowered to 5 (position 87), but as we do not consider it to be malicious, it is a false positive. Thus, the false positive rate in this case is 1%.



(a) Start at 650

(b) Start at 200

Figure 9.5: Wormhole attack

9.6 Discussion

Beside the quantitative results obtained from the experiments, a diverse set of open issues has to be discussed and analyzed. Topics such as the study of network survivability, minimization of resource consumption, calculation of optimal threshold values, evaluation of the influence of the point of attack, and approaches to reduce false positives of improve detection times are worth to study.

9.6.1 Network Survivability

We say the correct operation is maintained while the network provides the correct picture of the observed phenomenon during the whole time, although the attack has not been completely isolated. In the most general case, this is accomplished while the majority of the nodes provide correct information. Thus, without any detection mechanism, the attacker has to compromise at least $\lceil N/2 \rceil$ sensors, where N is the number of sensors in the given network. However, if such mechanism is present and we consider that the data coming from isolated nodes is being discarded, the attacker has to compromise the majority of the remaining nodes, which increases total number of nodes the attacker has to compromise in order to compromise the whole network. This is demonstrated in the following experiment.

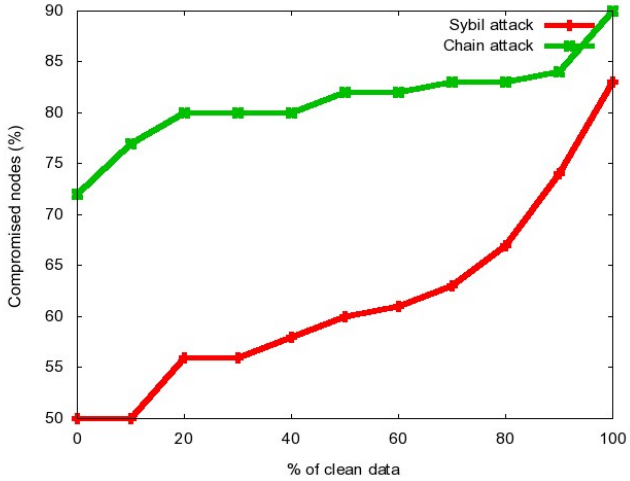


Figure 9.6: Max. % of compromised nodes

The experiment is performed in the same surrounding as the previous ones, in the presence of two attacks, the Sybil attack and the chain attack, which consists of k compromised nodes, where the first (k-1) always forward the data to the next one in the chain, while the last one performs misrouting. The performances of the proposed detector in the presence of this attack are reported in [236].

The attacks compromise random nodes and it is assumed that the nodes from the whole network can be compromised. In Figure 9.6 we observe that in the presence of a detection mechanism the attacker has to introduce more effort in order to compromise the network, where the maximal percentage of compromised nodes that permits network survivability is 83.5%in the case of the Sybil, and 90% in the case of the chain attack, when the attack in both cases starts after the end of training. As the percentage of clean data decreases (*i.e.*,the data without traces of attacks), it is harder to detect all the malicious nodes, for which the attacker needs to introduce less effort in order to compromise the network. However, in the case of the chain attack the effort is always much higher than in the case where there is no detection mechanism. On the other hand, in the case of the Sybil for the situations the training data contains at least 10% of the clean data, the presence of a detection mechanism increases the effort of the attacker necessary to compromise the network.

9.6.2 Resource Consumption

With the aim of proving the viability of performing the training in a smartphone-like device, we have carried out the evaluation of the resource consumption using a Sony Ericsson Xperia X10 Mini with Qualcomm MSM7227 600MHz CPU and Android 1.6. It is important to point out that this is not one of the most powerful smartphones, but rather an average one.

In the case the smartphone monitors 40 nodes, the full battery can provide around one

million training periods of both SOM and GA. For example, if we have 10% of the battery reserved for the algorithm training, this capacity is high enough to perform the training once per day for around 350 years.

Further experiments concerning memory consumption have been performed, as it seems to be the most important issue for implementing the approach in the sensor nodes. The memory consumption of BETA, SOM and GA depending on the number of nodes that are being examined, where this number varies from 2 to 200, is presented in Figure 9.7 The corresponding memory consumptions have the following ranges: (36-124kB), (144-735kB) and (336-3670kB) for BETA, SOM and GA respectively. Thus, given the resource growth trends, we can expect that in near future the implementation of SOM to be viable in ordinary sensor nodes. However, this cannot be claimed for GA. Yet, even the GA that examines 200 nodes can be easily implemented in current smartphone-like devices.

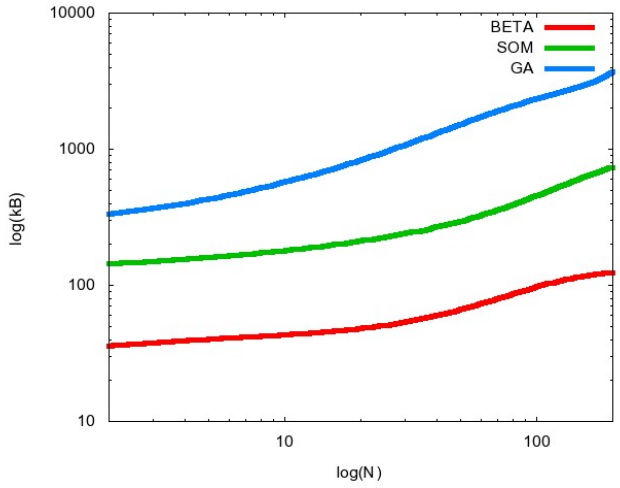


Figure 9.7: Memory consumption vs. number of nodes

9.6.3 Characterization

Regarding the characterization using n-grams, in most of the cases the sensors give consistent output, so we do not expect for the characterization to result in having a huge number of different n-grams. However, in some cases this can happen, when we can apply one of the following possibilities for reducing this number. One possibility is to divide the range of values the sensor can give into few equidistant ranges, and assign a unique value or meaning to all the values that belong to one range. This significantly reduces the number of possible n-grams. Another possibility is to take an average of the values that belong to a certain range.

In most general case, the nodes are distinguished in the network based on their IDs as the information about their position is not always available and can be poisoned as well. Thus, although the information about the position of the nodes in the simulator is known to the base station, we do not use it in the calculations in order not to lose the generality of the approach. This information is only used to present final results. For this reason, in our approach the reputation is assigned to IDs, so the data coming from the same ID is examined in order to look for the inconsistencies.

However, the nodes which IDs have been stolen end up with low reputation, although they are not necessarily malicious. Since the adversary has managed to steal their IDs, these nodes are compromised. We believe that all the nodes with the compromised IDs should be confined since they are potentially malicious. This is exactly what our approach provides: fast detection and prevention of further spreading of the malicious activity. The following steps are left for the decision of the base station, which can further revoke their secret keys, assign new ones, reprogram them, improve their security measures, etc. Furthermore, the

base station can re-assign their reputation values, so they can be re-integrated in the network. Due to the existing redundancy, we can say with high probability that there exists at least one more node in the network that performs the same function as a compromised node. Thus, its temporal confinement does not significantly affect the network, while the opposite could end up in serious damage.

9.6.4 Optimal threshold value

Regarding the optimal value of the threshold that distinguishes the normal from anomalous behavior, it depends on many factors, the most important being the criticality of its operation and the state of the network, *i.e.*,the set of nodes that are active and are available for performing certain operation. If we need to perform a time-critical operation that can carry out the task, we need to assume the risk of including the low reputation nodes in the set of nodes to carry out the task. On the other hand, if security is critical, only the nodes with the highest reputations should be used although it might result in performance losses. The threshold value should also be dynamic, *i.e.*,it should be changed during time in order to adapt to the changing properties of the network.

9.6.5 Starting Point Of Attack

One more point that needs further discussion is the starting point of the attack. We have assumed that the network functions normally for some period of time before the start of the attack. This is a reasonable assumption, as it is highly unlikely for an attack to start from the moment 0 since the networks have been developed and tested in closed and secured environments. However, our approach is capable of detecting the attack that starts not long after the network initiation (Figure 9.3).

9.6.6 Reducing false positives

Another important point in our work is the usage of both spatial and temporal characterization in order to reduce the number of false positives, which was proved to be beneficial when a single detection algorithm was deployed [80]. However, in this work the decision is made based on multiple algorithms, which also helps reducing the number of false positives (for example, the detection of the Sybil attack presented in Figure 9.2 and Figure 9.3 has been performed using only temporal characterization and the number of false positives is 0). This is an important conclusion, since we can avoid the definition of groups, which is not a straightforward process. However, in this way we can only detect the attacks that affect the temporal distribution of the output, but the attacks such as pulse-delay attack can only be detected through spatial inconsistency.

9.6.7 Detection time

Regarding the detection and the confinement time, they both depend on the time tick in the simulator, which is equivalent to the time epoch (also called sensing interval) in the sensor networks. In essence, this is the time between sending two consecutive sensed values and depends on many factors: the purpose of the network, the usual sleep time of the nodes in order to save the energy, etc. Some of the examples are the following ones: 1s for habitat monitoring [237], 40s for railway bridge monitoring [238], etc. Thus, for the cases where the sensing interval is around 1s, the absolute detection time of our approach for the experiments presented in this work ranges from 1s to 130s, which is very fast. As a comparison, reported detection times are around 2000s [239]. On the other hand, for the epoch of 40s, the detection time ranges from 40s to 5200s. Anyhow, the influence of the attack, *i.e.*,the amount of the malicious data it manages to insert into the network, is the same in both cases. However, in the cases when the sensing interval is higher, the attacker has more time, and due to this more possibilities, to launch additional attacks, especially in the cases of laptop attack. Thus,

the deployment of our approach is more beneficial in the applications with lower sensing intervals.

9.7 Conclusions

In this work we have presented a novel and holistic approach to detect insider attacks in WSN.

We have proposed and TRS using clustering algorithms to detect outliers in data that deploy a feature set that is more general than those presented by the solutions of the state-of-the-art.

Furthermore, it does not depend on the presence (or non-presence) of anomalous data during the training, thus it is possible to detect unknown attacks. We have also presented a self-sustained organization of detectors that makes it robust to parameter variations (*e.g.*,starting point) of the attack, and we have proved that a single set of detectors was capable of detecting various types of attacks.

Due to the fact that trust and reputation values are assigned to the nodes according to the decision of the clustering algorithm, the response to attacks consists in assigning low reputation to malicious nodes which will render them isolated from the network and impede them to further propagate their malicious activity.

In our experiments we have demonstrated that our system is capable of detecting and confining various attacks that affect the core network protocols with detection rate of 100%, maintaining low false positive rate.

However, it should be pointed out that false positives in sensor networks are not an issue as big as in network security due to high node redundancy.

We have also demonstrated that our approach is capable of rapid attack detection. Also, it has been proven that the inclusion of the proposed detection mechanisms significantly increases the effort the attacker has to introduce in order to compromise the network.

Although all these results mean a significant contribution to the security of WSN, the main goal we pursue with this chapter is to present a more in-deep analysis of the implications of developing a TRS.

From the point of view of applying the proposed TRS methodologies to improve the performance or security of other systems (WNS in this case), we have presented a more detailed case of study than in previous chapters. Therefore, this work serves as an example of a complete process of analyzing, designing and optimizing a TRS in order to achieve the goals of the underlying system.

Key aspects as mathematical modeling, experimental optimization of some features or quantitative and qualitative analysis of the results and implications of those experiments and presented.

10. Conclusions and Future Work

"I never trust people's assertions, I always judge of them by their actions."

— Ann Radcliffe, *The Mysteries of Udolpho, 1764*

This Ph.D. Thesis has addressed the improvement of the methodologies related to TRS in order to allow a more precise, systematic, complete and secure analysis and design of this kind of systems.

In this Chapter, a synthesis of the conclusions derived from the research undertaken in this Ph.D. thesis is presented, highlighting the contributions of this dissertation to the state-of-the-art. Moreover, we also highlight the open research lines and future research directions derived from this work.

10.1 Summary

As described in the motivation of this Ph.D. thesis, we can see how every discipline related to the concepts of trust and reputation focuses its studies on a specific set of topics, but none of them tries to take advantage of knowledge generated in the others disciplines to improve its behavior or performance.

Detailed topics in some fields are completely obviated in others, and even though the study of some topics within several disciplines produces complementary results, these results are not usually used outside the discipline where they were generated.

Previous research, as shown in Chapter 2, lacks of completeness and only proposes partial solutions to specific fields of application (*e.g.*,trust management models applied to e-commerce environments, WSN, P2P networks, etc.).

This leads us to a very high knowledge dispersion and to a lack in the reuse of methodologies, policies and techniques among different fields.

Reviewing the objectives presented in Chapter 1.3, this Ph.D. thesis has achieved the following results:

- We have compile an extensive literature about the utilization and the characteristics of TRS in different fields such as philosophy, psychology, sociology, economics, business management, communications and networking, online services, etc.
- We have compile of an extensive literature about previously proposed Trust Management Systems, identifying their main components and processes, advantages and disadvantages, etc.
- We have developed a **generic architecture for TRS**, identifying the entities and processes involved in this kind of systems regardless of the field of knowledge or the specific case of use where we apply them. This architecture allows us to define a new unified vocabulary and standard concepts regarding TRS that will be the base of the next contribution.
- We have proposed a **analysis methodology for TRS** that systematically allows anybody to identify all the assets and process involved in this kind of systems. This methodology

allows systematizing the process of understanding the behavior of any TRS independently of the field of knowledge or the specific case of use where we apply them. It allows to draw conclusions about potential limitations of the performance of such TRSs.

- We have proposed a **design methodology for TRS** in order to provide such systems a certain functionality or performance. This methodology describes the steps to select all entities and processes involved in the development and deployment of a TRS in a real-life scenario.

  As a further result of this design methodology a **taxonomy of types of TRS** according to their functional objectives is proposed. In addition, considerations to maximize the performance of any TRS based on their specific goals are presented.

- We have defined **a generic framework for analyzing security** of any kind of system. Based on this framework all assets and processes prone of being attacked can be identified in a systematic way.

  The application of this framework of analysis has allowed us to propose a **complete taxonomy of attacks against TRS**. This taxonomy has allowed to identify and study attacks that had not yet been identified in the literature.

  The combination of the generic security framework and the taxonomy of attacks against TRSs has helped to define a new **methodology to systematically analyze vulnerabilities and possible countermeasures of any TRS in real environments**. Therefore, we can make design decisions that minimize the probability of an attack being successfully completed, as well as we can make design decisions to minimize the impact of an attack in those cases where it cannot be completely avoided.

- Finally, we have developed a TRS simulator that allows to analyze the performance of applying a TRS with a specific set of features to any type of environment.

In addition to the contributions made directly to the field of the TRS, we have made original contributions to different areas of knowledge thanks to the application of the analysis, design and security methodologies previously presented to solve problems from the following fields:

- **Detection of thermal anomalies in Data Centers**. Thanks to the application of the TRS analysis and design methodologies, we successfully implemented a thermal anomaly detection system based on a TRS.

  Its main contribution is the autonomous management of the diverse information available in the data center: by making use of sensor topological information and arranging data in different areas we differentiate between individual sensor *trust* and area-wide *reputation*. Thus, allow us to split CRAC and workload data center anomalies from anomalies due to the malfunction of information gathering sensors.

  In this work we compare the detection performance of two types of algorithms, *i.e.,*SOM and GNG. We show how SOM provides better results for CRAC anomaly detection, yielding detection rates of 100%, in training data with malfunctioning sensors. We also show that GNG yields better detection and isolation rates for workload anomaly detection, reducing the false positive rate when compared to SOM. It is important to note the very low detection and isolation rate, that allows rapid actuation upon a Data Center anomaly.

- **Improving the performance of a harvesting system based on swarm computing and social odometry**. Through the implementation of a TRS, we achieved to improve the ability of coordinating a distributed network of autonomous robots.

  The main contribution lies in the analysis and validation of the incremental improvements that are achieved with proper use information that exist in the system and that are relevant for the TRS, and the implementation of the appropriated trust algorithms based on such information. Simulation results quantitatively showed that the benefits of this

approach were based on the use of *categorization*, *dissemination* and especially *memory*. Therefore, all of them allowed us to achieve better performances than classical odometry approaches. An important issue regarding this topic was the important restrictions imposed by the computational resources available in the autonomous robots.

- **Improving Wireless Mesh Networks security** against attacks against the integrity, confidentiality or availability of data and communications supported by these networks. Thanks to the implementation of a TRS we improved the detection time rate against these kind of attacks and we limited their potential impact over the system.

  The experimental results lead to the fact that the information originated by the concept of trust created in the neighborhood of the attacked nodes is enough to isolate routing and DoS attacks. However, more sever attachs, such as DDos, exceed the isolation capacities of the close neighbors, and a global response based on all the information handled by the TRS is required.

- **We improved the security of Wireless Sensor Networks against advanced attacks**, such as insider attacks, unknown attacks, etc.

  Thanks to the TRS analysis and design methodologies previously described we can implement countermeasures against such attacks in a complex environment. Actually, this contribution can be seen as an advanced version of the previous one (in a more complex and restrictive environment). A deep analysis of detection rates, isolation capacity, system degradation, etc. depending on the intensity of the attacks and the algorithms used in the TRS implemented were provided. Of particular importance is the analysis of the computational impact of these algorithms when they are executed in very low resources nodes.

  In our experiments we have demonstrated that our system is capable of detecting and confining various attacks that affect the core network protocols with detection rate of 100%, maintaining low false positive rate. However, it should be pointed out that false positives in sensor networks are not an issue as big as in network security due to high node redundancy. We have also demonstrated that our approach is capable of rapid attack detection. Also, it has been proven that the inclusion of the proposed detection mechanisms significantly increases the effort the attacker has to introduce in order to compromise the network.

10.2 Future Research Directions

The research developed in this Ph.D. thesis has addressed the improvement of the methodologies related to TRS in order to allow a more precise, systematic, complete and secure analysis and design of this kind of systems.

However, some interesting points of future research have emerged during the evolution of this work. The following paragraphs propose future research directions and improvements of the work presented in this dissertation:

- **Models and Analysis Methodology for TRS**. This work has presented a novel and complete architecture, and a methodology for analyzing TRS. These tools allow to improve the comprehension of any kind of system utilizing a TRS. As we have validated in the last part of this dissertation, they have proved to be enough to face problems in real-life environments.

  However, we have not dealt with other interesting issues such as the historical evolution of trust and reputation, alternative psychological and social mechanisms to enable distributed decision making (*e.g.*,contracts, law, fear, ...) and their implication when applied to non-social environments, etc. Therefore, a more theoretical in-deep study of all the facets and implications of the ideas and processes regarding trust and reputation could be performed. Compiling and processing this kind of information would allow us to generate a complete treatise about Trust and Reputation.

- **Design Methodology for TRS**. This work has presented a methodology for designing TRS in order to provide such systems a certain functionality or performance. However, if we perform a wider study of cases of use we could derive a more deep and detailed knowledge by detecting TRS patterns.

  As it happens in other fields of knowledge, such as software development, based on simple design rules, we can find more evolved design mechanisms. They are known as design patterns. A pattern is a well-known way to solve a specific kind of problem. It we apply this idea to this work, we could find that some TRS architectures and processes are usually better to solve some real-live problems or to achieve a specific performance than others.

  Just as an example, some possible TRS patterns might be: pure-reputation TRS, pure-trust TRS, pure-first-hand information TRS, zero-memory TRS, etc. Knowing when it is better to use one of them instead of the others when designing a TRS, knowing all its advantages and disadvantages, and knowing its critical elements could mean a great improvement in the design phase of any TRS.

- **Security Framework and Taxonomy of TRS attacks**. This work has presented a generic framework for analyzing security of any kind of system, that has allowed us to propose a taxonomy of attacks against TRS where some of those attacks had not yet been identified in the literature.

  A very promising future work will be the study all these new potential attacks: knowing how to effectively implement the attacks, and identifying possible countermeasures and algorithms to prevent them and to isolate the impact of successful attacks.

- **Application of TRS to solve problems of real-life environments**. This work has presented four cases of use where we have successfully applied TRS to improve the performance or the security of real-life environments. Following this line of work, we could extend this design and deployment of TRS to a number of different fields and specific problems. Even though the number of this kind of cases is endless, we point some of special interest because both their wide range of application and the novelty of applying TRS to them.

  A first case of study is the use of TRS to create or improve **Human-Fault-Tolerant systems**. Human beings are prone to make mistakes and many application scenarios are based on the collaboration of these special entities (*i.e.*,humans). In these cases, TRS could stand as a way to improve the overall performance of the system, as well as a mechanism to fast human-fault detection and isolation. Due to the variability and unpredictability of human behavior they are presented as perfect scenarios to test and validate the architectures and methodologies proposed in this work.

  A second case of study is the use of TRS to know the **consumer preferences** regarding new products of services. One of the main uses of TRS is its capability of enabling complex collective decisions, and the consumer's choice of the valuable features of a product or a service are exactly that.

  The third propose case of study is the use of all the proposed architecture and methodologies for analyzing and secure TRS applied to what we might call **self-trust**. Obviously, we use self-trust as the trust someone has in oneself. Although it seems a simple concept it has wide implications.

  In the literature there are countless studies of the concept of self-confidence. However, confidence is only a component of we know as trust. Therefore, this studies are narrow and incomplete. In addition, the use of a systematic methodology to analyze and improve self-trust could enable to improve important human features, such as: self-confidence, self-performance, self-knowledge, etc. And the most important point of this approach is that it would be always based on a scientific and systematic methodology.

Appendix A
TRS-sim: Trust and Reputation System Simulator

A.1 Introduction

In this appendix the architecture and main functionality of *TRS-SIM* will be presented. *TRS-SIM* is a simulator that allows to model all the components and processes of a TRS as described in the architecture presented in Chapter 3. However, low-level details about its specific implementation will not provided because are out of the scope of this Ph.D. Thesis.

TRS-SIM has been successfully applied to all the experimental results presented in this Ph.D. Thesis as well as in most of the research works described in Section 1.5.

A.2 TRS-Sim Architecture

TRS-SIM implements the TRS architecture described in Chapter 3. Therefore, it allows any researcher to carry out any experiment regarding any kind of TRS environment. The process to effectively use the simulator follows the same steps than the design methodology presented in Chapter 4: the researcher has to choose the feature of all the components and processes involved in a TRS dynamic. As we will describe in the next sections, the software provides a set of libraries and basic-components that simplify the simulation process of the most common TRS architectures.

The *TRS-SIM* core components and process can be integrated in any external software as a library. However, a shell client has been implemented to easy its use. Regarding this client, all parameters can be provided through the use of a configuration file or via command line arguments.

- *Underlying System*: *TRS-SIM* allows to define an underlying system based on the specification of: the sources of information (*entities*), the type of data they provide (boolean, numeric, etc.), and their spatial location. All these variables can change over time. Therefore, the researcher can define the main characteristics of the observer underlying system and its evolution over time.

- *Observers*: both the spatial location and the observation range can be defined. Observers have associated a specific *Trust Calculation Algorithm*. Every observer has a Trust-Table that allow for storing and processing any kind of Trust Information, included previously gathered information, categorization information, etc. All or some information from this tables can be communicated thanks to the *disseminators*.

- *Disseminator*: the simulator allows to define the spatial location and the communication range of the disseminators. Disseminator have associated a specific *dissemination protocol*. However, *TRS-SIM* do not simulate low-level communications. It is focused on the TRS dynamics. Therefore, it cannot be use to simulate complex communication environments. If that is needed to obtain some specific experimental results, a proper approach is to integrate the *TRS-SIM* library into a network-specific simulator.

- *Reputation Servers*: the simulator allows to define the *Reputation Calculation Algorithm*. However, only one reputation server can be specified. Therefore, hierarchical or distributed reputation server scenarios cannot be simulated. In addition to its specific gathered information, the *Reputation Server* can store and process any amount of observers' Trust-Tables.
- *Trust and Reputation Calculation Algorithms*: some common algorithms are already provided by *TRS-SIM* and can be used as Trust or Reputation calculation Algorithms, *e.g.*,Self-Organized-Maps, Genetic Algorithm, Immunologic process, Beta function, etc.

A.3 Attacks

One of the main features of *TRS-SIM* is that a wide range of attacks can be easily implemented. The behavior of *observers*, *disseminators*, and *reputation server* can be modified by *Parasites* or *Filters*. *Parasites* are focused on modify the gathering and calculation process, and *filters* change the dissemination process.

Therefore, a *parasite* can change the values observed by and observer from entities of the underlying system, can modify the observers' Trust-Table, change calculated trust values, or even simulate the existence of new observers. *Filters* can modify the content of the transmitted trust and reputation information, change the routing tables of the disseminator, drop information packages, etc.

A.4 Conclusions

In this appendix the architecture and main functionality of *TRS-SIM* has been presented. *TRS-SIM* is a simulator that allows to model all the components and processes of a TRS as described in the architecture and methodologies presented in this Ph.D. Thesis. Therefore, a researcher can completely setup a scenario by defining the entities of the underlying system, observers, disseminator, reputation server, and all the processes involved in a TRS dynamic. In addition, a wide range of attacks can be easily implemented and integrated. It can be used as a shell client or integrated with other software as a library.

However, there are some limitations that future versions of *TRS-SIM* should address: only one reputation server can be defined, and the dissemination protocol do not model low-level communication issues.

TRS-SIM has been successfully applied to all the experimental results presented in this Ph.D. Thesis as well as in most of the research works described in Section 1.5.

Bibliography

[1] P. Brown and H. Lauder, "Human capital, social capital and collective intelligence", in *Social Capital: Critical Perspectives*, S. Baron, J. Field, and T. Schuller, Eds., Oxford: Oxford University Press, 2001, pp. 226–242.

[2] J. Surowiecki, *The Wisdom of Crowds*. Anchor, Aug. 2005, ISBN: 0385721706.

[3] T. W. Malone, R. Laubacher, and C. N. Dellarocas, "Harnessing crowds: mapping the genome of collective intelligence", English, *SSRN eLibrary*, 2009.

[4] G. M. Olson, A. Zimmerman, and N. Bos, *Scientific Collaboration on the Internet*. MIT Press, 2008, ISBN: 0262151200.

[5] D. R. Hofstadter, *Godel, Escher, Bach: An Eternal Golden Braid (Penguin Philosophy)*, New Ed. Penguin Books Ltd, ISBN: 0140179976.

[6] P. Lévy, *Collective intelligence: Mankind's emerging world in cyberspace*, ser. Helix books. Perseus Books, 1999, ISBN: 9780738202617.

[7] H. Bloom, *Global Brain: The Evolution of Mass Mind from the Big Bang to the 21st Century*. Wiley, 2001, ISBN: 9780471419198.

[8] E. Conklin, *Dialogue mapping: Building shared understanding of wicked problems*. J. Wiley, 2006.

[9] J. Howe, *Crowdsourcing: How the Power of the Crowd Is Driving the Future of Business*. Random House Business, 2008, ISBN: 9781905211111.

[10] C. W. Churchman, "Realism in management science: A report", *Management Science*, vol. MT-1, no. 3, pp. 63–81, 1961.

[11] O. F. E. R. Arazy, W. Morgan, and R. Patterson, "Wisdom of the crowds: decentralized knowledge construction in wikipedia", *Social Science Research Network Working Paper Series*, Dec. 2006.

[12] J. E. Introne, "Supporting group decisions by mediating deliberation to improve information pooling", in *Proceedings of the ACM 2009 international conference on Supporting group work*, ser. GROUP '09, Sanibel Island, Florida, USA: ACM, 2009, pp. 189–198, ISBN: 978-1-60558-500-0.

[13] M. Klein, "Enabling large-scale deliberation using attention-mediation metrics", English, *SSRN eLibrary*, 2011.

[14] X. Su and T. M. Khoshgoftaar, "A survey of collaborative filtering techniques", *Adv. in Artif. Intell.*, vol. 2009, 4:2–4:2, 2009, ISSN: 1687-7470.

[15] E. Bonabeau, M. Dorigo, and G. Theraulaz, "Swarm intelligence: From natural to artificial systems", *J. Artificial Societies and Social Simulation*, vol. 4, no. 1, 2001.

[16] R. Leighton and R. P. Feynman, *Surely You Are Joking, Mr.Feynman!* Vintage, Nov. 1992, ISBN: 009917331X.

[17] Amazon, *Amazon site*, http://www.amazon.com/.

[18] IMDB, *IMDB site*, http://www.imdb.com/.

[19] F. B. Viégas, M. Wattenberg, and M. Mckeon, "The hidden order of wikipedia", in, 2007, pp. 445–454.

[20] F. B. Viégas, "The visual side of wikipedia", in *HICSS*, 2007, p. 85.

[21] *40th Hawaii International International Conference on Systems Science (HICSS-40 2007), CD-ROM / Abstracts Proceedings, 3-6 January 2007, Waikoloa, Big Island, HI, USA*, IEEE Computer Society, 2007.

[22] L. von Ahn, B. Maurer, C. Mcmillen, D. Abraham, and M. Blum, "Recaptcha: Human-based character recognition via web security measures", *Science*, vol. 321, no. 5895, pp. 1465–1468, 2008.

[23] J. Levin and B. Nalebuff, "An introduction to vote-counting schemes", *Journal of Economic Perspectives*, vol. 9, no. 1, pp. 3–26, 1995.

[24] Digg, *Digg site*, http://www.digg.com/.

[25] Wikipedia, *Kasparov vs. the world*, http://wki.pe/Kasparov_versus_the_World.

[26] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine", in *Proceedings of the seventh international conference on World Wide Web 7*, ser. WWW7, Brisbane, Australia: Elsevier Science Publishers B. V., 1998, pp. 107–117.

[27] J. Wolfers and E. Zitzewitz, "Prediction markets", Stanford University, Graduate School of Business, Research Papers 1854, Apr. 2004.

[28] E. Servan-Schreiber, J. Wolfers, D. M. Pennock, and B. Galebach, "Prediction markets: Does money matter?", *Electronic Markets*, vol. 14, no. 3, pp. 243–251, 2004.

[29] A. Abdul-Rahman, "A framework for decentralised trust reasoning", PhD thesis, University College London, 2005.

[30] P. Dasgupta, "Trust as a commodity", 2000.

[31] D. Hume, T. Green, and T. Grose, *A treatise of human nature: Being an attempt to introduce the experimental method of reasoning into moral subjects ; and, Dialogues concerning natural religion*, ser. A Treatise of Human Nature: Being an Attempt to Introduce the Experimental Method of Reasoning Into Moral Subjects ; And, Dialogues Concerning Natural Religion v. 1. Longmans, Green, and Co., 1890.

[32] O. Lagerspetz, *Trust: The tacit demand*, ser. Library of ethics and applied philosophy. Kluwer Academic Publishers, 1998, ISBN: 9780792348740.

[33] M. Deutsch, *The Resolution of Conflict: Constructive and Destructive Processes*, ser. Carl Hovland Memorial Lectures Series. Yale University Press, 1977, ISBN: 9780300021868.

[34] R. Hardin, "The street-level epistemology of trust", *Politics & Society*, vol. 21, no. 4, pp. 505–529, Dec. 1993.

[35] J. B. Rotter, "Interpersonal trust, trustworthiness, and gullibility.", *American Psychologist*, vol. 35, no. 1, pp. 1–7, 1980, ISSN: 0003-066X.

[36] M. Karlins and H. Abelson, *Persuasion: How opinions and attitudes are changed*. Springer Pub. Co., 1959.

[37] D. Gambetta, "Can we trust trust?", in *Trust: Making and Breaking Cooperative Relations*, Basil Blackwell, 1988, pp. 213–237.

[38] W. J. Adams and Nathaniel, "Toward a decentralized trust-based access control system for dynamic collaboration", in *6th Annual IEEE Systems, Man and Cybernetics Information Assurance Workshop (IAW 2005)*, West Point, NY, USA: IEEE, Jun. 2005, pp. 317–324, ISBN: 0-7803-9290-6.

[39] N. Luhmann, *Trust and Power*. John Wiley and Sons Ltd., 1979, ISBN: 0471997587.

[40] H. Tajfel and J. C. Turner, *An integrative theory of intergroup conflict*, 1979.

[41] H. S. James Jr., "The trust paradox: a survey of economic inquiries into the nature of trust and trustworthiness", English, *SSRN eLibrary*,

[42] W. Poundstone, *Prisoner's dilemma*, ser. Anchor books. Anchor Books, 1993.

[43] R. Axelrod, *The evolution of cooperation*. Basic Books, 2006, ISBN: 9780465005642.

[44] D. H. Mcknight and N. L. Chervany, "The meanings of trust", University of Minnesota, Carlson School of Management, Tech. Rep., 1996.

[45] F. D. Schoorman, R. C. Mayer, and J. H. Davis, "An integrative model of organizational trust: Past, present, and future", *Academy of Management Review*, vol. 32, no. 2, pp. 344–354, 2007.

[46] C. Hillenbrand and K. Money, "Corporate responsibility and corporate reputation: Two separate concepts or two sides of the same coin?", *Corporate Reputation Review*, vol. 10, no. 4, pp. 261–277, 2007.

[47] R. Marimon, J. P. Nicolini, and P. Teles, "Competition and reputation", European University Institute, Economics Working Papers, 1999.

[48] C. Fombrun, *Reputation: Realizing value from the corporate image*. Harvard Business School Press, 1996, ISBN: 9780875846330.

[49] C. Fombrun and C. Riel, *Fame & Fortune: How Successful Companies Build Winning Reputations*, ser. Financial Times Prentice Hall Books. Pearson Education, 2004.

[50] ISO, *Standard iso2600 website*, http://bit.ly/1P5fueV.

[51] G. Davies, *Corporate reputation and competitiveness*, 2. Reprint. London [u.a.]: Routledge, 2006, XIII, 272, ISBN: 978-0-415-28743-2.

[52] T. Peters, "The brand called you", *Fast Company*, vol. 10, no. 10, pp. 1–14, 1997.

[53] D. B. Bromley, *Reputation, Image and Impression Management*. John Wiley & Sons, Apr. 1993, ISBN: 0471938696.

[54] W. Diffie and M. E. Hellman, "New directions in cryptography", *IEEE Transactions on Information Theory*, vol. IT-22, no. 6, pp. 644–654, 1976.

[55] R. Rivest, A. Shamir, and L. Adleman, "A method for obtaining digital signatures and public-key cryptosystems", *Communications of the ACM*, vol. 21, pp. 120–126, 1978.

[56] G Theodorakopoulos and J. S. Baras, *On trust models and trust evaluation metrics for ad hoc networks*, 2006.

[57] Y. L. Sun, S. Member, Z. Han, and K. J. R. Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks", *IEEE Journal on Selected Area in Communications*, vol. 24, pp. 305–317, 2006.

[58] L. C. Dept and L. Capra. (2004). Towards a human trust model for mobile ad-hoc networks.

[59] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks", *Proceedings of the 2004 ACM workshop on Wireless security WiSe 04*, no. October, p. 1, 2004.

[60] G. Theodorakopoulos, "Distributed trust evaluation in ad-hoc networks", PhD thesis, 2004.

[61] Y. Sun, W. Yu, Z. Han, and K. J. R. Liu, "Trust modeling and evaluation in ad hoc networks", *GLOBECOM 05 IEEE Global Telecommunications Conference 2005*, vol. 3, pp. 1862–1867, 2005.

[62] T. A. Zia, "Reputation-based trust management in wireless sensor networks", in *2008 International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Sydney, Australia: IEEE, Dec. 2008, pp. 163–166, ISBN: 978-1-4244-2956-1.

[63] Z. Banković, J. M. Moya, D. Fraga, A. Araujo, J. Vallejo, and J. M. de Goyeneche, "Distributed intrusion detection system for wireless sensor networks based on a reputation system coupled with kernel self-organizing maps", To be published in: Int. Comp. Aided Design.

[64] Z. Yao, D. Kim, I. Lee, K. Kim, and J. Jang, "A security framework with trust management for sensor networks", in *Security and Privacy for Emerging Areas in Communication Networks, 2005. Workshop of the 1st International Conference on*, 2005, pp. 190 –198.

[65] J. Moya, Á. Araujo, Z. Banković, J. de Goyeneche, J. Vallejo, P. Malagón, D. Villanueva, D. Fraga, E. Romero, and J. Blesa, "Improving security for scada sensor networks with reputation systems and self-organizing maps", *Sensors*, vol. 9, no. 11, p. 9380, 2009.

[66] T. Grandison and M. Sloman, "A survey of trust in internet applications.", *IEEE Communications Surveys and Tutorials*, vol. 3, no. 4, pp. 2–16, 2000.

[67] eBay, *eBay site*, http://www.ebay.com/.

[68] Booking, *Booking site*, http://www.booking.com/.

[69] AirBnB, *AirBnB site*, http://www.airbnb.com/.

[70] S. Buchegger and J.-Y. L. Boudec, "A robust reputation system for p2p and mobile adhoc networks", 2004.

[71] S. D. Kamvar, 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*, ser. WWW '03, Budapest, Hungary: ACM, 2003, pp. 640–651, ISBN: 1-58113-680-3.

[72] S. Marti and H. Garcia-Molina, "Taxonomy of trust: Categorizing p2p reputation systems", *Comput. Netw.*, vol. 50, pp. 472–484, 4 2006, ISSN: 1389-1286.

[73] J. Golbeck and J. Hendler, "Inferring trust relationships in web-based social networks", *ACM Transactions on Internet Technology*, vol. 7, 2005.

[74] V. Buskens, "The social structure of trust", *Social Networks*, vol. 20, pp. 265–289, 1998.

[75] J. Golbeck, "Computing with trust: Definition, properties, and algorithms", in *Securecomm and Workshops, 2006*, 2006, pp. 1 –7.

[76] ——, *Computing with Social Trust (Human-Computer Interaction Series)*. Springer, 2008, ISBN: 1848003552.

[77] P. B. Lowry, D. Zhang, L. Zhou, and X. Fu, "The impact of national culture and social presence on trust and communication quality within collaborative groups", in *HICSS*, 2007, p. 12.

[78] D. Fraga, Á. Gutiérrez, J. C. Vallejo, A. Campo, and Z. Banković, "Improving social odometry robot networks with distributed reputation systems for collaborative purposes", *Sensors*, pp. 11 372–11 389, 2011.

[79] Z. Bankovic, D. Fraga, J. M. Moya, J. C. Vallejo, P. Malagón, Á. Araujo, J.-M. de Goyeneche, E. Romero, J. Blesa, D. Villanueva, and O. Nieto-Taladriz, "Improving security in wmns with reputation systems and self-organizing maps", *Journal of Network and Computer Applications*, vol. 34, no. 2, pp. 455 –463, 2011, Efficient and Robust Security and Services of Wireless Mesh Networks, ISSN: 1084-8045.

[80] Z. Banković, J. M. Moya, D. Fraga, A. Araujo, J. C. Vallejo, and J.-M. de Goyeneche, "Distributed intrusion detection system for wireless sensor networks based on a reputation system coupled with kernel self-organizing maps", *Integr. Comput.-Aided Eng.*, vol. 17, pp. 87–102, 2 2010, ISSN: 1069-2509.

[81] M. Zapater, D. Fraga, P. Malagón, Z. Bankovic, and J. M. Moya, "Self-organizing maps versus growing neural gas in detecting anomalies in data centres", *Logic Journal of the IGPL*, vol. 23, no. 3, pp. 495–505, 2015.

[82] Z. Banković, D. Fraga, J. M. Moya, J. C. Vallejo, P. MalagóN, A. Araujo, J.-M. De Goyeneche, E. Romero, J. Blesa, D. Villanueva, and O. Nieto-Taladriz, "Bio-inspired enhancement of reputation systems for intelligent environments", *Inf. Sci.*, vol. 222, pp. 99–112, Feb. 2013, ISSN: 0020-0255.

[83] Z. Banković, J. C. Vallejo, D. Fraga, and J. M. Moya, "Detecting false testimonies in reputation systems using self-organizing maps", *Logic Journal of the IGPL*, vol. 21, no. 4, pp. 549–559, 2013.

[84] Z. Bankovic, D. F. Aydillo, J. M. M. Fernández, and J. C. V. López, "Detecting unknown attacks in wireless sensor networks that contain mobile nodes", *Sensors*, vol. 12, no. 8, pp. 10 834–10 850, 2012.

[85] J. M. Moya, J. C. Vallejo, D. Fraga, Á. Araujo, D. Villanueva, and J.-M. De Goyeneche, "Using reputation systems and non-deterministic routing to secure wireless sensor networks", *Sensors*, vol. 9, no. 5, p. 3958, 2009, ISSN: 1424-8220.

[86] D. Fraga, Z. Bankovic, and J. M. Moya, "A taxonomy of trust and reputation system attacks", in *11th IEEE International Conference on Trust, Security and Privacy in Computing and Communications, TrustCom 2012, Liverpool, United Kingdom, June 25-27, 2012*, 2012, pp. 41–50.

[87] Z. Bankovic, J. Moya, D. Fraga, J. Vallejo, and P. Malagon, "Holistic solution for confining insider attacks in wireless sensor networks using reputation systems coupled with clustering techniques", in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on*, 2011, pp. 61–72.

[88] Z. Bankovic, D. Fraga, J. C. Vallejo, and J. M. Moya, "Improving reputation systems for wireless sensor networks using genetic algorithms", in *Proceedings of the 13th Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '11, Dublin, Ireland: ACM, 2011, pp. 1643–1650, ISBN: 978-1-4503-0557-0.

[89] Z. Bankovic, D. Fraga, J. C. Vallejo, and J. M. Moya, "Self-organizing maps versus growing neural gas in detecting data outliers for security applications", in *Hybrid Artificial Intelligent Systems - 7th International Conference, HAIS 2012, Salamanca, Spain, March 28-30th, 2012. Proceedings, Part II*, 2012, pp. 89–96.

[90] Z. Bankovic, J. C. Vallejo, D. Fraga, and J. M. Moya, "Detecting bad-mouthing attacks on reputation systems using self-organizing maps", in *Computational Intelligence in Security for Information Systems - 4th International Conference, CISIS 2011, Held at IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011. Proceedings*, Á. Herrero and E. Corchado, Eds., ser. Lecture Notes in Computer Science, vol. 6694, Springer, 2011, pp. 9–16, ISBN: 978-3-642-21322-9.

[91] Z. Banković, J. M. Moya, D. Fraga, and J. C. Vallejo, "Detecting unknown attacks in wireless sensor networks using clustering techniques", in *Proceedings of the 6th International Conference on Hybrid Artificial Intelligent Systems - Volume Part I*, ser. HAIS'11, Wroclaw, Poland: Springer-Verlag, 2011, pp. 214–221, ISBN: 978-3-642-21218-5.

[92] Z. Bankovic, D. Fraga, J. M. Moya, J. C. Vallejo, Á. Araujo, P. Malagón, J. de Goyeneche, D. Villanueva, E. Romero, and J. Blesa, "Detecting and confining sybil attack in wireless sensor networks based on reputation systems coupled with self-organizing maps", in *Artificial Intelligence Applications and Innovations - 6th IFIP WG 12.5 International Conference, AIAI 2010, Larnaca, Cyprus, October 6-7, 2010. Proceedings*, 2010, pp. 311–318.

[93] Z. Bankovic, J. M. Moya, E. Romero, J. Blesa, D. Fraga, J. C. Vallejo, Á. Araujo, P. Malagón, J. de Goyeneche, D. Villanueva, and O. Nieto-Taladriz, "Using clustering techniques for intelligent camera-based user interfaces", *Logic Journal of the IGPL*, vol. 20, no. 3, pp. 589–597, 2012.

[94] P. Arroba, D. Fraga, J. C. Vallejo, Á. Araujo, and J. M. Moya, "A methodology for developing accessible mobile platforms over leading devices for visually impaired people", in *Ambient Assisted Living - Third International Workshop, IWAAL 2011, Held at IWANN 2011, Torremolinos-Málaga, Spain, June 8-10, 2011. Proceedings*, 2011, pp. 209–215.

[95] Z. Banković, E. Romero, J. Blesa, J. M. Moya, D. Fraga, J. C. Vallejo, Á. Araujo, P. Malagón, J. de Goyeneche, D. Villanueva, and O. Nieto-Taladriz, "Using self-organizing maps for intelligent camera-based user interfaces", in *Hybrid Artificial Intelligence Systems, 5th International Conference, HAIS 2010, San Sebastián, Spain, June 23-25, 2010. Proceedings, Part II*, 2010, pp. 486–492.

[96] E. Romero, Á. Araujo, J. M. Moya, J. de Goyeneche, J. C. Vallejo, P. Malagón, D. Villanueva, and D. Fraga, "Image processing based services for ambient assistant scenarios", in *Distributed Computing, Artificial Intelligence, Bioinformatics, Soft Computing, and Ambient Assisted Living, 10th International Work-Conference on Artificial Neural Networks, IWANN 2009 Workshops, Salamanca, Spain, June 10-12, 2009. Proceedings, Part II*, 2009, pp. 800–807.

[97] Á. Araujo, D. Fraga, J. M. Fernandez, and O. Nieto-Taladriz, "Domotic platform based on multipurpose wireless technology with distributed processing capabilities", in *Proceedings of the IEEE 15th International Symposium on Personal, Indoor and Mobile Radio Communications, PIMRC 2004, 5-8 September 2004, Barcelona, Spain*, 2004, pp. 3003–3007.

[98] J.-H. Cho, A. Swami, and I.-R. Chen, "Why trust is not proportional to risk", *2012 Seventh International Conference on Availability, Reliability and Security*, vol. 0, pp. 11–18, 2007.

[99] W. Sherchan, S. Nepal, and C. Paris, "A survey of trust in social networks", *ACM Comput. Surv.*, vol. 45, no. 4, 47:1–47:33, Aug. 2013, ISSN: 0360-0300.

[100] J. Sabater and C. Sierra, "Review on computational trust and reputation models", *Artificial Intelligence Review*, vol. 24, pp. 33–60, 2005.

[101] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision", *Decis. Support Syst.*, vol. 43, pp. 618–644, 2 2007, ISSN: 0167-9236.

[102] S. Weeks, "Understanding trust management systems", pp. 94–105.

[103] M. Kinateder, E. Baschny, and K. Rothermel, "Towards a generic trust model - comparison of various trust update algorithms", in *Trust Management*, ser. Lecture Notes in Computer Science, P. Herrmann, V. Issarny, and S. Shiu, Eds., vol. 3477, Berlin, Heidelberg: Springer Berlin / Heidelberg, 2005, ch. 13, pp. 119–134, ISBN: 978-3-540-26042-4.

[104] H. Li and M. Singhal, "Trust management in distributed systems", *Computer*, vol. 40, no. 2, pp. 45–53, Feb. 2007, ISSN: 0018-9162.

[105] E. Aivaloglou, S. Gritzalis, and C. Skianis, "Trust establishment in ad hoc and sensor networks", English, in *Critical Information Infrastructures Security*, ser. Lecture Notes in Computer Science, J. Lopez, Ed., vol. 4347, Springer Berlin Heidelberg, 2006, pp. 179–194, ISBN: 978-3-540-69083-2.

[106] W. J. Adams, G. C. Hadjichristofi, and N. J. D. IV, "Calculating a node's reputation in a mobile ad hoc network.", in *IPCCC*, IEEE, 2005, pp. 303–307, ISBN: 0-7803-8991-3.

[107] S. P. Marsh, "Formalising trust as a computational concept", PhD thesis, University of Stirling, 1994.

[108] Fortune, *Fortune's most admired companies list*, http://cnnmon.ie/1fbpaln.

[109] C. Castelfranchi and R. Falcone, "Principles of trust for mas: Cognitive anatomy, social importance, and quantification", in *Proceedings of the International Conference on Multi-Agent Systems*, Paris, France, 1998, pp. 72–79.

[110] G. Zacharia, M. I. of Technology. Dept. of Architecture. Program in Media Arts, and Sciences, *Collaborative reputation mechanisms for online communities*. Massachusetts Institute of Technology, School of Architecture, Planning, Program in Media Arts, and Sciences, 1999.

[111] A. Abdul-Rahman and S. Hailes, "Supporting trust in virtual communities", in *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 6 - Volume 6*, ser. HICSS '00, Washington, DC, USA: IEEE Computer Society, 2000, pp. 6007–, ISBN: 0-7695-0493-0.

[112] M. Schillo, P. Funk, I. Stadtwald, and M. Rovatsos, *Using trust for detecting deceitful agents in artificial societies*, 2000.

[113] B. Yu and M. P. Singh, "Towards a probabilistic model of distributed reputation management", in *Proceedings of the Fourth Workshop on Deception Fraud and Trust in Agent Societies*. 2001, pp. 125–137.

[114] J. Sabater and C. Sierra, "Regret: A reputation model for gregarious societies", 2001, pp. 61–69.

[115] K. Aberer and Z. Despotovic, "Managing trust in a peer-2-peer information system", in *Proceedings of the Tenth International Conference on Information and Knowledge Management*, ser. CIKM '01, Atlanta, Georgia, USA: ACM, 2001, pp. 310–317, ISBN: 1-58113-436-3.

[116] B. Esfandiari and S. Chandrasekharan, *On how agents make friends: Mechanisms for trust acquisition*, 2001.

[117] J. Carbo and J. Molina-Lopez, "An extension of a fuzzy reputation agent trust model (afras) in the art testbed", *Soft Computing - A Fusion of Foundations, Methodologies and Applications*, vol. 14, pp. 821–831, 8 2010, 10.1007/s00500-009-0470-9, ISSN: 1432-7643.

[118] *Evolving and managing trust in grid computing systems*, vol. 3, 2002, 1424–1429 vol.3.

[119] J. Carter and E. Bitting, "Reputation formalization for an information-sharing multi-agent sytem", in *Computational Intelligence*, 2002, pp. 515–534.

[120] V. Cahill, B. Sh, E. Gray, N. Dimmock, A. Twigg, J. Bacon, C. English, W. Wagealla, S. Terzis, P. Nixon, C. Bryce, G. D. M. Serugendo, J. marc Seigneur, M. Carbone, K. Krukow, C. Jensen, Y. Chen, and M. Nielsen, "Using trust for secure collaboration in uncertain environments", *IEEE Pervasive Computing*, vol. 2, pp. 52–61, 2003.

[121] Y. Wang and J. Vassileva, "Bayesian network-based trust model", 2003, pp. 372–378.

[122] B. Dragovic, E. Kotsovinos, S. Hand, S. H, and P. R. Pietzuch, "Xenotrust: Event-based distributed trust management", in *In Proceedings of International Workshop on Database and Expert Systems Applications*, 2003, pp. 410–414.

[123] B. Shand, N. Dimmock, and J. Bacon, "Trust for ubiquitous, transparent collaboration.", in *PerCom*, IEEE Computer Society, 2003, pp. 153–160, ISBN: 0-7695-1893-1.

[124] T. D. Huynh, N. R. Jennings, and N. R. Shadbolt, "Fire: An integrated trust and reputation model for open multi-agent systems", in *In Proceedings of the 16th European Conference on Artificial Intelligence (ECAI*, 2004, pp. 18–22.

[125] W. Nejdl, D. Olmedilla, and M. Winslett, "Peertrust: Automated trust negotiation for peers on the semantic web", in *In Workshop on Secure Data Management in a Connected World (SDM.04*, 2004, pp. 118–132.

[126] K. MacMillan, K. Money, S. Downing, and C. Hillenbrand, "Giving your organisation spirit: An overview and call to action for directors on issues of corporate governance, corporate reputation and corporate responsibility", 2004, pp. 15–42.

[127] M. Krasniewski, P. Varadharajan, B. Rabeler, S. Bagchi, and Y. Hu, "Tibfit: Trust index based fault tolerance for arbitrary data faults in sensor networks", in *Dependable Systems and Networks, 2005. DSN 2005. Proceedings. International Conference on*, 2005, pp. 672–681.

[128] W. T. L. Teacy, J. Patel, N. R. Jennings, and M. Luck, "Travos: Trust and reputation in the context of inaccurate information sources", *Journal of Autonomous Agents and Multi-Agent Systems*, vol. 12, p. 2006, 2006.

[129] G. V. Crosby, N. Pissinou, and J. Gadze, "A framework for trust-based cluster head election in wireless sensor networks", in *Proceedings of the Second IEEE Workshop on Dependability and Security in Sensor Networks and Systems*, ser. DSSNS '06, Washington, DC, USA: IEEE Computer Society, 2006, pp. 13–22, ISBN: 0-7695-2529-6.

[130] E. Kotsovinos and A. Williams, "Bambootrust: Practical scalable trust management for global public computing.", in *SAC*, H. Haddad, Ed., ACM, 2006, pp. 1893–1897, ISBN: 1-59593-108-2.

[131] *FilmTrust: Movie recommendations using trust in web-based social networks*, vol. 1, 2006, pp. 282–286.

[132] A. Jøsang and W. Quattrociocchi, "Advanced features in bayesian reputation systems", in *Proceedings of the 6th International Conference on Trust, Privacy and Security in Digital Business*, ser. TrustBus '09, Linz, Austria: Springer-Verlag, 2009, pp. 105–114, ISBN: 978-3-642-03747-4.

[133] T. S. Ferguson, "A bayesian analysis of some nonparametric problems", *The Annals of Statistics*, vol. 1, no. 2, pp. 209–230, 1973, ISSN: 00905364.

[134] A. Jøsang and J. Haller, "Dirichlet reputation systems", in *INTERNATIONAL CONFERENCE ON AVAILABILITY, RELIABILITY AND SECURITY*, IEEE Computer Society, 2007, pp. 112–119.

[135] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks", *ACM Trans. Sen. Netw.*, vol. 4, no. 3, 15:1–15:37, Jun. 2008, ISSN: 1550-4859.

[136] A. Srinivasan, J. Teitelbaum, and J. Wu, "Drbts: Distributed reputation-based beacon trust system.", in *DASC*, IEEE Computer Society, 2006, pp. 277–283, ISBN: 0-7695-2539-3.

[137] A. Jøsang, "Conditional reasoning with subjective logic", *Multiple-Valued Logic and Soft Computing*, vol. 15, no. 1, pp. 5–38, 2009.

[138] Becerra, Manuel, Lunnan, Randi, Huemer, and Lars, "Trustworthiness, risk, and the transfer of tacit and explicit knowledge between alliance partners", *Journal of Management Studies*, vol. 45, no. 4, pp. 691–713, Jun. 2008, ISSN: 0022-2380.

[139] J.-H. Cho, K. Chan, and S. Adali, "A survey on trust modeling", *ACM Comput. Surv.*, vol. 48, no. 2, 28:1–28:40, Oct. 2015, ISSN: 0360-0300.

[140] L. Xiong, L. Liu, and M. Ahamad, "Countering feedback sparsity and manipulation in reputation systems", in *Collaborative Computing: Networking, Applications and Worksharing, 2007. CollaborateCom 2007. International Conference on*, 2007, pp. 203 –212.

[141] J. R. Douceur, "The sybil attack", in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, ser. IPTPS '01, London, UK: Springer-Verlag, 2002, pp. 251–260, ISBN: 3-540-44179-4.

[142] Z. Banković, J. C. Vallejo, D. Fraga, and J. M. Moya, "Detecting bad-mouthing attacks on reputation systems using self-organizing maps", in *Proceedings of the 4th international conference on Computational intelligence in security for information systems*, ser. CISIS'11, Torremolinos-Málaga, Spain: Springer-Verlag, 2011, pp. 9–16, ISBN: 978-3-642-21322-9.

[143] R. Bhattacharjee, "Avoiding ballot stuffing in ebay-like reputation systems. third workshop on economics of peer-to-peer systems", in *In: P2PECON .05: Proceeding of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems*, ACM Press, 2005, pp. 133–137.

[144] R. Kerr and R. Cohen, "Smart cheaters do prosper: Defeating trust and reputation systems", in *Proceedings of The 8th International Conference on Autonomous Agents and Multiagent Systems - Volume 2*, ser. AAMAS '09, Budapest, Hungary: International Foundation for Autonomous Agents and Multiagent Systems, 2009, pp. 993–1000, ISBN: 978-0-9817381-7-8.

[145] M. Feldman, C. Papadimitriou, J. Chuang, and I. Stoica, "Free-riding and whitewashing in peer-to-peer systems", in *Proceedings of the ACM SIGCOMM workshop on Practice and theory of incentives in networked systems*, ser. PINS '04, Portland, Oregon, USA: ACM, 2004, pp. 228–236, ISBN: 1-58113-942-X.

[146] E. Amoroso, *Fundamentals of computer security technology*. PTR Prentice Hall, 1994, ISBN: 9780131089297.

[147] J. D. Howard and T. A. Longstaff, *A common language for computer security incidents*, 1998.

[148] U. Lindqvist and E. Jonsson, "How to systematically classify computer security intrusions", *Security and Privacy, IEEE Symposium on*, vol. 0, p. 0154, 1997, ISSN: 1540-7993.

[149] I. Krsul, "Software vulnerability analysis", PhD Thesis, PhD thesis, Department of Computer Sciences, Purdue University, 1998.

[150] R. Bibsey and D. Hollingworth, "Protection analysis: Final report", *Information Sciences Institute University of Southern California Marina Del Rey CA USA Technical Report ISISR7813*, vol. 81, no. 2-3, p. 31, 1978.

[151] R. Abbott, R. Project, L. L. Laboratory, and U. S. N. B. of Standards, *Security analysis and enhancements of computer operating systems: Final report*, ser. Security Analysis and Enhancements of Computer Operating Systems: Final Report v. 6,n.º 13. National Bureau of Standards, 1976.

[152] M. Bishop and M. Bishop, "A taxonomy of unix system and network vulnerabilities", Tech. Rep., 1995.

[153] M. Bishop and D. Bailey, *A critical analysis of vulnerability taxonomies*, 1996.

[154] H. JD, "An analysis of security incidents on the internet 1989-1995.", PhD thesis, Department of Computer Sciences, Purdue University, 1997.

[155] D. L. Lough, *A taxonomy of computer attacks with applications to wireless networks*, 2001.

[156] S. Hansman and R. Hunt, "A taxonomy of network and computer attacks", *Computers & Security (COMPSEC)*, vol. 24, no. 1, pp. 31–43, 2005.

[157] E. D. Bell and J. L. La Padula, *Secure computer system: Unified exposition and multics interpretation*, Bedford, MA, 1976.

[158] D. D. Clark and D. R. Wilson, "A comparison of commercial and military computer security policies", in *1987 IEEE Symposium on Security and Privacy*, IEEE Computer Society Press, 1987, pp. 184–194.

[159] D. Parker, *Fighting computer crime*. Scribner, 1983, ISBN: 9780684177960.

[160] J. Koomey, "Growth in data center electricity use 2005 to 2010", Analytics Press, Oakland, CA, Tech. Rep., 2011.

[161] N. Rasmussen, "Calculating total cooling requirements for data centers", American Power Conversion, Whitepaper, 2007.

[162] R. American Society of Heating and A.-C. Engineers, *2008 ashrae environmental guidelines for datacom equipment - expanding the recommend environmental envelope*, 2008.

[163] A. T. 9, "Thermal guidelines for data processing environments-expanded data center classes and usage guidance", ASHRAE, Tech. Rep., 2011.

[164] D. Atienza, G. De Micheli, L. Benini, J. Ayala, P. Valle, M. DeBole, and V. Narayanan, "Reliability-aware design for nanometer-scale devices", in *Proceedings of the 2008 Asia and South Pacific Design Automation Conference*, IEEE Computer Society Press, Jan. 2008, pp. 549–554, ISBN: 978-1-4244-1921-0.

[165] L. Lima, P. Novais, R. Costa, J. Bulas-Cruz, and J. Neves, "Group decision making and quality-of-information in e-health systems.", *Logic Journal of the IGPL*, vol. 19, no. 2, pp. 315–332, 2011.

[166] E. Corchado, A. Arroyo, and V. Tricio, "Soft computing models to identify typical meteorological days.", *Logic Journal of the IGPL*, vol. 19, no. 2, pp. 373–383, 2011.

[167] R. F. Sullivan, "Alternating cold and hot aisles provides more reliable cooling for server farms", Uptime Institute, Tech. Rep., 2000.

[168] N. El-Sayed, I. A. Stefanovici, G. Amvrosiadis, A. A. Hwang, and B. Schroeder, "Temperature management in data centers: Why some (might) like it hot", *SIGMETRICS Perform. Eval. Rev.*, vol. 40, no. 1, pp. 163–174, Jun. 2012, ISSN: 0163-5999.

[169] R. Romadhon, M. Ali, A. Mahdzir, and Y. Abakr, *Optimization of cooling systems in data centre by computational fluid dynamics model and simulation*, Undetermined.

[170] B. Haaland, W. Min, P. Z. G. Qian, and Y. Amemiya, "A statistical approach to thermal management of data centers under steady state and system perturbations", *Journal of the American Statistical Association*, vol. 105, no. 491, pp. 1030–1041, 2010.

[171] E. K. Lee, H. Viswanathan, and D. Pompili, "Model-based thermal anomaly detection in cloud datacenters", in *Distributed Computing in Sensor Systems (DCOSS), 2013 IEEE International Conference on*, 2013, pp. 191–198.

[172] J. Sedano, J. R. Villar, L. Curiel, E. Corchado, and E. A. de la Cal, "Learning and training techniques in fuzzy control for energy efficiency in buildings", *Logic Journal of the IGPL*, vol. 20, no. 4, pp. 757–769, 2012.

[173] O. Depren, M. Topallar, E. Anarim, and M. K. Ciliz, "An intelligent intrusion detection system (ids) for anomaly and misuse detection in computer networks", *Expert Syst. Appl.*, vol. 29, no. 4, pp. 713–722, Nov. 2005, ISSN: 0957-4174.

[174] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998, ISBN: 0132733501.

[175] B. Fritzke, "A growing neural gas network learns topologies", in *Advances in Neural Information Processing Systems 7*, MIT Press, 1995, pp. 625–632.

[176] J. M. Moya, A. Araujo, Z. Banković, J.-M. de Goyeneche, J. C. Vallejo, P. Malagón, D. Villanueva, D. Fraga, E. Romero, and J. Blesa, "Improving security for scada sensor networks with reputation systems and self-organizing maps", *Sensors*, vol. 9, no. 11, pp. 9380–9397, 2009, ISSN: 1424-8220.

[177] J. Lopez, R. Roman, I. Agudo, and C. Fernandez-Gago, "Trust management systems for wireless sensor networks: Best practices", *Comput. Commun.*, vol. 33, pp. 1086–1093, 9 2010, ISSN: 0140-3664.

[178] *Spec cpu2006 benchmark descriptions*, 2013.

[179] M. Yoo, A. B. Jette, M. A. Grondona, "Slurm: simple linux utility for resource management", *LECTURE NOTES IN COMPUTER SCIENCE*, 2003.

[180] T. Larsen, M. Bak, N. Andersen, and O. Ravn, "Location estimation for autonomously guided vehicle using an augmented Kalman filter to autocalibrate the odometry", in *FUSION 98 Spie Conference*, Las Vegas, Nevada: CSREA Press, 1998, pp. 33–39.

[181] S. Thrun, W. Burgard, and D. Fox, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3D mapping", in *Proceedings of the IEEE International Conference on Robotics and Automation*, Piscataway, NJ: Robotics and Automation Society, 2000, pp. 321–328.

[182] R. Grabowski, L. Navarro-Serment, C. Paredis, and P. Khosla, "Heterogeneous teams of modular robots for mapping and exploration", *Autonomous Robots*, vol. 8, no. 2, pp. 293–308, 2000.

[183] S. Nouyan, "Teamwork in a swarm of robots - an experiment in search and retrieval", PhD thesis, Université Libre de Bruxelles, 2008.

[184] R. Vaughan, K. Stoy, G. Sukhatme, and M. Matarić, "Lost: Localization-space trails for robot teams", *IEEE Transactions on Robotics and Automation*, vol. 18, no. 5, pp. 796–812, 2002.

[185] S. Nouyan, A. Campo, and M. Dorigo, "Path formation in a robot swarm: Self-organized strategies to find your way home", *Swarm Intelligence*, vol. 2, no. 1, pp. 1–23, 2008.

[186] A. Gutiérrez, A. Campo, F. Monasterio-Huelin, L. Magdalena, and M. Dorigo, "Collective decision-making based on social odometry", *Neural Computing & Applications*, vol. 19, no. 6, pp. 807–823, 2010.

[187] A. Gutiérrez, A. Campo, F. C. Santos, C. Pinciroli, and M. Dorigo, "Social odometry in populations of autonomous robots", in *Proceedings of the 6th international conference on Ant Colony Optimization and Swarm Intelligence*, ser. ANTS '08, Berlin, Heidelberg: Springer-Verlag, 2008, pp. 371–378, ISBN: 978-3-540-87526-0.

[188] L. Feng, J. Borenstein, and H. Everett, *Where am I? Sensors and Methods for Autonomous Mobile Robot Positioning*. Ann Arbor, MI: University of Michigan Press, 1994.

[189] A. Gutiérrez, A. Campo, F. C. Santos, F. Monasterio-Huelin, and M. Dorigo, "Social odometry: Imitation based odometry in collective robotics", *International Journal of Advanced Robotic Systems*, vol. 6, no. 2, pp. 129–136, 2009.

[190] A. Gutiérrez, A. Campo, F. Monasterio-Huelin, and L. Magdalena, "Self-organized distributed localization based on social odometry", in *Introduction to Modern Robotics I*, D. Chugo and S. Yokota, Eds., Annerley, Australia: iConcept Press, 2011, ch. 1, pp. 1–24.

[191] F. C. Santos, J. M. Pacheco, and T. Lenaerts, "Cooperation prevails when individuals adjust their social ties", *PLOS COMPUTATIONAL BIOLOGY*, vol. 2, no. 10, pp. 1284–1291, Oct. 2006.

[192] A. Josang and R. Ismail, "The beta reputation system", in *Proceedings of the 15th Bled Electronic Commerce Conference (Bled EC)*, Slovenia, June 17 - 19: Bled eCommerce Conference, 2002, 41:1 –41:14.

[193] Z. Banković, S. Bojanić, O. Nieto, and A. Badii, "Unsupervised genetic algorithm deployed for intrusion detection", in *Hybrid Artificial Intelligence Systems*, ser. Lecture Notes in Computer Science, Springer, 2008, pp. 132 –139.

[194] A. Munoz and J. Muruzábal, "Self-organizing maps for outlier detection", *Neurocomputing*, vol. 18, no. 1-3, pp. 33 –60, 1998, ISSN: 0925-2312.

[195] École Polytechnique Fédérale de Lausanne, *E-puck website*, http://www.e-puck.org/.

[196] F. Mondada, M. Bonani, X. Raemy, J. Pugh, C. Cianci, A. Klaptocz, S. Magnenat, J. christophe Zufferey, D. Floreano, and A. Martinoli, "The e-puck, a robot designed for education in engineering", in *In Proceedings of the 9th Conference on Autonomous Robot Systems and Competitions*, 2009, pp. 59–65.

[197] A. L. Christensen, "Efficient neuro-evolution of hole-avoidance and phototaxis for a swarm-bot", Université Libre de Bruxelles, Bruxelles, Belgium, DEA thesis, 2005.

[198] J. R. Douceur, "The sybil attack", in *Revised Papers from the First International Workshop on Peer-to-Peer Systems*, Springer-Verlag, 2002, pp. 251 –260, ISBN: 3-540-44179-4.

[199] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis & defenses", in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, Berkeley, CA, USA: ACM, 2004, pp. 259 –268, ISBN: 1-58113-846-6.

[200] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures", in *Sensor Network Protocols and Applications, 2003. Proceedings of the First IEEE. 2003 IEEE International Workshop on*, 2003, pp. 113 –127.

[201] Q. Zhang, P. Wang, D. S. Reeves, and P. Ning, "Defending against sybil attacks in sensor networks", in *Proceedings of the Second International Workshop on Security in Distributed Computing Systems (SDCS) (ICDCSW'05) - Volume 02*, IEEE Computer Society, 2005, pp. 185 –191, ISBN: 0-7695-2328-5-02.

[202] H. Chan, A. Perrig, and D. Song, "Random key predistribution schemes for sensor networks", in *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, 2003, pp. 197 –213, ISBN: 1081-6011.

[203] W. Du, J. Deng, Y. S. Han, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks", in *Proceedings of the 10th ACM conference on Computer and communications security*, Washington, DC, USA: ACM, 2003, pp. 42 –51, ISBN: 1-58113-738-9.

[204] L. Eschenauer and V. D. Gligor, "A key-management scheme for distributed sensor networks", in *Proceedings of the 9th ACM conference on Computer and communications security*, Washington, DC, USA: ACM, 2002, pp. 41 –47, ISBN: 1-58113-612-9.

[205] D. Mukhopadhyay and I. Saha, "Location verification based defense against sybil attack in sensor networks", in *Distributed Computing and Networking*, Springer, 2006, pp. 509 –521.

[206] M. Demirbas and Y. Song, "An RSSI-based scheme for sybil attack detection in wireless sensor networks", in *Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, New York, NY, USA: IEEE Computer Society, 2006, pp. 564 –570, ISBN: 0-7695-2593-8.

[207] A. D. Wood and J. A. Stankovic, "Denial of service in sensor networks", *Computer*, vol. 35, no. 10, pp. 54 –62, 2002.

[208] A. Pathan, H. Lee, and C. S. Hong, "Security in wireless sensor networks: Issues and challenges", in *The 8th International Conference on Advanced Communication Technology, ICACT 2006*, vol. 2, Gangwon-Do, Korea, Feb. 20 - 22, 2006, pp. 1043 –1048.

[209] Y. Hu, A. Perrig, and D. Johnson, "Packet leashes: A defense against wormhole attacks in wireless networks", in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 3, 2003, pp. 1976 –1986, ISBN: 0743-166X.

[210] M. Conti, R. D. Pietro, and L. V. Mancini, "ECCE: enhanced cooperative channel establishment for secure pair-wise communication in wireless sensor networks", *Ad Hoc Netw.*, vol. 5, no. 1, pp. 49 –62, 2007, ISSN: 1570-8705.

[211] N. Subramanian, C. Yang, and W. Zhang, "Securing distributed data storage and retrieval in sensor networks", *Pervasive Mob. Comput.*, vol. 3, no. 6, pp. 659 –676, 2007.

[212] P. Papadimitratos, Z. Haas, P. Papadimitratos, and Z. J. Haas, "Secure routing for mobile ad hoc networks", in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS)*, 2002, pp. 27 –31.

[213] Y. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks", *Wirel. Netw.*, vol. 11, no. 1-2, pp. 21 –38, 2005.

[214] A. Perrig, R. Canetti, D. Song, and J. D Tygar, "Efficient and secure source authentication for multicast", in *Proc. Network and Distributed System Security Symposium, NDSS*, 2001, pp. 35 –46.

[215] M. G. Zapata, "Secure ad hoc on-demand distance vector routing", *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, no. 3, pp. 106 –107, 2002.

[216] P. Papadimitratos and Z. J. Haas, "Secure link state routing for mobile ad hoc networks", in *Proceedings of the 2003 Symposium on Applications and the Internet Workshops (SAINT'03 Workshops)*, IEEE Computer Society, 2003, p. 379, ISBN: 0-7695-1873-7.

[217] J. M. Moya, J. C. Vallejo, D. Fraga, A. Araujo, D. Villanueva, and J. de Goyeneche, "Using reputation systems and Non-Deterministic routing to secure wireless sensor networks", *Sensors*, vol. 9, no. 5, pp. 3958 –3980, 2009, ISSN: 1424-8220.

[218] M. K. Marina and S. R. Das, "Ad hoc on-demand multipath distance vector routing", *SIGMOBILE Mob. Comput. Commun. Rev.*, vol. 6, pp. 92–93, 3 2002, ISSN: 1559-1662.

[219] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: A link layer security architecture for wireless sensor networks", in *SenSys*, J. A. Stankovic, A. Arora, and R. Govindan, Eds., ACM, 2004, pp. 162–175, ISBN: 1-58113-879-2.

[220] D. J. Malan, M. Welsh, and M. D. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography", in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conf. on*, Santa Clara, CA, USA, 2004, pp. 71–80.

[221] H. Bar-El, "Introduction to side-channel attacks", 2003.

[222] T. G. Roosta, "Attacks and defenses of ubiquitous sensor networks", PhD thesis, EECS Department, University of California, Berkeley, 2008.

[223] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks", in *Procs. of the 2nd ACM workshop on Security of ad hoc and sensor networks*, ser. SASN '04, Washington DC, USA: ACM, 2004, pp. 66–77, ISBN: 1-58113-972-1.

[224] C. Hartung, J. Balasalle, and R. Han, "Node compromise in sensor networks: the need for secure systems", University of Colorado at Boulder, Tech. Rep., Jan. 2005.

[225] I. Krontiris, T. Giannetsos, and T. Dimitriou, "Lidea: A distributed lightweight intrusion detection architecture for sensor networks", in *Procs. of the 4th international conference on Security and privacy in communication netowrks*, ser. SecureComm '08, Istanbul, Turkey: ACM, 2008, 20:1–20:10, ISBN: 978-1-60558-241-2.

[226] R. Roman, "Applying intrusion detection systems to wireless sensor networks", in *In CCNC 2006: Proceeding of the 3rd IEEE Consumer Communications and Networking Conf.*, 2006, pp. 640–644.

[227] T. H. Hai, F. Khan, and E.-N. Huh, "Hybrid intrusion detection system for wireless sensor networks", in *Procs. of the 2007 international conference on Computational science and Its applications - Volume Part II*, ser. ICCSA'07, Kuala Lumpur, Malaysia: Springer-Verlag, 2007, pp. 383–396, ISBN: 3-540-74475-4, 978-3-540-74475-7.

[228] S. Kaplantzis, A. Shilton, N. Mani, and Y. Sekercioglu, "Detecting selective forwarding attacks in wireless sensor networks using support vector machines", in *Intelligent Sensors, Sensor Networks and Information, 2007. ISSNIP 2007. 3rd Int. Conf. on*, Dec. 2007, pp. 335 –340.

[229] Z. Yu and J. Tsai, "A framework of machine learning based intrusion detection for wireless sensor networks", in *Sensor Networks, Ubiquitous and Trustworthy Computing, 2008. SUTC '08. IEEE Int. Conf. on*, 2008, pp. 272 –279.

[230] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion detection for routing attacks in sensor networks", *IJDSN*, vol. 2, no. 4, pp. 313–332, 2006.

[231] *Adaptive security analyzer*, http://www.privacyware.com/index_ASAPro.html.

[232] Z. Banković, J. M. Moya, A. Araujo, and J.-M. De Goyeneche, "Intrusion detection in sensor networks using clustering and immune systems", in *Procs. of the 10th international conference on Intelligent data engineering and automated learning*, ser. IDEAL'09, Burgos, Spain: Springer-Verlag, 2009, pp. 408–415, ISBN: 3-642-04393-3, 978-3-642-04393-2.

[233] K. Rieck and P. Laskov, "Linear-time computation of similarity measures for sequential data", *J. Mach. Learn. Res.*, vol. 9, pp. 23–48, 2008, ISSN: 1532-4435.

[234] A. Mu'noz and J. Muruzábal, "Self-organizing maps for outlier detection", *Neurocomputing*, vol. 18, no. 1-3, pp. 33–60, 1998.

[235] M. Studený and J. Vejnarová, "The multiinformation function as a tool for measuring stochastic dependence", in *Procs. of the NATO Advanced Study Institute on Learning in graphical models*, Erice, Italy: Kluwer Academic Publishers, 1998, pp. 261–297.

[236] Z. Banković, J. C. Vallejo, P. Malagón, A. Araujo, and J. M. Moya, "Eliminating routing protocol anomalies in wireless sensor networks using ai techniques", in *Procs. of the 3rd ACM workshop on Artificial intelligence and security*, ser. AISec '10, Chicago, Illinois, USA: ACM, 2010, pp. 8–13, ISBN: 978-1-4503-0088-9.

[237] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson, "Wireless sensor networks for habitat monitoring", in *Procs. of the 1st ACM international workshop on Wireless sensor networks and applications*, ser. WSNA '02, Atlanta, Georgia, USA: ACM, 2002, pp. 88–97, ISBN: 1-58113-589-0.

[238] K. Chebrolu, B. Raman, N. Mishra, P. K. Valiveti, and R. Kumar, "Brimon: A sensor network system for railway bridge monitoring", in *Proceeding of the 6th international conference on Mobile systems, applications, and services*, ser. MobiSys '08, Breckenridge, CO, USA: ACM, 2008, pp. 2–14, ISBN: 978-1-60558-139-2.

[239] M. Conti, R. Di Pietro, L. V. Mancini, and A. Mei, "Emergent properties: Detection of the node-capture attack in mobile wireless sensor networks", in *Procs. of the first ACM conference on Wireless network security*, ser. WiSec '08, Alexandria, VA, USA: ACM, 2008, pp. 214–219, ISBN: 978-1-59593-814-5.

"It is impossible to go through life without trust: That is to be imprisoned in the worst cell of all, oneself."

— Graham Greene, *The Ministry of Fear*