A DATA SECURITY PERSPECTIVE ON INFORMATION TRANSMISSION OVER DISTRIBUTED SYSTEMS

Teză destinată obţinerii
titlului ştiinţific de doctor inginer
la
Universitatea "Politehnica" din Timişoara
în domeniul ŞTIINŢA CALCULATOARELOR
de către

Ing. Răzvan Virgil BOGDAN

Conducător ştiinţific: Prof.Dr.Ing. Mircea VLĂDUŢIU
Referenţi ştiinţifici: Prof.Dr.Ing. Mircea PETRESCU
Prof.Dr.Ing. Daniela POPESCU
Prof.Dr.Ing. Nicolae ROBU

Ziua susţinerii tezei: 11.09.2009

Seriile Teze de doctorat ale UPT sunt:

1. Automatică
2. Chimie
3. Energetică
4. Ingineria Chimică
5. Inginerie Civilă
6. Inginerie Electrică
7. Inginerie Electronică şi Telecomunicaţii
8. Inginerie Industrială
9. Inginerie Mecanică
10. Ştiinţa Calculatoarelor
11. Ştiinţa şi Ingineria Materialelor

Universitatea „Politehnica” din Timişoara a iniţiat seriile de mai sus în scopul diseminării expertizei, cunoştinţelor şi rezultatelor cercetărilor întreprinse în cadrul şcolii doctorale a universităţii. Seriile conţin, potrivit H.B.Ex.S Nr. 14 / 14.07.2006, tezele de doctorat susţinute în universitate începând cu 1 octombrie 2006.

România, 300159 Timişoara, Bd. Republicii 9,
tel. 0256 403823, fax. 0256 403221
e-mail: editura@edipol.upt.ro

Cuvânt înainte

Teza de doctorat a fost elaborată pe parcursul activităţii mele în cadrul grupului de cercetare ACSA (Advanced Computing Systems & Architectures), de la Facultatea de Automatică şi Calculatoare, Departamentul Calculatoare, a Unversităţii "Politehnica" din Timişoara.

Mulţumiri deosebite şi recunoştinţă se cuvin a fi adresate domnului profesor coordonator, prof.dr.ing. Mircea Vlăduţiu pentru sesiunile de lucru desfăşurate periodic de-a lungul anilor şi pentru încurajările din momentele mai puţin favorabile. Discuţiile deosebit de competente din domeniul fiablităţii, securităţii sistemelor de calcul, şi nu numai, mi-au fost de un real sprijin în elaborarea tezei.

Doresc de asemenea să mulţumesc în mod aparte domnului rector prof.dr.ing. Nicolae Robu pentru sprijinul acordat în diseminarea rezultatelor acestei teze de doctorat.

O parte a acestei teze este rezultatul colaborării pe care am avut-o cu ing. Versavia Ancuşa, căreia vreau să îi mulţumesc pentru puterea de muncă excepţională şi iscusinţa de care dă dovadă.

Această teză nu ar fi fost posibilă fără susţinerea necontenită şi nemotivată pe care familia mi-a oferit-o în toţi anii formării. Le mulţumesc pe această cale pentru răbdarea şi înţelegerea de care au dat dovadă.

Nu în ultimul rând, îi mulţumesc lui Dumnezeu pentru inspiraţia şi prezenţa sa în fiecare moment al realizării tezei şi nu numai.

Timişoara, septembrie 2009 Răzvan Bogdan

Familiei mele: mama, tata şi fratele meu

Bogdan, Răzvan Virgil

A Data Security Perspective On Information Transmission Over Distributed Systems

Teze de doctorat ale UPT, Seria 10, Nr. 21, Editura Politehnica, 2009, 110 pagini, 54 figuri, 18 tabele.

ISSN: 1842-7707

ISBN: 978-973-625-909-8

Cuvinte cheie: arhitecturi din al treilea val, grid inteligent, securitate, model de ameninţări, testabilitate, metrici, sistem distribuit

Rezumat,

Prin subiectul abordat, teza de doctorat răspunde unor probleme de maximă actualitate privind neajunsurile existente în arhitecturi specifice celui de-al treilea val computaţional. În acest sens a fost propusă o arhitectură care să adreseze aceste aspecte.

S-au implementat şi propus diferite tehnici pentru atingerea unui sistem securizat şi fiabil. Pentru a cuantiza adaosul de securitate obţinut prin aceste metode, diferite metrici au fost propuse, a căror aplicabilitate este de ordin general în securitatea sistemelor.

ABSTRACT

In this Ph.D. thesis is introduced a new parallel model, as a solution to existent problems appearing in the third wave of computers, namely ambient intelligence. The proposed architecture is supported by intelligent agents, being called intelligent grid. It was developed inside ACSA (Advanced Computing Systems and Architectures) Laboratory together with my colleague eng. Versavia Ancusa.

State-of-the-art concerns regarding intelligent agents based systems are showing the necessity of developing techniques for addressing the problem of security and dependability. A design for reliability and testability imposes as designing desideratum. In this regard specific methods are tested and employed, but also different solutions are provided. A very important feature of this intelligent grid is that the dependability of individual resources may not be able to be guaranteed. Giving the fact that every single resource is used outside of organizational boundaries, it becomes a real problem when guarantying that a resource being used is not malicious in some way. In order to perform a relevant simulation based assessment of architecture reliability, very accurate fault models must be realized. Therefore, it is necessary to perform a rigorous study and analysis of threats. Furthermore, the errors occurrence models are very important for the reliability assessment, thus an analysis of them is also required. Security and reliability can be achieved in terms of redundancy and the authentication of different exchanged messages can be realized by digital signatures based on hash functions. Other two methods are also proposed for intelligent grid in order to attain security and dependability. The first method aims at detecting and correcting different errors from the exchanged messages between agents. The second method provides targeted detection schemes for different types of attacks from a dictionary attacks. More than this the experimental part of this last method revealed that a certain feedback polynomial is the most appropriate to be used in order to detect the attacks from the developed attacks dictionary. Nevertheless, in order to test the offered solutions, relative risk measures are presented which appliance is not limited to the subject of the proposed architecture.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

> "Security engineering, especially in this *third wave*, requires you to think differently. You need to figure out not how something works, but how something can be made to not work. You have to imagine an intelligent and malicious adversary inside your system, constantly trying new ways to subvert it. You have to consider all the ways your system can fail, most of them having nothing to do with the design itself. You have to look at everything backwards, upside down, and sideways. You have to think like an alien." Bruce Schneier [112]

1.1. Motivation

As presented in the 2008's first number of the Communications of the ACM, Bell's Law for the birth and death of computer classes maintains its viability and expands with additional predictions. The article begins with a walkthrough from 1950 to 2010. In the early 1950s, a person could walk inside a computer and by 2010 a single computer (or "cluster") with millions of processors will have expanded to the size of a building. More importantly, computers are beginning "to walk" inside of us. These ends of the computing spectrum illustrate the vast dynamic range in computing power, size, cost and other factors for early 21st century classes.

A computer class is defined as a series of computers, in a particular price range, with a programming environment and user interface for communication with other information processing systems and people. The common nature of stored-program computers is that one computer may be programmed to replicate the function of another class. Thus, one class may subsume or kill another class [28]. Computer classes die or are overtaken by lower-priced, more rapidly evolving computers. Another factor in analyzing the death and birth of computer classes is that generality always wins against a computer built for a specific function. According to Bell's Law of Computer Classes, a new generation of computers emerges every 10 years. The first generation (1950-1960) was composed of vacuum tube computers, while the second one (1958-1970) was marked by the invention of the transistor, thus being composed from transistor computers. The third class (1965-1985) was discernible by TTL and ECL integrated circuits. MOS and CMOS integrated circuits enabled the forth generation since 1971. This computer classes are grouped into waves. In the first wave (1965-1990), the meaning was centered on a single device, like mainframes; while in the 1990s, the second wave (1971- present) meaning was in the connection. Computation and information access in a distributed environment were the central point of interest.

The *third wave*, starting now, brings the collection of devices and their reactive interface into focus. This wave of computing represents a new way for the interaction between the electronics and the human individuals. Some interesting features implied by these are the implicit resistance to failure (if a component fails, the goal can still be accomplished) and the inclusion of goals and constraints into the interface.

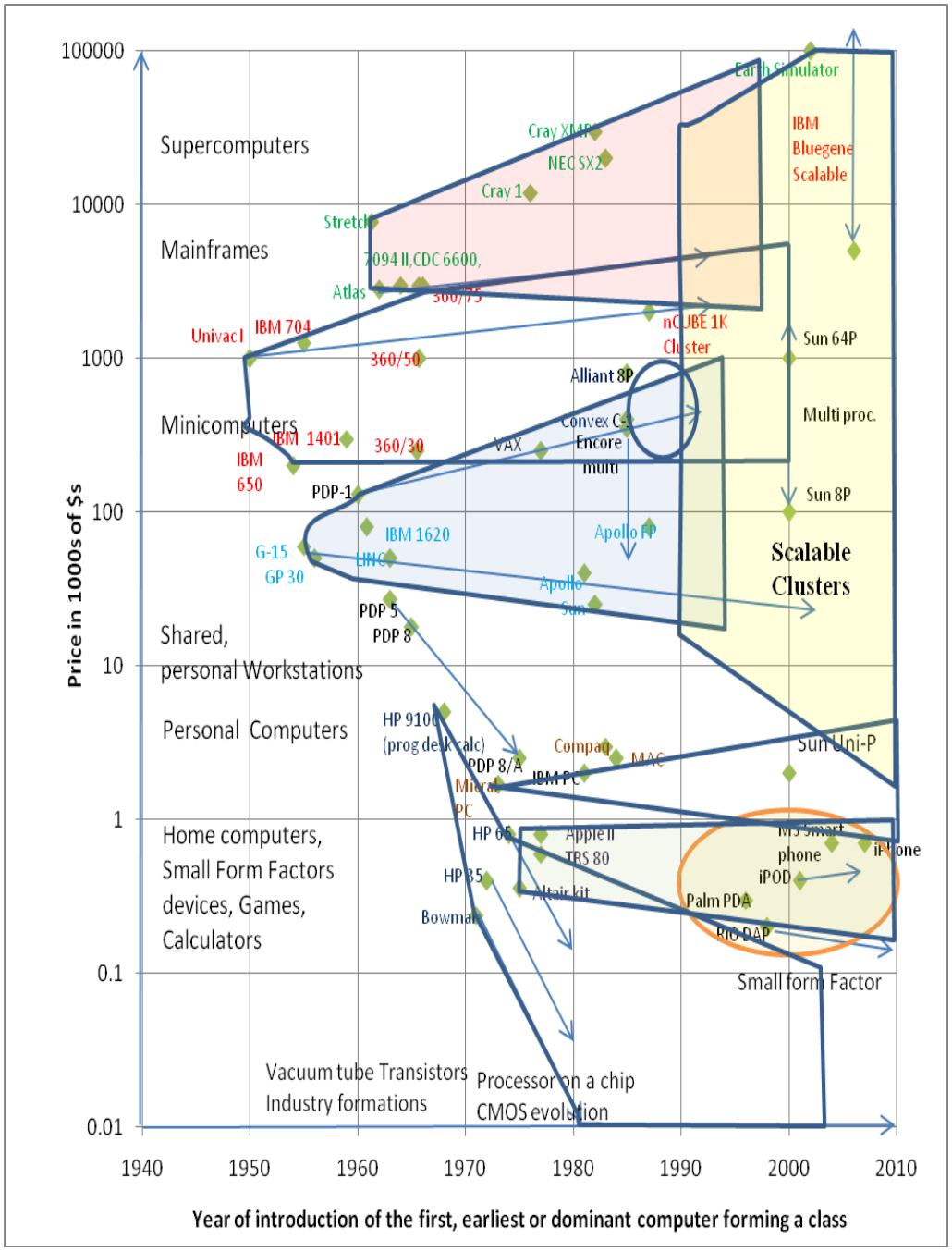

Fig.1.1. Introduction price versus date of the first or early platforms to establish a computer class or lower priced sub-class originating from the same company or industry [28]

It is expected that around 2010, this third wave should lead to the world-wide personal computer. Being such a new paradigm the problems it poses starting

with ultra low cost, ultra low power electronics for sensing and to ultra high speed signal processing for enhanced visual experiences at the other end. Ambient intelligence concept is part of the third wave of computers, being an environment where besides sensors, actuators and multimedia processing, another participant might appear, namely the PCs. This participant is usually not taken into consideration in a classical ambient intelligence network. Actually, the PC can be seen just like another node in the network. It can be used in order to provide computational power and therefore can be used for processing different operations. When the human user interacts with it, the PC can be seen as a merged sensor-actuator entity. Different problems like power consumption, portability, scalability and configurability, reliability, security have been identified lately in the literature [38, 95] for ambient intelligence.

Taking into consideration the wide-broad implications of these requirements, the overall term of dependability appears as a system's attribute to be assured. We intend in providing a solution so as to address the problems of ambient intelligence in the realm of a dependable and secured architecture.

The sub-problem of security is particularly of a tremendous significance. In the first number of the IEEE Security & Privacy magazine, Matt Bishop defined computer and network security, or cybersecurity, as being critical issues [52]. But just protecting the systems that hold data about citizens, corporations, and so on it is not enough. The infrastructure of networks, routers, domain name servers, and switches that connect these systems together must not fail, otherwise computers will no longer be able to communicate in a reliably manner. Several questions arise, such as what exactly the infrastructure is, what threats it must be secured against, and how protection can be provided on a cost-effective basis.

Security's components have been defined on levels. The basic level (let's call it Level 0) is identified in the form of requirements. Requirements define security goals. They answer the question, "What do you expect security to do for you?" Following next, the US National Institute of Standards and Technology describes the process of obtaining a secure system as progressing from "*having policies (Level 1) to having detailed procedures (Level 2), implementing these procedures (Level 3), testing compliance with and effectiveness of the procedures (Level 4), and finally fully integrating policies and procedures into daily operations (Level 5).*" Policy defines the meaning of security. It answers the question, "What steps do you take to reach the goal set out above?" Mechanisms or procedures enforce policy. They answer the question, "What tools, procedures, and other ways do you use to ensure that the above steps are followed?" These components exist in all manifestations of security. Security mechanisms detect and prevent attacks and recover from those that succeed [8, 9]. Analyzing the security of a system requires an understanding of the mechanisms that enforce the security policy. It also requires a deep understanding of the related assumptions and trust, which lead to the threats and the degree to which they may be realized. Such knowledge allows one to design better mechanisms and policies to neutralize these threats. This process leads to risk analysis.

The field of distributed systems arose at the junction of personal computers and local area networks. The research that followed from the mid-1970's through the early 1990's created a conceptual framework and algorithmic base that has proven to be of enduring value in all work involving two or more computers connected by a network — whether mobile or static, wired or wireless, sparse or pervasive. The requirements for such systems are the following:

- remote communication, including protocol layering, remote procedure call, the use of timeouts, and the use of end-to-end arguments in placement of functionality
- fault tolerance, including atomic transactions, distributed and nested transactions, and two-phase commit
- high availability, including optimistic and pessimistic replica control, mirrored execution, and optimistic recovery
- remote information access, including caching, function shipping, distributed file systems, and distributed databases
- security, including encryption-based mutual authentication and privacy.

The distributed network is seen as a support of second wave computers used in order to emulate third wave solution.

A last factor of motivation is stated so firmly in [5]: new technologies like nanosystems, biochips and quantum computing, as well as new concepts of man-machines systems, like ambient computing, grid computing, require attention to specific dependability and security issues.

1.2. Thesis goals

The first goal of this thesis is to propose a networked-based architecture in order to come with a solution to the necessity of raising the abstraction level existing in ambient intelligence. The solution is supported by intelligent agents and the architecture is called intelligent grid. A very important feature of this intelligent grid is that the dependability of individual resources may not be able to be guaranteed. Giving the fact that every single resource is used outside of organizational boundaries, it becomes a real problem when guarantying that a resource being used is not malicious in some way.

In order to perform a relevant simulation based assessment of architecture reliability, very accurate fault models must be realized. Therefore, the next major goal of the proposed report is to perform a rigorous study and analysis of threats. Furthermore, the errors occurrence models are very important for the reliability assessment, thus an analysis of them is also required. The expanding profile of intelligent grid requires a reliable security system, capable of responding to any attack on resources. This can be achieved by developing a comprehensive threat model. Without such a model security designers might concentrate their efforts on some threats while leaving the system vulnerable to others. In this regard, we aim at building a wide-ranging threat model.

The process of communication between agents in the intelligent grid is based on messages. Whenever there is a message exchange between agents, a malicious attack can occur and the message can be corrupted. The third goal aims at studying and providing a reliable method of correcting these different message errors. In this regard, the security of the architecture can be further improved by detecting and correcting the errors introduced by a possible attack via a transmission channel. Nonetheless authentication problems are to be considered at message exchange.

A fourth goal of this thesis is to provide detection schemes for different attacks on the intelligent grid network. The idea presented in this report is to identify peculiar attacks, each attack being identified by a certain signature. The method implied for this task offers a tailored detection scheme for each particular

attack. Each signature is associated to a type of attack, the attacks being organized in a dictionary attacks.

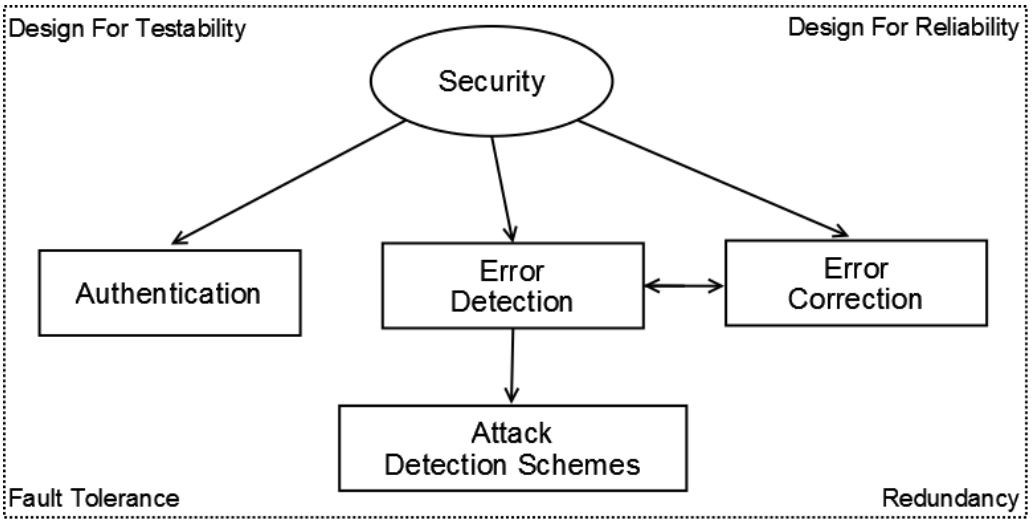

Fig.1.2. Security mechanisms and designing principles

The next goal is to study which feedback polynomial is the most optimum to be used in order to construct the targeted detection schemes. In this regard, different metrics are being employed based on which measure, a certain feedback polynomial should be identified as being the most optimum for the proposed goal.

Giving assurance that such type of network belonging to the third wave of computers, but being emulated on a second wave of computer, in a distributed environment, is *a reliable one* is a very important problem in terms of accessing resources, safe communication between nodes. In this regard a *design for reliability* is required [36]. Reliability measures those deviations that can appear between the system and the specification [113]. Security involves a subspace of reliability, but in those points that make the cost of deviation per unit time as very high. Therefore another motivation for addressing a security problem is that one of *cost*. Security and cost are inversely proportional: in order to halve the vulnerabilities, the investment in security should be doubled. The costs of security errors can be measured in terms of effects and are counted in billions of dollars [113, 114]. In order to quantify the improvements of the different solutions, we aim at following a *design for testability* based on relative risk measures. The security directions to be addressed, doubled by the designing principles with the corresponding mechanism are presented in Fig.1.2.

One of the most important goals is to publish the ideas of this report in state-of-the-art conferences, so as to demonstrate the high value of the presented ideas and employed techniques in intelligent grid for attaining security and dependability by means of fault tolerance.

The final overall and most vital goal of the present thesis that urged into fostering a security and dependability challenge, is *to address the complete designing of a secured system* based on the security levels presented above.

1.3. Thesis outline

The thesis structure is presented as follows. Chapter number two is presenting the actual trend: the tendency is evolving from mixing all kinds of equipments into mixing all kinds of networks, and extending functionality and reliability. The concept of ambient intelligence and present problems is introduced. Ambient intelligence is designed to be used into an environment suitable to be controlled; therefore its primary target is the ”smart” home. Different problems have been identified from the state-of-the-art literature and a solution is identified. This solution is fructified by proposing an architecture entitled intelligent grid in chapter three.

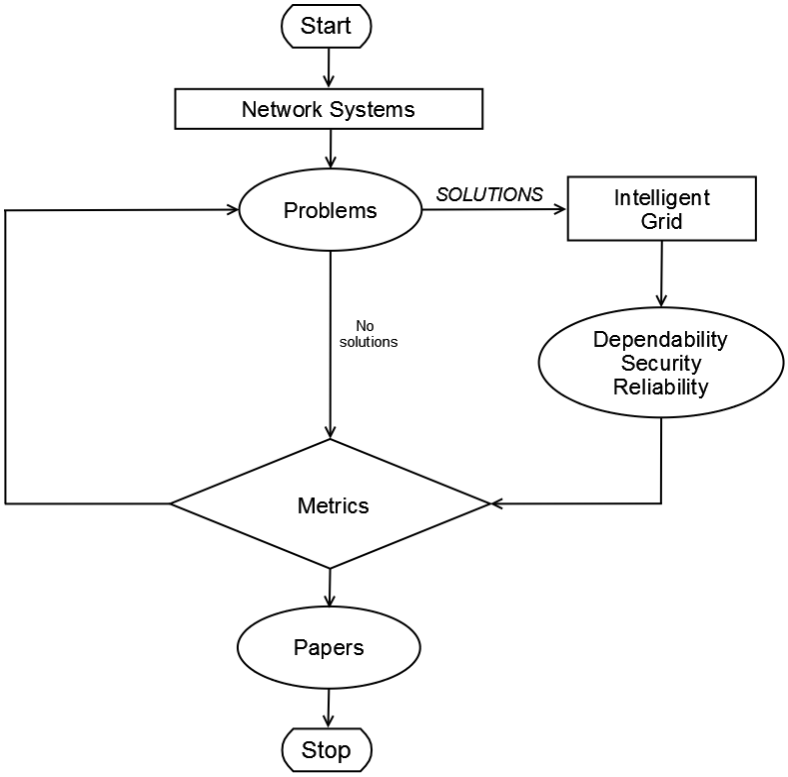

Fig.1.3. Thesis path

In chapter number four the problem of dependability and security is largely addressed. First of all the authentication problem at message level is taken into discussion and a solution is offered. Following next it is studied a technique based on General Linear Feedback Shift Register for detecting and correcting message errors when exchanged between agents. The following targeted point was the detection of different attacks on the intelligent grid network. Furthermore, dedicated schemes are being introduced for specific attacks, while the experimental part is relying on specific metrics. Peculiar improvements for the detection scheme technique fostered in obtaining detection schemes for all the attacks of a constructed dictionary.

Nontheless, in order to accomplish a thoroughly analysis of the security improvements relative risk measures were introduced in chapter five. The conclusions, published papers, thesis impact and future work are presented in chapter six.

In Fig.1.3 is presented the thesis path. In the case of networked systems (chapter two), ambient intelligence is part of the third wave of computers. Different problems have been identified and a solution has been proposed entitled intelligent grid (chapter three). The challenges of dependability with its subset of security, aiming a design for reliability addressed by fault tolerance techniques are further detailed (corresponding to chapter four). The gains of the proposed techniques are residing in doubled directions. First of all, are quantified by relative risk measures (chapter five), and secondly by the impact the proposed solutions, according to the published papers (chapter six).

2. NETWORKED SYSTEMS

2.1. Networks

The term "computer network" means [7] a collection of autonomous computers interconnected by a single technology. Two computers are said to be interconnected if they are able to exchange information. The nature of the connection among those two computers does not matter. It can be copper wire, fiber optics, microwaves, infrared etc.

Their architecture and dispersion changed throughout the years. In the 1980s the computer networks were only "an academic curiosity", while in the 1990s their usage extended from universities and large businesses to the daily reality for millions of people. In the mid-1990s, numerous kinds of LANs and WANs existed, along with multiple protocol stacks. By 2003, the only wired LAN in widespread use was Ethernet, and virtually all WANs were on the Internet. Therefore, interesting enough, the Internet is not a single network but a network of networks. The latest technological achievements in this domain proposed the wireless network.

There is a groundless confusion in the literature between a computer network and a distributed system. The significant peculiarity is that in a distributed system, a collection of independent computers appears to its users as a single coherent system. Typically, it has a single paradigm that it presents to the users. Usually a layer of software based on top of the operating system, called middleware, is responsible for implementing this model.

2.1.1. Sensor networks

A sensor network is a group of specialized transducers with a communications infrastructure intended to monitor and record conditions at diverse locations.

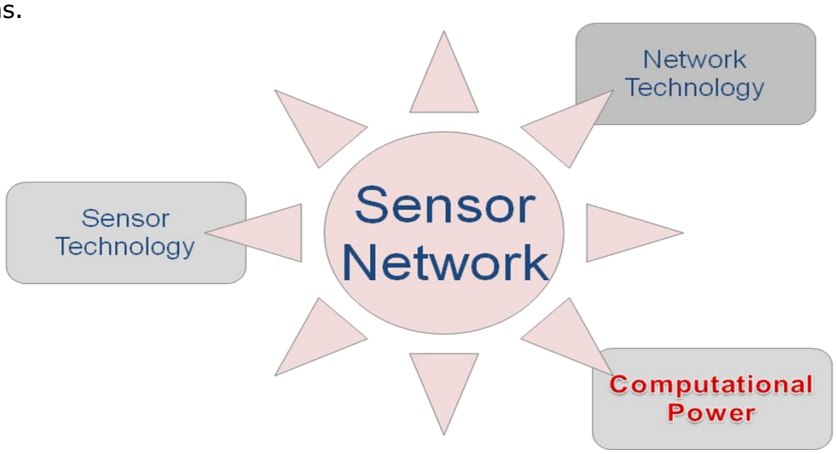

Fig.2.1. Involved technologies in sensor networks

A sensor network [16] consists of multiple detection stations called sensor nodes, each of which is small, lightweight and portable. Every sensor node is equipped with a transducer, microcomputer, transceiver and power source. The transducer generates electrical signals based on sensed physical effects and phenomena. The microcomputer processes and stores the sensor output. The transceiver, which can be hard-wired or wireless, receives commands from another computer and transmits data to that computer.

Table 2.1. Attributes of sensor networks [16]

Sensors	Size: small (e.g., micro-electro mechanical systems (MEMS)), large (e.g., radars, satellites) Number: small, large Type: passive (e.g., acoustic, seismic, video, IR, magnetic), active (e.g., radar) Composition or mix: homogeneous (same types of sensors), heterogeneous (different types of sensors) Spatial coverage: dense, sparse Deployment: fixed and planned (e.g., factory networks), ad hoc (e.g., air-dropped) Dynamics: stationary (e.g., seismic sensors), mobile (e.g., on robot vehicles)
Sensing entities of interest	Extent: distributed (e.g., environmental monitoring), localized (e.g., target tracking) Mobility: static, dynamic Nature: cooperative (e.g., air traffic control), non-cooperative (e.g., military targets)
Operating environment	Benign (factory floor), adverse (battlefield)
Communication	Networking: wired, wireless Bandwidth: high, low
Processing architecture	Centralized (all data sent to central site), distributed (located at sensor or other sites), hybrid
Energy availability	Constrained (e.g., in small sensors), unconstrained (e.g., in large sensors)

The power for each sensor node is derived from the electric utility or from a battery. The involved technologies in sensor networks are presented in Fig.2.1.

2.1.2. Multimedia networks

Multimedia content is playing an increasing part in business and private communications. These networks offer a best-effort service and should be able to meet the delivery requirements of interactive applications. Quality of service is also of great importance, and should be tailored to the delivery of multimedia content.

The term ”multimedia” refers to a spectrum of media classes used to represent information. Multimedia traffic represents the transmission of data acting as different media over communication networks. The spectrum of media is presented in Fig.2.2 and can be classified into three groups: (i) text, (ii) visuals, and (iii) sound. In Fig.2.2, the symbolic textual material, first of all may include the traditional unformatted plain text. It can also include formatted text with control characters, mathematical expressions, music scores and phonetic transcription of

speech or even other symbolic representations such as hypertext. The second category can include line drawings, maps, images and photographs, animation, simulation, video- and tele-conferencing, virtual reality objects. The sound category can include telephone/broadcast-quality speech to represent voice, wideband audio for music reproduction, and recordings of sounds such as electrocardiograms or other biomedical signals.

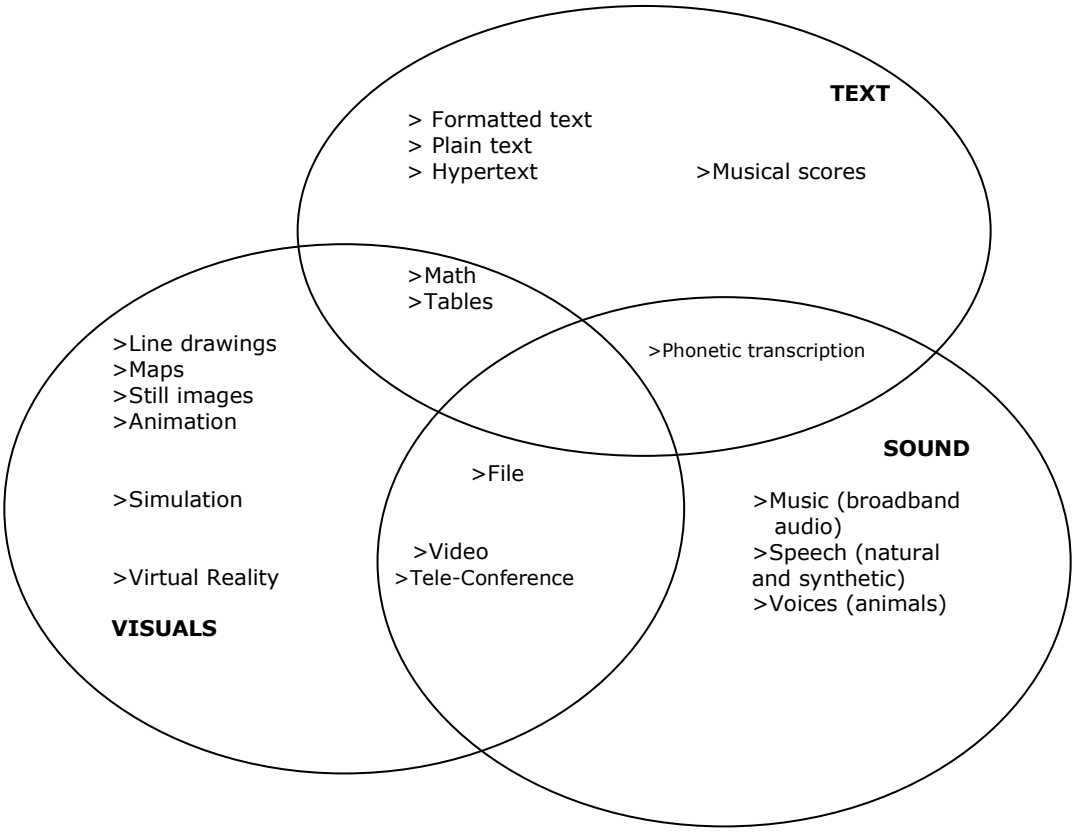

Fig.2.2. Diversity of Multimedia Data Signals [93]

Multimedia traffic represents the transmission of data acting as different media over communication networks. These networks offer a best-effort service and should be able to meet the delivery requirements of interactive applications. The quality of service metric is of critical importance, and should be tailored to the delivery of multimedia content.

2.2. Grids

Requirements of huge amounts of computing power are a necessity in a spectrum of domains. Scientists are analyzing terabytes and petabytes of data to provide better weather forecasting, develop more reliable models for detecting natural disasters, high energy physics and so on. Computing power is also required

in vast quantities in the life sciences industry for drug research. Financial industries require huge amounts of processing power to complete balance sheets, credit analysis, and so on.

In this regard, huge computing power is required in different industries. If we look at the computing resources available, we will discover that the laptops of today are as powerful as servers a decade ago. Moore's law, which states that computing power doubles every eighteen months, is valid even today and will probably be true for the next five to six years. With the advancements in the field of multi-core technologies, this growth can be extended further [27]. Therefore computing power, but in the same time the demand, is increasing. In this race, researchers have found an able ally in the form of networking. Between 2001 and 2010, while networking capabilities is supposed to increase by 4000 times, processing power is supposed to increase 60 times. This means that at the same cost 4000 times the same bandwidth will be available in 2010 as compared to 2001. Consequently, the computing architectures developed a decade back would probably require a rethink based on the technological progress in the fields of computers and networks. In the last decade we witnessed the development of a field called cluster computing in which different computing resources are connected together using a very high speed network like the Gigabit Ethernet or Infiniband.

In addition to the technological progress and the huge requirement of computing power, enterprises have also undergone a fundamental shift in Information Technology (IT) operations in the last few years. Enterprises are now witnessing increasing collaboration and data sharing among the different participating entities, resulting in the need and use of distributed resources and computing.

The double need of requiring more computing power and the integration of heterogeneous components into the IT infrastructure has led to the development of *grid technologies*. Initiated from the academic and the research community to accomplish their needs, it is gradually being adopted by the enterprises which have high computing needs like the life sciences, finance, and manufacturing industries. However, the widespread adoption of grid computing as an automatic choice in enterprises depends upon the ability of the researchers and practitioners in reducing the pitfalls that lie on the way. One such pitfall is security [8, 49, 80, 81, 82].

One of the earliest proposers of grid technology is Ian Foster of the Argonne National Laboratory. In 1998, in his book co-authored with Carl Kesselman and entitled "The Grid: Blueprint for a New Computing Infrastructure", Foster defined the grid as "*A computational grid is a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities*" [30]. Later on, Foster's definition of a computational grid has evolved. In a following article, The Anatomy of the Grid, co-authored with Steve Tuecke and which appeared in 2000, he transformed the definition to include elements of social and policy issues, stating that grid computing is concerned with, "*coordinated resource sharing and problem solving in dynamic, multi-institutional virtual organizations*" [33]. In a more generic sense, a grid is a "*hardware and software infrastructure that allows service oriented, flexible, and seamless sharing of heterogeneous network of resources for compute and data intensive tasks and provides faster throughput and scalability at lower costs*" [9].

It should be mentioned that grid is not a technology which has been developed from scratch. Actually it is an assembly of different existing technologies such as cluster computing, peer-to-peer (P2P), and Web services technologies. In

Fig.2.3 it is presented the evolution of grid computing technology from the clusters and P2P and the latest trend – adopting Web services technologies.

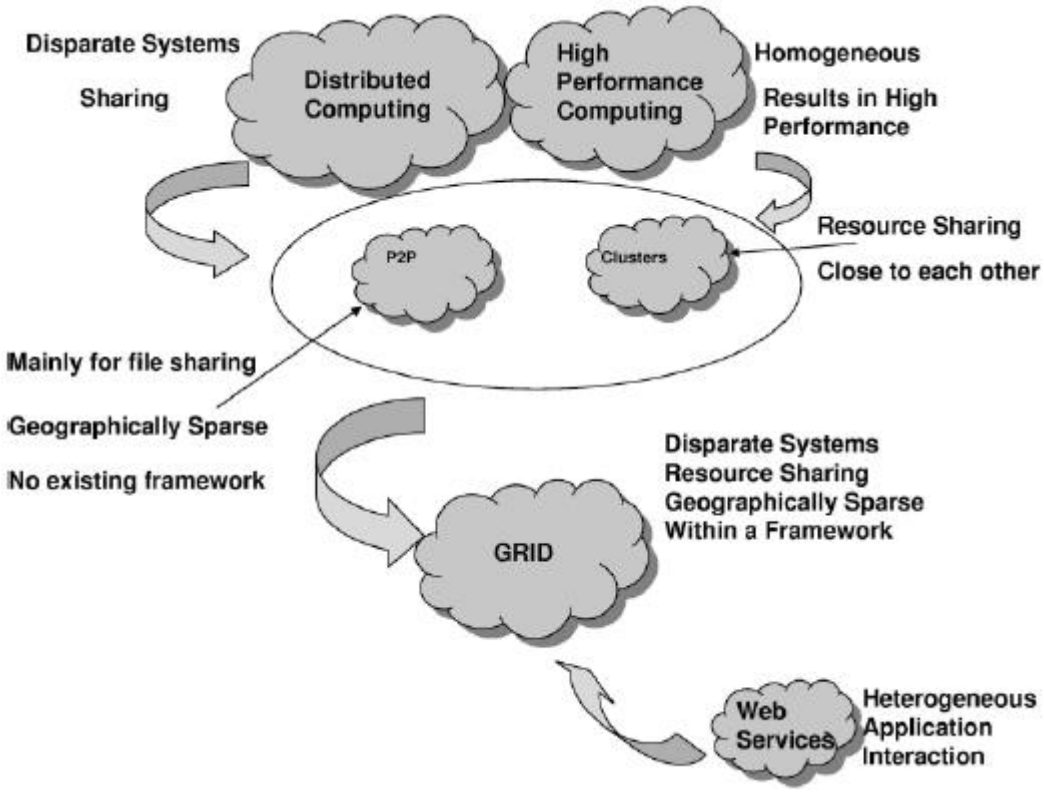

Fig.2.3. Evolution of grid computing [9]

Cluster computing and peer-to-peer computing have evolved from the distributed and high performance computing technologies [9]. In the case of cluster computing, different computing resources (machines, servers) are connected by high-speed interconnects in order to provide high performance. P2P can be defined as a class of applications that use resources available at the edges of the Internet. Peer-to-peer design requirements imply independence from DNS and an important autonomy from central servers. These constraints are due to the fact that accessing decentralized resources means operating in an environment of unstable connectivity and unpredictable IP addresses. Current P2P systems deal with many more participants than the grids. While the grids are focusing on the integration of important resources to deliver significant qualities of service within an environment of at least limited trust, the P2P systems offer limited and specialized services and have made few concerns about trust.

Compared to P2P applications, grid systems put together resources that are better connected, more powerful and more diverse.

In [31] it is presented that the idea which motivates both Grid and P2P computing - that of a worldwide computer within which access to resources and services can be negotiated as and when needed - will go beyond the range of research only when we are successful in developing a technology that combines elements of what we today call both P2P and grid computing.

Functionally, one can speak of several types of grids [44]:

- Computational grids (including CPU scavenging grids) which are mainly focusing on computationally-intensive operations.
- Data grids or the controlled sharing and management of large amounts of distributed data.
- Equipment grids which are mainly focusing on a primary piece of equipment e.g. a telescope, and the surrounding grid is used to control the equipment remotely and to analyze the data produced.

Mainly, there are four benefits of using grids:

- Performance and Scalability: even a small percentage in improvement results might lead to huge cost savings
- Resource Utilization: grid computing offers the methods to utilize the resources more efficiently through the process of resource sharing.
- Management and Reliability: grid computing provides a single interface for managing the heterogeneous resources
- Virtualization: the grid offers virtualization of heterogeneous resources implying a better management of the resources.

2.2.1. Grid computing concerns

In Fig.2.4 are presented the main concern areas of grid computing, namely application engineering, licensing, manageability and lastly, we come to the issue of security and dependability. Apart from the typical security challenges like authentication, confidentiality and integrity, a grid system offers other peculiar challenges, such as authorization, threats model, credential management [68, 49, 9, 8].

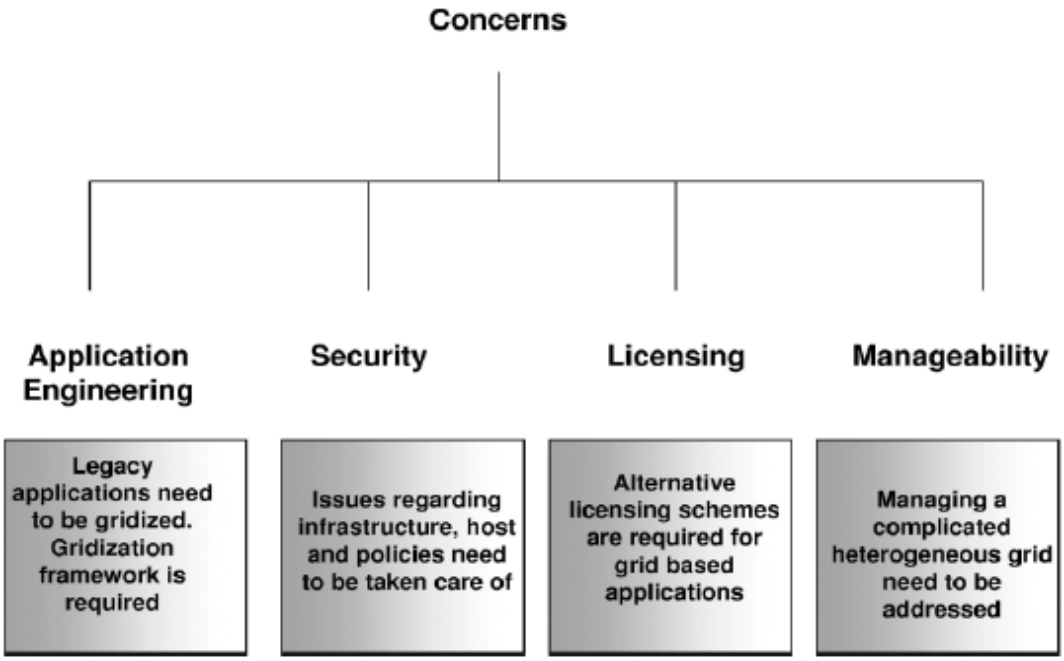

Fig.2.4. Grid computing issues and concern areas [9]

Table 2.2. Five major classes of grid applications

Category	**Examples**	**Characteristics**
Distributed supercomputing	DIS Stellar dynamics Chemistry	Very large problems needing lots of CPU, memory etc.
High throughput	Chip design Parameter studies Cryptographic problems	Harness many otherwise idle resources to increase aggregate throughput
On demand	Medical instrumentation Network-enabled solvers Cloud detection	Remote resources integrated with local computation, often for bounded amount of time
Data intensive	Sky survey Physics data Data assimilation	Synthesis of new information from many or large data sources
Collaborative	Collaborative design Data exploration Education	Support communication or collaborative work between multiple participants

In the evolution of grid computing, the efforts were concentrated on implementing a high performance distributed computational system. Aspects of security threats were overlooked. As the fame of grid computing is growing, so are the targets for the potential attackers. Therefore certain solutions should be adopted in terms of security and dependability issues, such as to provide the context of adopting grid as a widespread IT virtualization solution.

2.2.2. Computational grids

The average computing environment is still inadequate for computationally sophisticated purposes such as predicting the outcome of complex actions or selecting from among many choices. This is the reason for which supercomputers have continued to evolve. The term computational grid was adopted for the infrastructure that will enable the increases in computation discussed above. The classical definition of computational grid is given in [30], as being a hardware and software infrastructure that provides dependable, consistent, pervasive, and inexpensive access to high-end computational capabilities. The term infrastructure is used because computational grid is concerned with large-scale pooling of resources, whether compute cycles, data, sensors, or people. The pooling process implies important significant hardware infrastructure in order to achieve the necessary interconnections and software infrastructure to monitor and control the resulting ensemble.

Grid computing appears to be a promising trend for three main reasons [30]:

1) its ability to make more cost-effective use of a given amount of computer resources,
2) as a way to offer solutions to problems that can't be solved without an enormous amount of computing power,

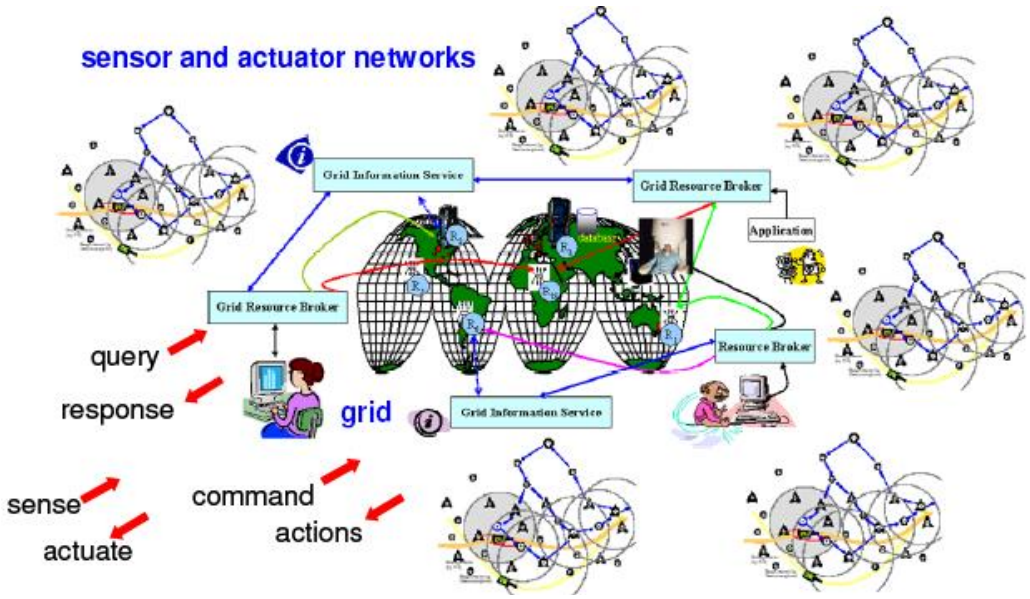

Fig.2.5. Sensor-grid architecture integrating sensor networks and grid computing [17]

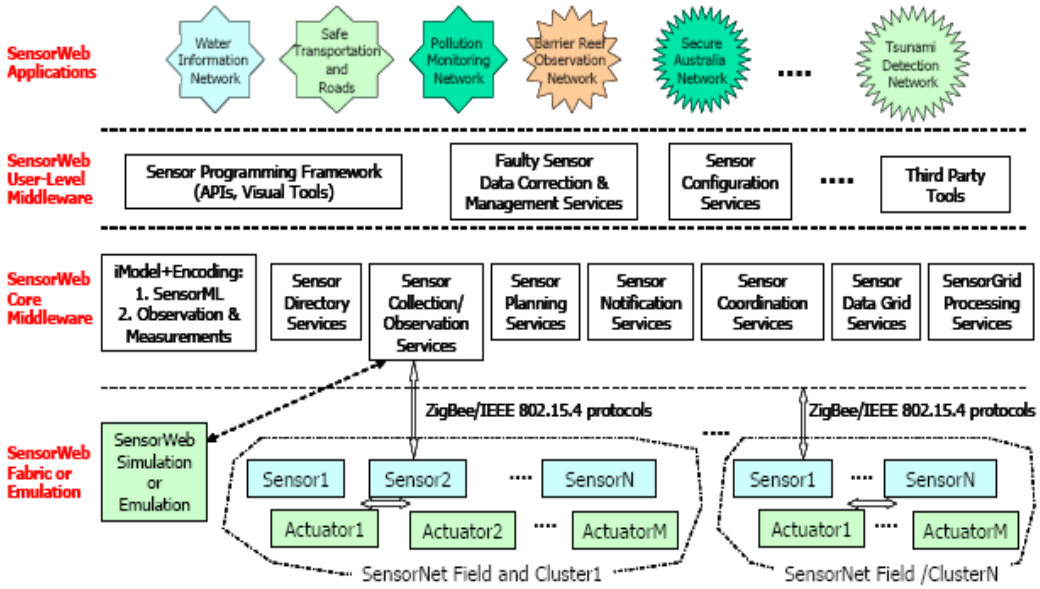

Fig.2.6. Open Sensor Web Architecture [17]

3) it suggests that the resources of different computers can be cooperatively harnessed and managed as collaboration in order to accomplish a common goal.

2.2.3. Sensor grids

In the last decade there has been a growth of the data-centric nature of grid computing applications. In this regard, the result is that newer grid applications are being developed which requires interaction with sensor and actuators for making an advantageous connection with the resource specific information. With the growth of sensor networking technologies as a different research area, scientists have worked on integrating the sensors and actuators with general purpose computing systems (Fig.2.5). As presented in [9], researchers are working on integrating grid systems with the sensor networks.

"Integrating sensor networks with computer grids is like giving "eyes" and "ears" to the computational grid" [17].

The processing and modeling of real-time data about phenomena in the physical world, with the power of a computational grid permits almost instantaneous response and decision on a large scale. Standard applications range from environment monitoring in order to be able to warn at the apparition of a natural disaster and to missile detection, tracking and interception. The idea of Service-Oriented Architecture (SOA) on the computational grid has been proposed as an upgrade to the initial architectures and middlewares, but in the context of sensor grids, this approach allows not only the discovery, access and sharing of the services, but also the sharing of the sensor-actuator infrastructure among a number of different applications and users. A proposed approach is the OpenSensor Web [17], whose architecture is described in Fig.2.6.

There are certain identified disadvantages. First of all, there are available different services, but only at middleware level. The application itself cannot develop its own service which can be accessed further on by another application, even if other applications can access the proposed services. It can be proposed at the middleware level a certain service, like a security plug-in, which can be later on used by other applications.

Another disadvantage is that the services are to be launched by the user. A certain program cannot launch many different instances of itself on other machines, it cannot change the working machine and also the interaction with other machine entities is limited. Nevertheless, there are security problems which should be addressed like identifying the threats in a threats model and also developing techniques for discovering the attacks [9].

2.2.4. Multimedia grids

Multimedia Grid (mmGrid) is presented as an extensible middleware architecture supporting multimedia applications in a grid computing environment [79]. The idea of this project is to provide support for applications from domains like graphics, visualization, streaming media and tele-immersion and also is to offer a mechanism for provisioning computing resources. The scheduling system is intended to be flexible, while the interactive and batch jobs have the capability to use grid-computing paradigm. The system's architecture is characterized by the fact that each service can run on a separate computer, even if it could potentially run on the same computer. These services are named to be authentication, registration, logging etc. What is very important to notice is that intelligent agents are employed in the architecture over the available resources. The tasks proposed in the protocols are carried away by intelligent agents. A software agent runs on every resource managed by the multimedia grid. One of the server's tasks will be to send messages

to the agents to perform actions required by the user. One agent can be designed such as to monitor the system it is running on. It can also ensure that certain Quality of Service (QoS) parameters are met and maintained over time. If these parameters are exceeding certain values, corrective actions can be taken by the agent.

Since mmGrid is focusing on graphics, visualization and streaming media applications and, in future, tele-immersion, the advance reservation of network bandwidth is a key requirement for good performance. Besides this, other basic requirements are the ability to submit batch jobs and to reserve a workstation for interactive applications.

2.3. Non-traditional parallel models

2.3.1. Ambient intelligence

A new paradigm – shifting technology started to emerge in the late years, as a third wave of computing and it represents a new way for the interaction between the electronics and the human individuals. As previously said, the third wave brought the collection of devices and their reactive interface into focus.

The name "ambient intelligence" is described [26] as a sensitive, adaptive, and responsive to the presence of people and objects environment where technology is embedded, hidden in the background and augments activities through smart non-explicit assistance. This environment should also preserve the security, privacy and trustworthiness while utilizing information when needed and appropriate. The common idea is that the human is in the center, the electronics invisibly in the background. The "ambient intelligence" paradigm differs in two major ways from the previous generations of computing [95]. First of all, the user interface has become reactive, that is actions are not explicitly requested but are the result of the mere presence of people or their avatars (of course, with their explicit or implicit goals and constraints). Secondarily, the meaning of computation can no longer be associated to a single device or a set of connected devices, but is located in the "collection of devices" (Fig.2.7). This means that the failure of a single component does not mean that the goal cannot be accomplished.

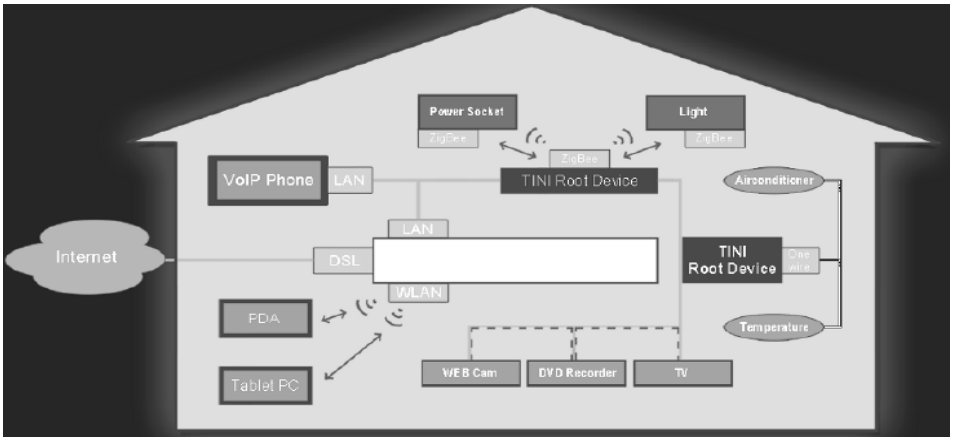

Fig.2.7. Ambient intelligence home infrastructure [104]

According to [38, 69, 95] there is a certain potential in merging sensor and actuator networks with multimedia networks, the result being a state-of-the-art ambient intelligence. Being such a new paradigm it poses problems starting with *ultra-low-cost, ultra low power electronics for sensing and ending with ultrahigh-speed signal processing for enhanced visual experiences*. From an economic point of view, the concept is highly alluring, because the demand is increasing in all the sub-markets computing, communications and consumer electronics.

Ambient intelligence is characterized by an environment

- where technology is embedded, hidden in the background
- that is sensitive, adaptive, and responsive to the presence of people and objects
- that augments activities through smart non-explicit assistance
- that preserves security, privacy and trustworthiness while utilizing information when needed and appropriate.

The purpose of ambient intelligence can vary from supporting human contacts, by offering him information and guidance, whenever necessary, to control of human health and security. Moreover, ambient intelligence will enable people to express themselves in ways that are unprecedented as natural interaction and augmented environments will become available. This may result in enhanced expressiveness, productivity, and well-being that may improve the quality of life substantially.

There have been identified specific and certain challenges [38, 69, 95] that should be addressed in ambient intelligence. These problems can be summed in four complex categories, namely

1. Power, cost and size
2. Portability, scalability and configurability
3. Reliability
4. Security and privacy.

The path to fully operational ambient intelligence is paved with a plethora of complications due to the fact that the research fields are heterogeneous, belonging to many disciplines. Ambient intelligence has recently started to being supported not only by the research groups, but also by industrial consortium and government agencies. All these efforts are aimed on offering new products and ways of exploring the full technological potential. This type of networking poses challenges from the quasi-infinite storage and computing power, to security, privacy, interoperability and the ease of use. Dependability and security of the devices, networks and systems, and trustworthiness of easy-to-use services delivered by providers of ambient intelligence will ultimately determine the acceptance of this type of network by the citizens. Such complex objective necessitates the cooperation at different levels and disciplines so as to reach the needed breakthroughs at technical level, but also the proofs offered to the society that such a solution is a dependable one.

Following next, it will be proposed a solution that can address the problems of ambient intelligence. This solution aims at respecting a design for reliability criteria, but in the same time maintaining the general context solution for ambient intelligence, namely raising the abstraction level [38].

3. INTELLIGENT GRID

It has been previously presented that the tendency in computing systems is evolving from mixing different types of equipments into mixing different categories of networks, and extending functionality and reliability. Ambient intelligence is designed to be used into an environment suitable to be controlled; therefore its primary target is the "smart" home [95].

Our point of view is that the three components, namely *general purpose computing systems, sensors and actuators networks and multimedia networks should all be connected in order to obtain a reliable ambient intelligence*. Therefore, into such an environment, besides the sensor, actuators and multimedia processing another participant might appear: the PCs. This participant is not taken into consideration in the ambient intelligence approach. On the other hand, today the multimedia ambient concept is centered on a PC approach (e.g. Microsoft Media Center, Apple FrontRow, and HP Digital Entertainment Center). The general purpose computing system can control the network. Providing the case this center fails, the whole network is down. The user is the system and configuration manager making this concept still connection and device oriented. The idea behind our approach is that the PC is just another node in the network. It can move (spatially), it can disappear or appear at will, or it can sleep. When it is awake and the human user interacts with it, it can be seen as a merged sensor-actuator entity. When it is awake it provides computational power, therefore the network can use it to process operations. Doing so, other nodes in the network can sleep, therefore reducing overall power consumption.

As demonstrated in [9, 38] sensor networks alone, though being an expanding market, can only monitor or automate a process. Multimedia networks provide, on the other hand, an explosion of input, output, storage and DSP devices. Classical computational grids provide high performance computing starting with low-cost components, composed into one network. State-of-the-art in providing computational power from a networked system has been acknowledged as being grid computing [3, 8, 9, 19, 30, 32, 33]. In order to achieve the prime target that is the "smart" home all these components must be assembled into a collection that works together for obtaining the final goals requested by the users. We propose that such a collection to be called *intelligent grid*.

Starting from the definition of ambient intelligence, it can be presented the definition of intelligent grid as being characterized by the fact that

- the *great majority* of technology is embedded, hidden in the background
- is sensitive, adaptive, and responsive to the presence of people and objects
- that augments activities through smart non-explicit assistance
- that preserves security, privacy and trustworthiness while utilizing information when needed and appropriate
- *it can accept explicit targeted tasks from the users*
- *it searches for computational power, being able to provide such a feature*.

The proposed approach of intelligent grid is compared with the previous networked systems in table 3.1. It is clearly shown that this intelligent grid includes not only the computation, but also the multimedia and sensor networks, thus being able to exhibit ambient intelligence and computation, virtually in the same time.

Table 3.1. Comparison of networks **[89]**

Network / Grid	Sensor grid	Ambient intelligence	Multimedia grid	Intelligent grid
Sensor network	X	X		X
Multimedia network		X	X	X
Computation network	X		X	X

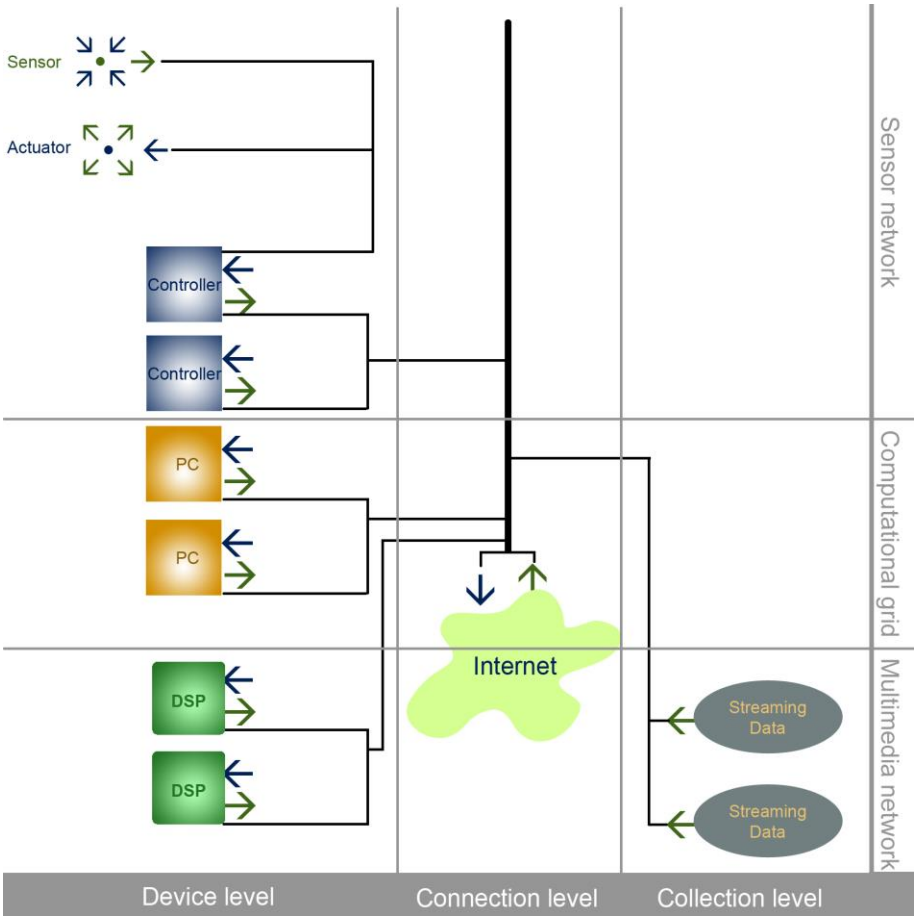

Fig.3.1. The intelligent grid **[88]**

The levels of this proposed model are illustrated in Fig.3.1. The pieces of equipment (sensor, actuators, controllers, PCs, DSPs) though extremely heterogeneous, are all grouped into the device level. The links between all this devices, as well as the association to the Internet are at the connection level, while the collection level is composed not only of streaming data sources but also of all the networks and their capabilities. In the same figure, the individual networks are also denoted, in order to clearly show the composition of networks into a higher, reactive network.

Giving assurance that such a type of network is a reliable one is a very important problem in terms of accessing resources, safe communication between nodes. In this regard a *design for reliability* is required. Each of the challenges of ambient intelligence will be further addressed in the context of intelligent grid, different solutions being offered so as to stick the principle of designg for reliability and by this, consolidating the position of surmounting the existent problems. One of the mechanisms used for obtaining reliable systems is the consensus issue. This approach is not the subject of this paper and it will be extensively studied in Versavia Ancusa's PhD report [107].

3.1. Power, cost and size

The power, cost and size approach, due to the heterogeneous nature of this model must be assessed by device: the sensors, actuators, controllers and DSPs are already very well on the way of becoming "disappearing electronics".

The main problem in "disappearing electronics" is that these super low-power devices should need no batteries or outside power supply, instead relying upon microgenerators. Therefore there is a need for devices which can harvest energy, called energy scavenging devices. Essentially, these devices can produce their own electricity from ambient sources. The free energy comes from solar, vibration, pressure and temperature gradients, as well as human power (solar obviously being the most well known and technologically developed of the bunch). Many solutions [94] emerged in the last years, though lately the idea of energy harvesting [36] is looked at from a system approach. All this is encouraged by the results of the industry [70], which prophecies that in 2012 the cost of solar photovoltaic will be $1.50 per watt, thus achieving the "magic number", which represents price parity with the electrical grid.

The area of nanogenerators is blossoming. Two main areas are developing in this area: nanoscale thermoelectric energy harvesting [71] and Nano-Piezotronics. The first area is based on a phenomenon called the Seebeck effect, permitting to turn temperature variance directly into electricity, based on thin-film thermoelectric devices.

As overviewed in [23] the field of MEMS (micro-electro-mechanical systems) is another interesting emerging technology, though from this field the most interesting recent approach is that of [101] converting nano-scale mechanical energy into electric energy by the means of nano-wires out of zinc oxide that create an electric charge when bent. A series of electrodes, in a zigzagging deployment is suspended over these tiny wires and when vibrations occur, the wires act like a brush over the electrodes, sending a stream of electricity.

Already some companies, such as Lightning Switch and Ad Hoc electronics, sell battery-free wireless light switches that convert the energy of a button push into a wireless signal. In [36] is showed that even a battery-free tire pressure sensors to be integrated into intelligent tires with cymbal transducers that convert impact acceleration into power for sensors. Such tires could measure all aspects of tire performance in real time.

This leaves the PCs to be analyzed from the power, cost and size methodology. The PCs are forecasted to disappear for a while now [13], and every major company (for example IBM (with the BladeCenter concept and its workstations) and Microsoft [11]) is preparing for it. The computer of the future

may have only an interface with the user, the main computation and tasks being located on a server, probably shared with other users.

3.2. Portability, scalability and configurability

Other problems which should be faced are represented by portability, scalability and configurability. Changing of the application usually means changing the software, which, of course, implies everything to fall apart. A real solution is considered to be the *raising of the abstraction level* [38].

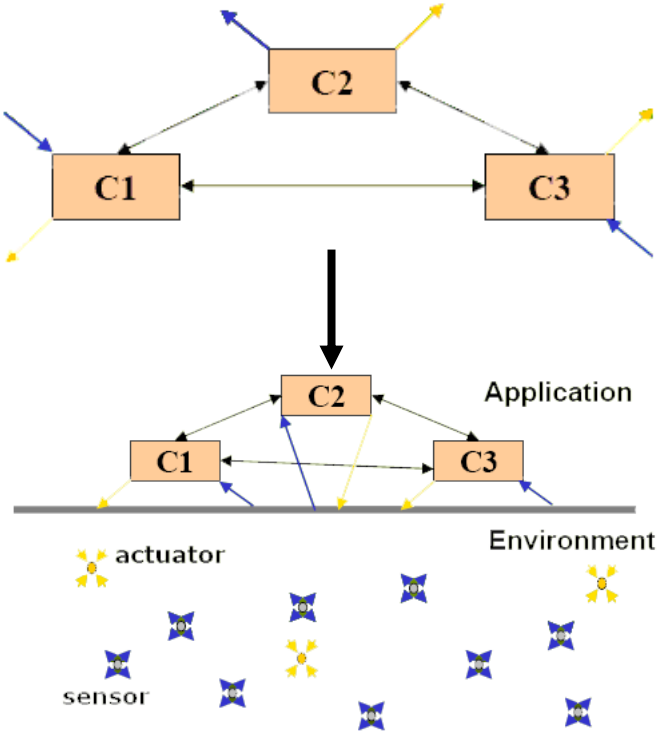

Fig.3.2. Raising the abstraction level [38]

In this regard, if we consider for example the sensor network, this particular type of network will be seen as a set of distributed computational functions aimed at achieving a set of common goals by interactions with the environment. This cooperation is established through a set of distributed sensors and actuators.

3.2.1. Middleware discussion

The need for middleware comes from computing environments where applications run on various types of computers, provided with different operating systems and software tools, interconnected by several types of networks. The purpose of middleware is to integrate these components efficiently and reliably in a distributed heterogeneous environment (Fig.3.3). A middleware provides a common

programming abstraction across a distributed heterogeneous system in order to facilitate the management of complex distributed systems.

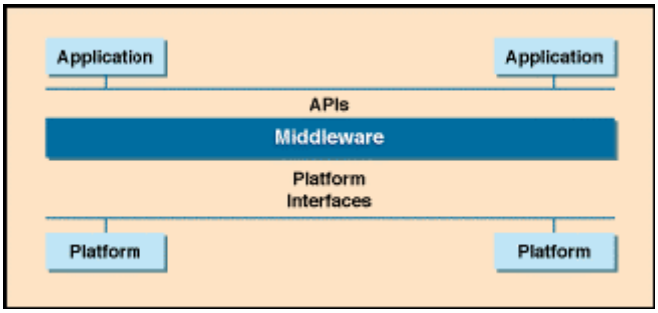

Fig.3.3 The middleware layer [72]

During the programming of any networks, first of all is a significant need for programming abstractions that simplify tasking, and for a middleware that supports such programming abstractions [43]. However, we must take into account the fact that our grid is composed of three distinct types of network, with their specific requirements. A summary of the programming requirements for each individual type of network is presented in table 3.2.

In choosing a middleware we must see if it can support all the requirements in their worst case scenario. There are already a series of middleware for every type of network, and in table 3.3 we present the types of approaches.

An area in which concepts from the area of artificial intelligence are passed into the field of distributed systems is Agent-Oriented Programming (AOP). The application in AOP is a collection of components called *agents*. The communication between agents is intrinsically peer to peer. It can be noticed that the only approach for middleware that covers all the requirements is mobile agents (Table 3.3). Among all the agent-oriented middlewares in use today the most widespread is JADE (Java Agent DEvelopment framework). JADE is a completely distributed middleware system [24] with a flexible infrastructure allowing easy extension with add-on modules. The framework facilitates the development of complete agent-based applications by means of a run-time environment implementing the life-cycle support features required by agents, the core logic of agents themselves, and a rich suite of graphical tools.

Table 3.2. Overview of the programming requirements **[89]**

Programming requirements	**Networks based on**		
	Sensor	**Multimedia**	**Computation**
Concealed issues	hardware and distribution	hardware	distribution
Restricted Resources			
Energy	X		
computing power	X	X	X
communication bandwidth	X	X	X
Network Dynamics	high	medium	low
Scale of Deployments	N*(100…1000)	N*(10…100)	N*(10…100)
Real-world Integration			

	Time scale	X	X	X
	Location scale	X	X	
Collection and Processing of Data				
	Preprocessing	X	X	
	Aggregating data	X	X	X
	Local processing		X	X

Table 3.3. Types of middleware and their usage **[90]**

Type of approach	**Networks based on**		
	Sensor	**Multimedia**	**Computation**
Events	X		
Remote Procedure Call		X	X
Object Request Broker		X	X
Message-oriented			X
Databases	X	X	
Mobile (Intelligent) Agents	X	X	X

As JADE is written completely in Java, it benefits from the huge set of language features and third-party libraries on offer, and thus proposes a rich set of programming abstractions allowing developers to construct JADE multi-agent systems with relatively minimal expertise in agent theory. JADE was initially developed by the Research & Development department of Telecom Italia s.p.a., but is now a community project and distributed as open source under the LGPL license. JADE is consistent with the FIPA specifications and IEEE standards, thus using different content languages and managing of conversations through predefined interaction protocols.

3.2.2. Intelligent agents

Agent technology has been the subject of extensive discussion and investigation [24] within the scientific community for several years, but it is perhaps only recently that it has seen any significant degree of exploitation in commercial applications. An agent is essentially "*a special software component that has autonomy that provides an interoperable interface to an arbitrary system and/or behaves like a human agent, working for some clients in pursuit of its own agenda*" [24]. Even if there can function an agent system based on a single agent, usually such architecture is composed of several agents, forming what is called multi-agent systems. The communication of these agents is realized by exchanging messages. Communication protocols enable agents to exchange and understand messages. Interaction protocols enable agents to have conversations, which for our purposes are structured exchanges of messages. This communication is established in order to achieve better goals in terms of response-time and accuracy for themselves or for the society/system in which they exist. Communication can enable the agents to coordinate their actions and behavior, resulting in systems that are more coherent. Coherence is the degree of which the system is behaving like a unit, without a form of explicit global control [105].

Multi-agent systems are being used in an increasingly wide variety of applications, ranging from comparatively small systems for personal assistance to

open, complex, mission-critical systems for industrial applications. Examples of industrial domains where multi-agent systems have been fruitfully employed include process control, system diagnostics, manufacturing, transportation logistics and network management. An agent can operate without the intervention of humans and owns the control over its actions and internal states. This feature is called autonomy. In order to achieve the goals of its tasks, the agent can cooperate with other agents or humans, being called social. A very important characteristic of intelligent agents is reactivity because it can perceive the external environment and responds to these changes in a timely fashion. An agent is able to perform goal-directed behaviors and it can take initiative in order to achieve its goal. Therefore is called proactive. It can also be mobile, being able to travel between different nodes in a given network in order to complete its task. In a distributed artificial intelligence approach, computational agents need to be distributed and embedded throughout the enterprise. The agents could function as intelligent application programs, active information resources, "wrappers" that buffer conventional components, and on-line network services.

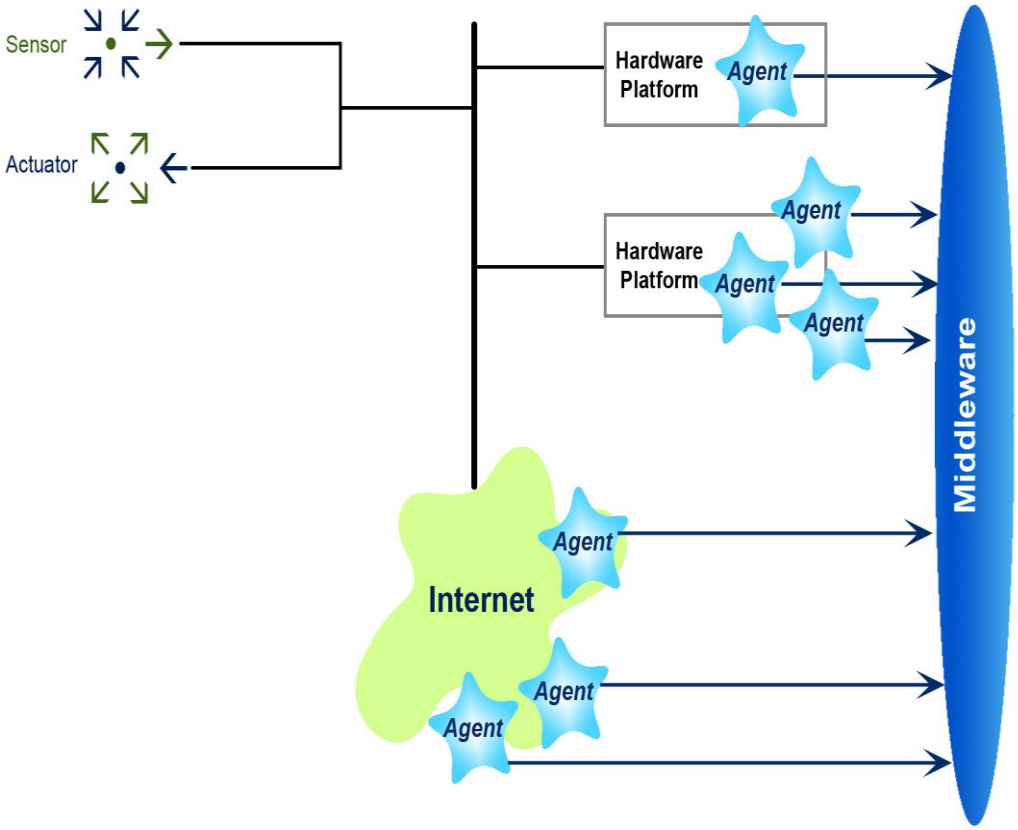

Fig.3.4. The intelligent grid invested with agents **[88]**

Using intelligent agents allows us to consider the controller/DSP/PC as an agent. The agents have several important features like autonomy, proactivity and

an ability to communicate. This allows them to execute *complex, and often long-term, tasks and to initiate a task even without an explicit stimulus from a user*. The communication allows an agent to interact with other agents in order to accomplish its own agenda. The architecture of the intelligent grid as described in Fig.3.1 on which the concept of intelligent agents is applied, is described in Fig.3.4.

Using this agent concept, at the device level we are left only with the sensors and actuators, at the connection level is present the middleware, while at the collection level we have the agents. The agents may run individually on a hardware platform, or may be more than one on a hardware platform. The interesting thing is that, raising the abstraction level, the physical aspect of the platform is not relevant. Our sole interest is in the agents, and the way they interact and solve the problems. The hardware platform on which an agent is located can be changed at any time, due to agent mobility. Another interesting factor is that agents are located on agent platforms, at logical level, and that agents can migrate between software platforms as well.

3.2.3. Middleware-based simulation for intelligent grid

In implementing the architecture presented in Fig.3.4, the first stage is to establish by simulation, the performance of JADE middleware in comparison with typical computational middleware.

Because the interesting part in the intelligent grid is the *computation power added to ambient intelligence,* the study aimed at illustrating how is performing a typical computational middleware, MPI, in comparisson to JADE. In order to do this, we modeled a simple problem. There are a number of computational nodes which process a number of inputs (S). The processing mode is very simple (of polynomial complexity). The number of nodes and the number of inputs can be varied and our scope is to measure the speed of the application on both middlewares.

As expected, the results when implemented with MPI, depicted in Fig.3.5 are of exponential growth. It can be seen that when the number of sensor inputs needing to be considered escalates, the overall time increases, and when the number of sensor inputs is constant, the communication among the computation nodes does increase with their number. We can dispute the fact that this increase in time even when S is constant is due to the scalability of this problem. This is not the purpose of this report. The goal was to study the modality in which a computationally classical type of middleware faces a new-age middleware like JADE.

In this regard, the same problem was implemented with JADE agents who launch CFP (call for proposal) message every time they need a value from a sensor. The agent that has the sensor directly connected replies with a PROPOSE of the value and if that value is accepted the initiating agent confirms that it received the value. The program was executed with one agent on every computational node, so that the comparison with MPI would be fair. The comparative results between the MPI and JADE implementation, for a constant number of inputs (40) are presented in figure 3.6.

It can be observed that for a few computation nodes, JADE overall time of execution is clearly worse than MPI. At 25 computation nodes, though, the overhead of communication in the MPI version allows the JADE version to win.

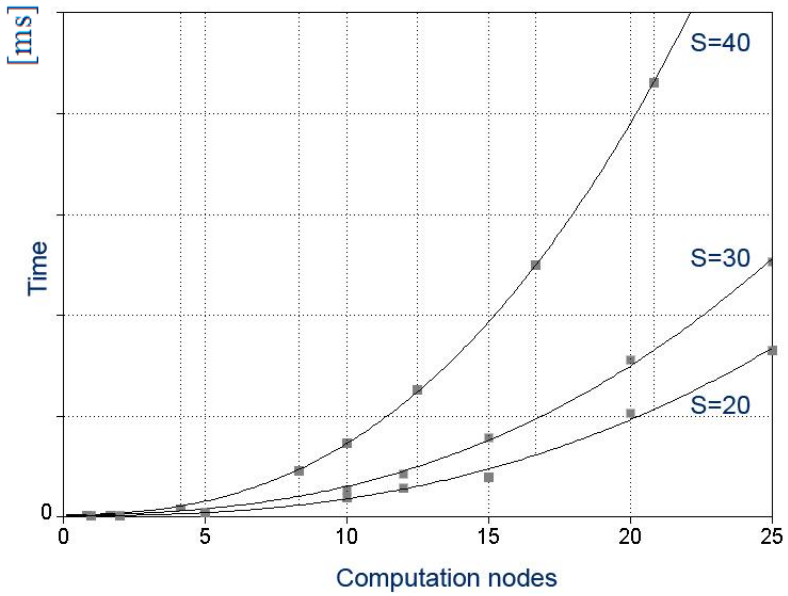

Fig.3.5. Results when implemented with MPI **[89]**

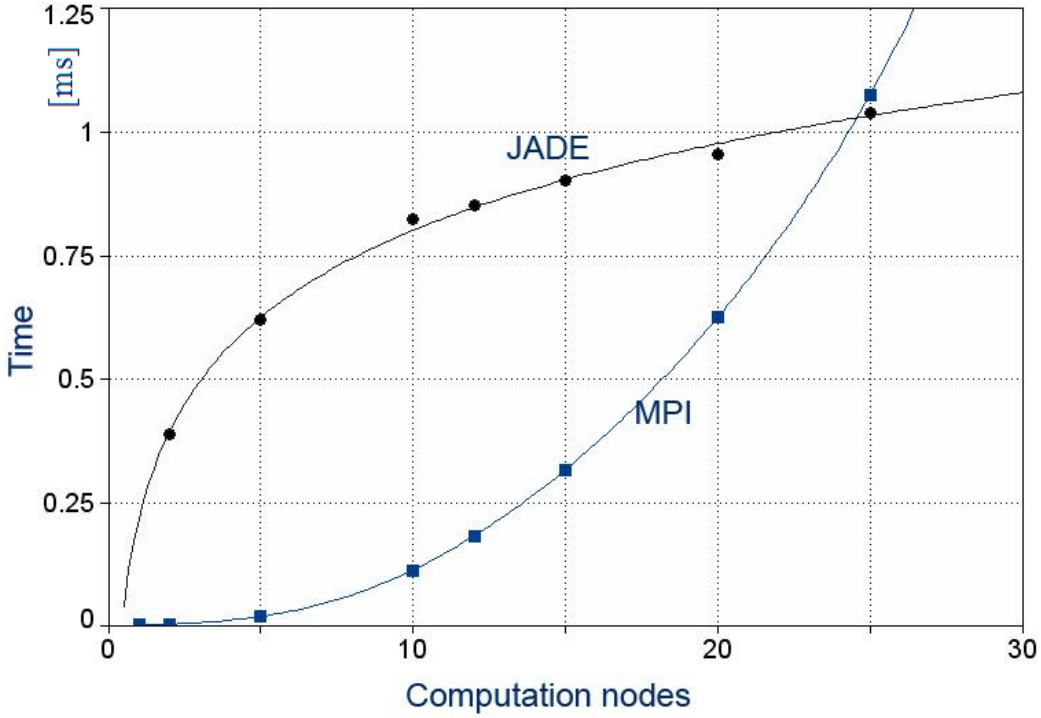

Fig.3.6. Comparative results (S = 40 = constant) **[89]**

For a simple problem, functioning on a relatively large number of computation nodes and required to intensively communicate with other nodes, the JADE version was expected to be slower than the MPI. When the communication

was over heading the classical solution, JADE proved to be faster, thus proving to be at least an interesting choice in implementing future intelligent type of grids. Also, the portability of the code makes JADE an interesting option.

3.3. Reliability

3.3.1. Redundancy

Reliability [5] is another major problem, due to the fact that unreliability is inherent to the disappearing electronics concept. This is primary caused by the fact that nodes may emerge unexpectedly, may move, may fail and may finish their energy reserves (temporary or not). All these problems are forced into focus by the cost, power and size constraints. Another reason why redundancy should be addressed is that it provides the security of a system [113]. Reliability can be achieved through three methods: redundancy, error correcting and detecting codes and consensus protocols.

This report will address the topics of redundancy and security and dependability (chapter 4). In this latter context, the solution based on error correcting and detecting codes will be addressed.

The solution used in ambient intelligence in order to achieve reliability is redundancy [38, 95]; therefore we expect it to be also a solution in intelligent grid.

Redundancy at sensing areas level

While the application potential of sensor networks is limitless, the construction of such networks is particularly challenging. One of the main challenges is to maintain long network lifetime as well as sufficient sensing area. Although sensors cost low power in general, their energy supply is for the moment very difficult. First of all, the high density of sensors wastes a lot of energy, but in the same time it provides opportunities to design energy efficient protocols. A broadly-used approach for reducing energy consumption in sensor networks is to turn off redundant sensors by scheduling sensor nodes to work alternatively based on heuristic schemes [4].

Reducing energy cost without introducing complexity in other parts of the system is still a tough problem. Even if turning off sensors is a commonly used strategy to reduce energy cost, this action might generate blind points and consequently, reduces the network's coverage range. For a given deployment area, the blind point are considered to be the regions that cannot be monitored by any sensors [4]. A sensor network provides the maximum sensing coverage when all sensors are powered on. However, keeping all sensors working will waste a lot of energy and thus reduces network lifetime. Because most of the applications may not require the maximal sensing coverage, it is critical to provide reliable heuristics for turning off sensors without degrading sensing coverage significantly in a statistical sense.

Redundancy at link level

As described in detail in literature, the failure of the transmission medium can not be separated from the failure of the component [46]. But if the same component is redundant linked to several other components we can overcome this particular difficulty. Then not only that we can determine the failure of the link or of the component, but we also improve the fault-tolerating capabilities.

The cost, implied by the extra wires can be justified in certain applications, in which the failure of the transmission medium has a much higher probability than the failure of the component itself. This aspect of redundancy was implemented on the intelligent grid.

Simulation results

The hypothesis in which the redundancy at link level was implemented is that the controllers and sensors are in a well known physical area. The sensors in the same area are connected to a controller, but sensor areas overlap in certain degree. An example is shown in Fig.3.7. This architecture leaves the marginal sensors with less coverage than the middle ones. A possible solution would be the ring connection of the sensors, but that is only possible in specific situations. Anyhow we shall concentrate on the middle part, in which the sensor redundancy/coverage is maximum.

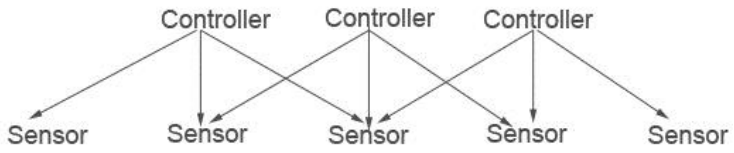

Fig.3.7. Redundancy at link level **[88]**

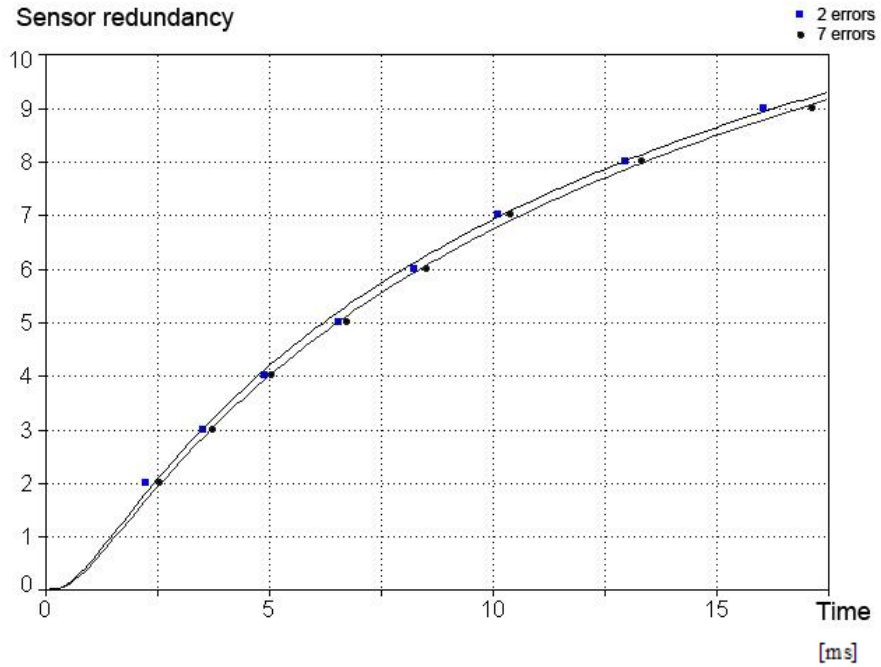

Fig.3.8. Sensor coverage variable, number of controllers fixed **[88]**

As previously justified, the used middleware is JADE. On every controller an agent requests, from time to time, to all the other agents all their read sensor values. The agents sent him those values, and the agent used all those values in order to do an action. For this experiment, that action was symbolic: setting a

variable [46]. The transmission wires between the sensors and controllers were supposed to be fail-stop, that is, if an error occurs, the line sends no value. The errors were injected in such manner as to affect as many sensor lines possible. The maximum number of errors in which at least one value of the sensor is received by any controller is:

$$Maximum\ errors = full\ covered\ sensors * (coverage - 1) \tag{3.1}$$

There are three independent variables implied by this architecture:

- Number of controllers
- Number of errors
- Sensor redundancy / coverage

The performance metric used to evaluate this variable was execution time. Fig.3.8, depicting the variation of time with the variation of sensor redundancy while maintaining the number of controllers invariable, shows that the number of injected errors has little effect on the overall execution time. The second graphic was made in order to show a zoom portion of the first figure where more data is depicted. The function best describing this variation is

$$y = (a + b\ lnx / x^2)^{-1} \tag{3.2}$$

We base this on the fact that the maximum r^2 measure of diversification was 0.9992 and the minimum value was 0.997 for a number of errors ranging form 0 to 7.

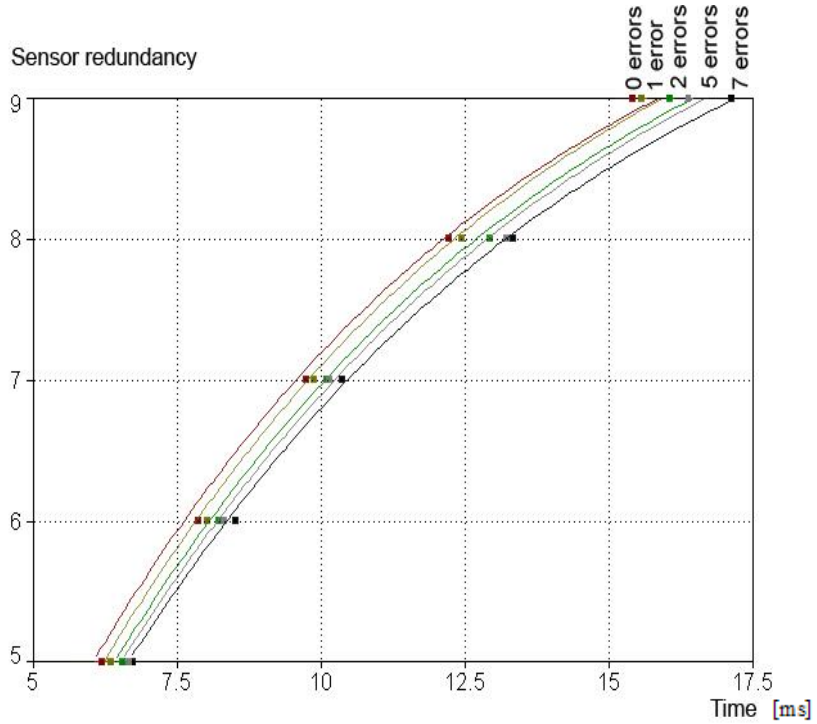

Fig.3.9. Sensor coverage variable, number of controllers fixed (9), zoom on the upper part of Fig.3.8 **[90]**

In Fig.3.10 are presented the results obtained by maintaining a fixed number of errors (1), while varying the total number of controllers.

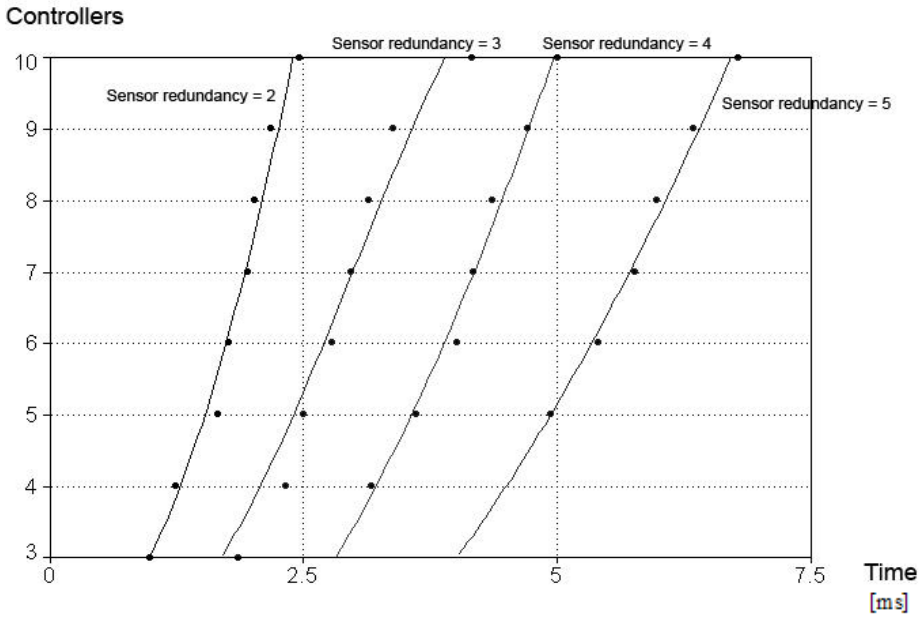

Fig.3.10. Number of controllers variable, number of errors fixed (1) **[88]**

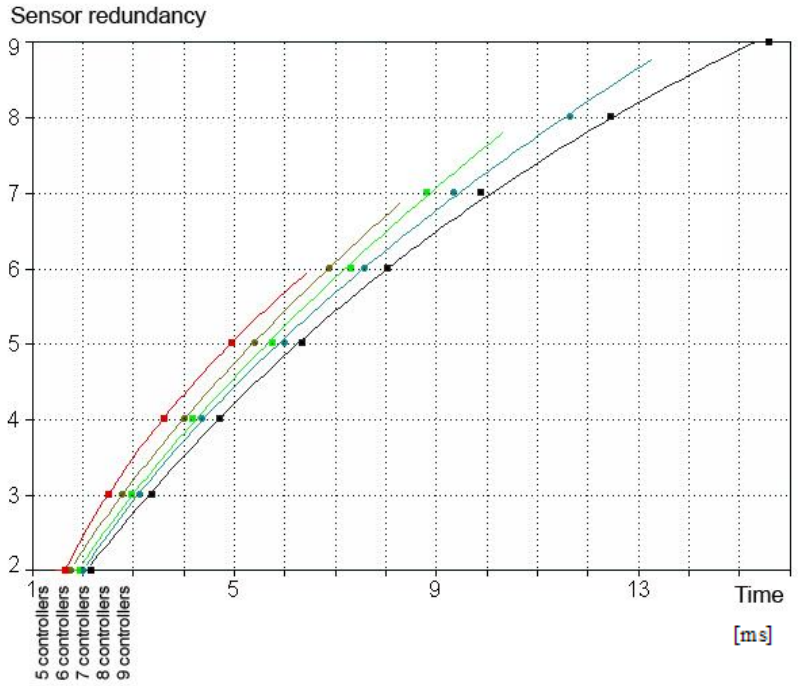

Fig.3.11. Sensor coverage variable, number of errors fixed (1) **[88]**

The effects of enlarging sensor redundancy are obvious, the time increasing with the growth of sensor coverage. It should be mentioned here that the function best describing all this variations is:

$$y=a+bx^2 \tag{3.3}$$

Fig.3.11 shows the way time varies when the total number of controllers is a parameter. The function best describing the variation is the power function (r^2 measure of diversification ranging between 0.9999 and 0.993). Some of the graphics are incomplete in the upper part because no coverage can be obtained, keeping in mind equation (3.1).

$$y=a+bx^c \tag{3.4}$$

The downfall of this technique appears when we consider the use of wireless networks. Not being any wires, we cannot apply link level redundancy. A solution to this problem is presented in the context of message redundancy. Even from 1985, Fred Schneider [25] suggested the use of consensus as a central paradigm for reliable distributed programming. Relative recently, in [106] an implementation was proposed by creating a generic and systematic way to transform various agreement problems into consensus, thus creating a unified framework to develop fault-tolerant agreement protocols in a modular, correct, and efficient way. The participants involved in the communication are largely described on [106]. The idea can be rephrased in the context of agents. In this regard, a consensus problem can be defined over a set of messages exchanged by the agents in intelligent grid. Every agent starts with an initial value v_i and the agents will have to decide on a certain common value called v. A more detailed presentation regarding the consensus problem in intelligent grid is to be found [107].

As a conclusion, it should be said that the redundancy at sensing areas implies the devices, our proposed model called intelligent grid implies the connection, while the consensus based protocol implies the collection (of messages). By using redundancy as a modality to attain reliability and security of intelligent grid, it is obtained a dependency between the maximum number of errors to exist in the network and the number of sensors which allows the system to function in a reliable manner. In this regard, it is demonstrated the fact that implementing this redundancy technique in intelligent grid, in order to attain reliability, is a viable solution so as to maintain the optimal functionality of the system even in the presence of errors. In other words, the proposed architecture aims at functioning in a fault tolerant behavior.

4. DEPENDABILITY AND SECURITY IN INTELLIGENT GRID

"Perfection is great if you can get it. But most of the time, we must live with less. Computing systems are as good an example as any: they aren't secure, yet we live with them." Fred B. Schneider [108]

4.1. Necessity of dependability and security

For thousands of years the fact that information means power is widely known. With the advent of computing age, the methods to store such information have been designed. It can be accessed whenever is needed from anywhere in the world by means of Internet connectivity. Almost every single computer in the world can be connected to each other through the Internet by techniques and solutions of networked information systems. Computers are used to store the most confidential data and failing to secure and safeguard such a system can turn our personal computer in the worst enemy.

Networked information systems have been identified as being one of the basic critical infrastructures of human society [84] because of the increasing dependence on such systems. The consequences of such an approach have been the concerns regarding networked systems security and dependability. More than this, a specific problem is regarded to the fact that state-of-the-art trends aim at mixing different kind of networking in order to provide higher functionalities [90]. Therefore, there is an urgent need to ensure the users that the provided trust of these information technologies and infrastructures offers a solid, *reliable* base [98].

The research in this area of high interest are focused on ensuring that a network information system, first of all, has a correct functioning in various operational environments and secondly, to provide the required protection of critical information (Fig.4.1). In order to stress even more the necessity of security and dependability, it should be said that the information theft via different types of attacks is a "*rapidly increasing problem*" [51] as recently studies prove it. Such an example is presented in Table 4.1 where two categories of malicious software are being surveyed. It can be noticed a high increase in the number of such applications in the year 2007 compared to the same period of 2006.

The most recently progressive ideas on computer innovation states that by 2040 or 2050, machine intelligence will surpass human intelligence, an event called by Raymond Kurzweil as "singularity" [15]. The problem which is raised in this context is to develop security techniques in the context of artificial intelligence solutions. It is admitted also that substantial efforts are to be employed in order to accurately identify abusive behavior. It should also be pointed out that the classical buffer overflows have been replaced by malicious scripts as the most common approach for penetrating systems. The goal of overcoming such behaviors is actually to obtain what is called a *reflective system*, namely a system that can reference and modify their own behavior in the context of a system malfunctioning.

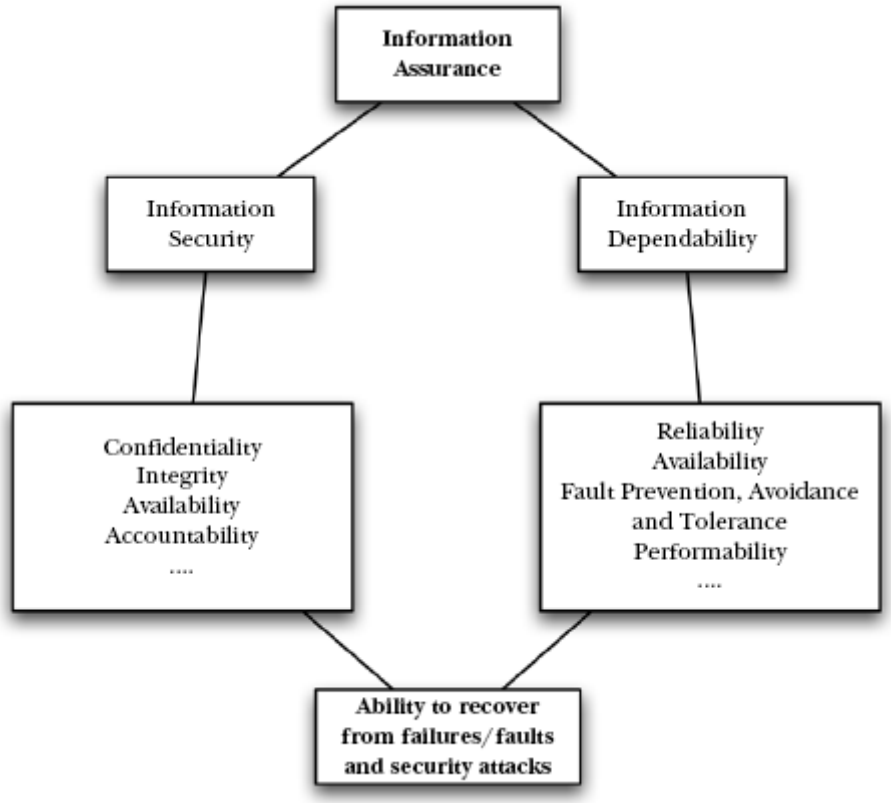

Fig.4.1. Information assurance: interaction between security and dependability [98]

Table 4.1. Increasing number of malicious-type applications [51]

	Applications		URLs	
Month	**2005–2006**	**2006–2007**	**2005–2006**	**2006–2007**
May	79	215	495	2100
June	154	212	526	2945
July	174	182	918	1850
August	168	172	958	2303
September	142	216	965	2122
October	154	237	863	1800
November	165	230	1044	1899
December	180	340	1912	2201
January	184	345	1100	1750
February	192	289	1678	3121
March	197	260	2157	1486
April	180	306	2683	1721
May	215	216	2100	3353

The security of the existing ambient intelligence model presents a particular challenge because this architecture combines different networks, being prone to vulnerabilities and attacks. In addition, these networks are generally deployed and then left unattended. Specific techniques are required to be studied in order to provide a fault-tolerant system. One way to address this goal is to apply different techniques in order to correct the errors introduced in the systems by a spectrum of

sources [61, 62, 65, 67]. In [29, 85] it is presented that sensor networks pose security and privacy challenges that will require new technological solutions. More than this, in the evolution of grid computing, the efforts were concentrated on implementing a high performance distributed computational system. Aspects of security threats were overlooked. As the fame of grid computing is growing, so are the targets for the potential attackers [9, 82, 96].

In [52], Matt Bishop defined computer and network security, or cybersecurity, as being "*critical issues*". Analyzing the security of a system requires, first of all, an understanding of the mechanisms that enforce a security policy.

4.2. Dependability, security and their attributes

The *function* of a system is what the system is intended to do and is described by functional specifications. Correct service is delivered when the service implements the system function. A *service failure* is an event that occurs when the delivered service deviates from correct service [5]. The period of delivery of incorrect service is a *service outage*. A deviation from correct state is called an *error*. The adjudged or hypothesized cause of error is called *fault*. Faults can be internal or external to a system. The prior presence of vulnerability, i.e., an internal fault that enables an external fault to harm the system, is necessary for an external fault to cause an error a possible subsequent failure(s). A fault is active when it causes an error, otherwise it is dormant. The mechanisms of faults, errors and failures are presented in Fig.4.2.

Dependability of a system is the ability to avoid service failures that are more frequent and more severe than is acceptable. It is usual to say that the dependability of a system should suffice for the dependence being placed on that system. The dependence of system *A* on system *B*, thus, represents the extent to which system *A*'s dependability is (or would be) affected by that of System *B*. The concept of dependence leads to that of trust, which can very conveniently be defined as accepted dependence [5]. The term dependability is used to encapsulate the concepts of reliability, availability, safety, maintainability, performability, and testability.

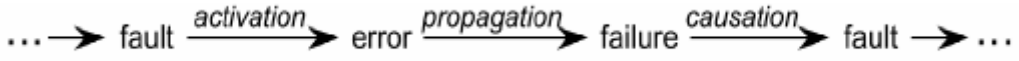

Fig.4.2. Chain of threats [5]

The *reliability* of a system is a function of time, $R(t)$, defined as the conditional probability that the system performs correctly throughout the interval of time, $[t_0, t]$, given that the system was performing correctly at time t_0. *Availability* is a function of time, $A(t)$, defined as the probability that a system is operating correctly and is available to perform its functions at the instance of time, t.

Safety is the probability, $S(t)$, that a system will either perform its functions correctly or will discontinue its functions in a manner that does not disrupt the operation of other systems or even compromise the safety of any people associated with the system. It is also a measure of the fail-safe capability of a system. The *performability* of a system is a function of time, $P(L,t)$, defined as the probability that the system performance will be at, or above, some level, L, at the instance of

time, *t*. *Graceful degradation* is the attribute of a system to automatically decrease its level of performance to compensate for hardware or software faults, allowing performance at some reduced level.

Maintainability is the probability, $M(t)$, that a failed system will be restored to an operational state within a period of time *t*. *Testability* is the ability to test for specific attributes within a system. *Integrity* refers to the ability of the computer system to ensure that the data is protected from unauthorized modifications.

Quality of service includes three quantifiable attributes:

1) availability of the correct service,
2) reliability of a correct service for a given duration of time, and
3) performance in terms of response time and throughput [92].

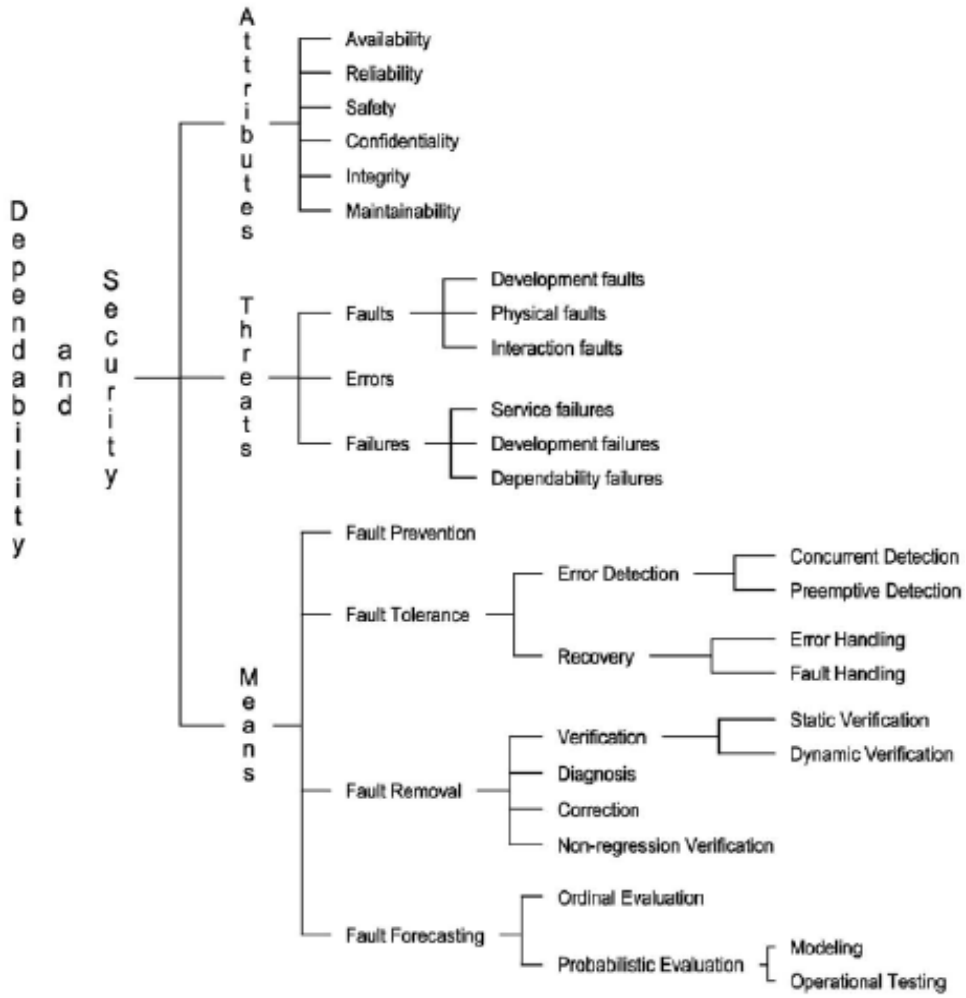

Fig.4.3. Dependability and security tree [5]

Security encapsulates attributes of authentication, confidentiality and integrity requiring the concurrent existence of 1) *availability* for authorized actions only, 2) *confidentiality*, and 3) *integrity*, as already defined. *Confidentiality* is the

absence of unauthorized disclosure of information. It is a service used to keep the content of information from all but those authorized to have it. *Non-repudiation* prevents an entity from denying previous commitments or actions. *Authentication* is related to identification and is applied to both entities and information: entity authentication and data origin authentication. Data origin authentication implicitly provides data integrity, in the sense that if a message is modified, the source has changed.

There are four major categories of means to attain dependability and security: *fault prevention*, to prevent the occurrence or introduction of faults, *fault tolerance*, to avoid service failure in the presence of faults, *fault removal*, to reduce the number and severity of faults and *fault forecasting*, to estimate the present number, the future occurrences, and the expected consequences of faults, errors and failures. A fault tolerant system is one that has the meanings to continue the correct perform in the presence of hardware failures and software errors [10].

4.3. Threats to dependability and security in intelligent grid

4.3.1. Faults

Physical and natural faults

This group of faults includes all fault classes that affect hardware [5].

In the field of networks that contain sensors, the physical attacks can be divided into two types [85], namely:

a) *Invasive Attacks*

This type of attack consists of reverse engineering followed by probing techniques that require access to the chip level components of the device. The result of such an attack is that the attacker will have access to the information from the chip and it might cause damage to the system.

b) *Non-invasive Attacks*

In this type of attack the embedded device is not opened and physically tampered with. An example of this type of attack is the side-channel attack. This latter type refers to any attack that is based on the information gathered from the physical implementation of a cryptosystem, in contrast to a vulnerability in the algorithms. In this regard, the attacker may analyze, for example, the power consumption, the frequency of the Electro Magnetic (EM) waves, the timing of the software operation execution. Through the power analysis, several key bits can be extracted. Other *side-channel attacks* which have not been explored in the context of networks that contain sensors are *timing attacks* and *frequency-based attacks*. The timing attack involves algorithms which have non-constant execution time and this can potentially leak secret information. Such execution time can be caused by conditional branching and some optimization techniques. Frequency-based attacks are able extract secret keys of symmetric cryptographic algorithms.

A certain discussion should be made in this point. As previously stated in [85], certain types of attacks were included in non-invassive attacks in the category of physical and natural faults. However, it can be noticed that all these attacks previously exampled occur during the use phase. They are all caused by elements of the use environment which is interacting with the system. We might say in this point that they are external. Because they originate in some human action in the use environment, we argue that a more accurate placement in terms of [5], is that

one that these are human-made faults, placed in the bigger class of interaction faults.

Some of the countermeasures for side-channel attacks used in traditional and embedded systems are:

- power consumption randomization,
- randomization of the execution of the instruction set,
- randomization of the usage of register memory,
- CPU clock randomization,
- using fake instructions,
- using bit splitting.

Further research is needed regarding these techniques applied in networks with sensors.

There are also threats to the transportation of data across the grid resources. The presence of security gaps intensify the security of the communication mediums. The security gaps are introduced in any secure path going through one or more middleboxes that need to perform some processing on passing data packets. These middleboxes include Network Address Translation (NAT) gateways, packet or content filters, proxy firewalls, and Wireless Application Protocol (WAP) gateways.

The resources in an intelligent grid are spread over the virtual organization and hence they have *distributed physical threats* such as temperature, humidity, liquid leaks, malicious intruders, accidents like fire, but also short-circuits or electrical surges, natural hazards such as earthquakes and floods. In [5] these kinds of threats are unified in a special category, named *natural faults*.

Human-made faults

There are two basic classes of human-made faults which can be distinguished by the objective of the humans interacting with the system during its use. In this regards, we can talk about malicious faults, introduced during either the system development with the intention to cause harm to the system, or during use stage, and non-malicious faults introduced without any malicious objective. From the point of view of the proposed architecture, our interest is to study the malicious faults.

In [5] it is stated that "*malicious human-made faults are introduced with the malicious objective to alter the functioning of the system during use*". In this case it's possible to talk only about the goals of such faults, which are:

- to disrupt or halt service, action which will cause *denials of service*,
- to access confidential information,
- to improperly modify the system.

In the case of intelligent grid, the analysis focuses on the so-called malicious logic faults. First of all, they presume development faults such as trojan horses, logic and timing bombs, trapdoors. Secondly, there are operational faults such as viruses and worms.

In this point, there is the need to talk about various *types of attacks* which can appear from a malicious logic fault in the case of intelligent grid.

Attacks on networks containing sensors can be put into specific categories, as it is presented in [85].

a) A mote-class attacker vs. a laptop-class attacker

A mote class attacker has access to a few motes with the same potential as other motes in the network. A laptop class attacker has access to more powerful

devices, such as laptops. This will give the adversary an advantage over the sensor network since it can launch more serious attacks.

b) An insider attacker vs. an outsider attacker

An outsider attacker has no special access to the sensor network, such as passive eavesdropping, but an insider attacker has access to the encryption keys or other code used by the network. In this case, an insider attacker could be a compromised node which is a legitimate part of the sensor network.

c) Passive vs. active attacker

A passive attacker is only interested in collecting sensitive data from the sensor network, which compromises the privacy and confidentiality requirement. On the other hand, the active attack goal is to interrupt the function of the networks and degrade the performance. In this regard, the attacker might inject faulty data into the network by pretending to be a legitimate node.

There are specific attacks that make sensor containing networks vulnerable. Such attacks are interception, node hijacking, sybil attack, sinkhole attack etc.

1. Interception

In [9] it is stated that this is one of the easiest means of attack in a sensor network because the anatomy of the attack consists of eavesdropping or information gathering in a passive manner. The data streams passing between different sensor nodes and the base station can be intercepted by a laptop type attacker. Such an attacker has a passive nature therefore the information about the attack and the identity of the attacker can be hided. According to [85], an adversary is more malicious if it can change the contents of the information in order to cause confusion.

2. Node Hijacking

First of all, a node can be compromised and the resulting action is to extract secret information from that node. The solution for this case is to exclude the specific node. Secondly, a new node can be introduced into the network. This node can introduce malicious information, but also can consume a lot of network bandwidth.

3. Sybil Attack

In this case, the attacker poses multiple identities to other nodes in the sensor network; therefore it is an important type of attack in terms of routing in sensor network. Assume that a node A is posing as n additional nodes $A_1, A_2,\ldots, A_n$. When routing a packet within the network, one of the fictitious nodes may be chosen as the next hop resulting in unauthorized data access, additional latency, resource wastage, looping, and sometimes even network partitioning. Some type of authentication may reduce the probability of such an attack. Even so, this attack is difficult to detect and prevent, especially in the case that the attacker is of laptop type and is an internal member of the network. The measurements for these three attacks can be taken from the range of digital signatures because are all a form of *masquerade attack* [9].

4. Sinkhole Attack

This type of attack is intended to make a specific node attractive to the surrounding nodes in terms of routing algorithm. Generally such an algorithm works by the principle of least cost path. In this case, exactly this peculiarity is used: if the attacker can advertise that a certain node is the most appropriate one according to the algorithm, then all the packets can be diverted to it. In sensor network this is an important problem because it is a constrained bandwidth environment.

5. Wormhole Attack

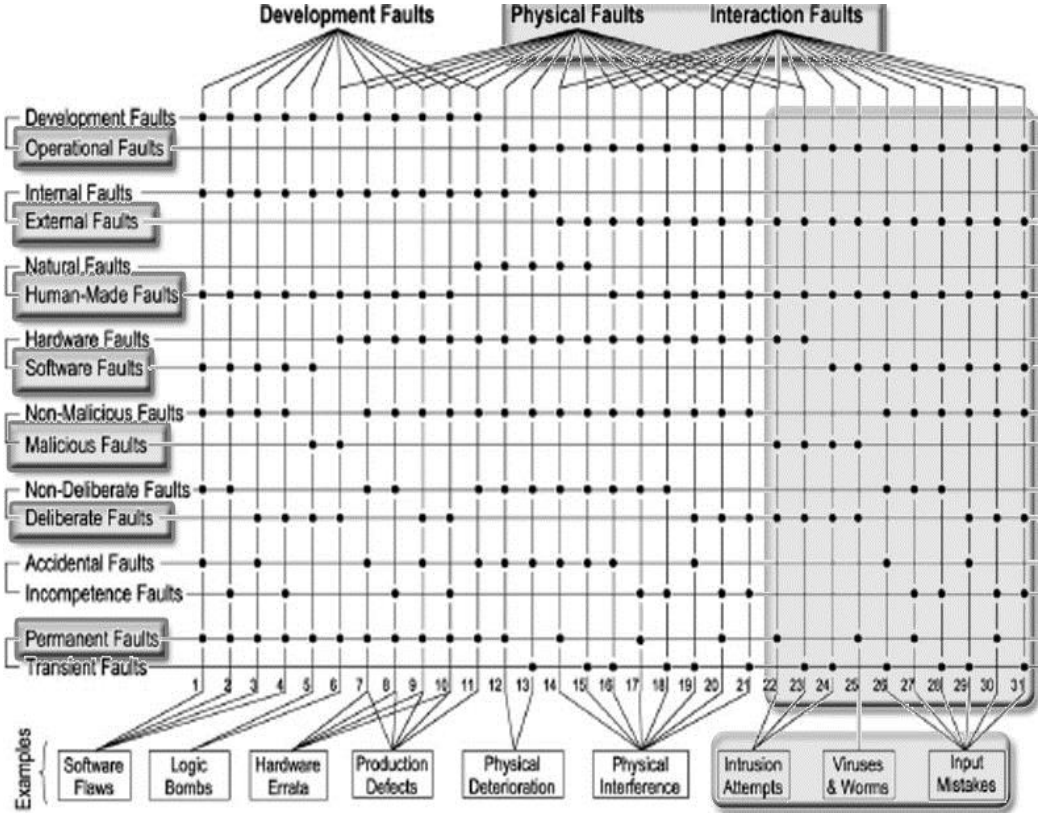

Fig.4.4. The classes of faults in intelligent grid **[63]**

The idea of this attack is that an adversary tunnels messages received in one part of the network because of a low latency link and after that will replay them in a different part. This attack can further create a sinkhole: the adversary on the other side of the wormhole can maliciously provide an artificial better route to the base station. The result of this activity is that the entire traffic will be drawn through the adversary if other routes are significantly less attractive.

6. Link Attack – Interruption

Present attack and the following four, are referring to routing table poisoning. Link attacks, in comparison with router attacks, are similar in case of link state and distance vector protocols. In the case of link attack – interruption, routing information can be intercepted by an opponent, so the information might be stopped of being further propagated. But this interruption is not effective in practice because the same information can be obtained from other sources. However, if the links are interrupted selectively, than the acknowledgements between neighbors will not prevent the formation of unsynchronized routing tables throughout the network. The results of this type of attack are looping and denial-of-service. There is also the case that a router drops the updates, but sends acknowledgment. In this case there will be an unsynchronized routing table. According to [9], this last problem has not been studied in the literature.

7. Link Attack – Modification/Fabrication

If an adversary has access to a link in the network, than routing information packets can be modified. Digital signatures are generally used for this problem. This is also a solution for distance vector protocols. These protocols are having a drawback in the form of extreme bandwidth consumption because the distance vectors are exchanged quite often. The efforts concerning the solution of digital signatures are taking in view the overhead introduced by digital signatures. Therefore efficient *digital signatures* are to be researched [9, 42].

8. Link Attack – Replication

Another routing table-related attack is the replication of old messages in which a malicious opponent has the control of routing updates and replays them with a delay. Giving the fact that the updates are valid, we cannot use digital signatures, but sequence information are generally used. This type of information can be in the form of sequence numbers or time-stamps. Therefore an update can be accepted only if the sequence number carries out a certain condition.

9. Router Attacks – Link State

If a router is compromised than it can add a false link, delete an already existing link (*proactive attack*) or the router ignores a change in link state of its neighbors (*inactive attack*). There is one solution based on a centralized attack analyzer module which detects attacks based on some possible alarm events sequences. This technique is called intrusion detection and it has been proved [9] that is not a scalable solution when connecting to the internet. The second solution is called protocol driven and the detection capability is embedded in the link state protocol itself. In this case a router won't believe an update unless it receives a confirmation link state update from the node which is sending the questionable link.

10. Router Attacks – Distance Vector

In this case a router can send wrong or even dangerous updates regarding a node from the network. In this case, digital signatures cannot be applied because the router itself is malicious. Validation scheme based on predecessor information have been developed, but more research is still needed in the area because most of the solution work on some constraining assumptions.

As a conclusion, we might comment on Fig.4.4. It can be seen that the proposed threat faults addresses 43.75% specific faults for intelligent grid. A viable solution should address as much as possible this percent of faults.

4.3.2. Failures and errors

From the failure domain viewpoint, in intelligent grid the failures might be distinguished as:

- *Content failures* because the content delivered by a service deviates from implementing the system function
- *Timing failures* because the time of arrival of the service content deviates from implementing the system function.

If there are both content and timing failures, there will be a *halt failure* of intelligent grid.

If the proposed system suffers service failures more frequently or more severely than acceptable, then a *dependability or security failure* will occur [5].

An error is that part of system's state that might lead to a failure. In the proposed architecture, starting from failures pathology we might encounter content errors and timing errors.

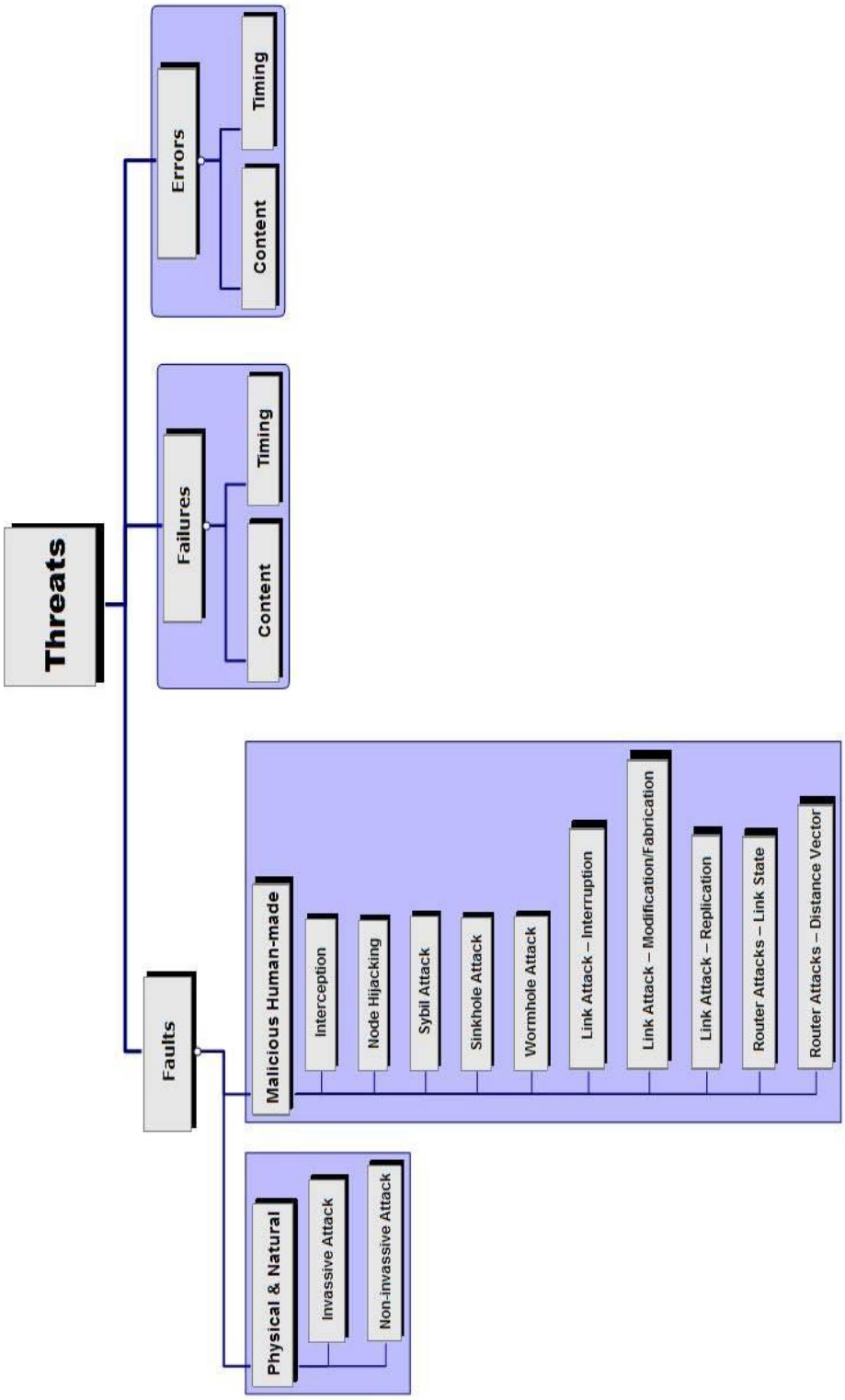

Fig.4.5. Threats model in intelligent grid

4.4. Authentication. Solution based on digital signatures

Our solution is based on *digital signatures* [9, 99] because asymmetric encryption (i.e., public key encryption) can be used to digitally "sign" a message, *M*, and in this way it will authenticate the sender. This means the receiver can be assured the message is actually from the sender, and not a third party masquerading as the sender.

This method will require a key pair. One key will be kept private by the sender, K_R, and the other that is distributed to the public, K_U. The entire message could be signed, or alternatively, a derived value is created from the message (e.g., hash code) and then signed by encrypting with the sender's private key. In this way the signature *S* is created. The message and signature are then sent together to the receiving party, who uses the public key to decrypt the signature.

In the next stage, the receiver creates the derived value from the message (via the same method as the sender) and the value is compared with the decrypted signature to verify that it was in fact sent from the sender.

This method authenticates the message in order to check if it was sent by the correct party.

In the terms of the owner – consumer paradigm, a possible way of attack would be to impersonate a consumer. This is the case of a Sybil attack. The mentioned action could lead to a possible transfer and execution of a malicious mobile agent or remote execution on the consumer container.

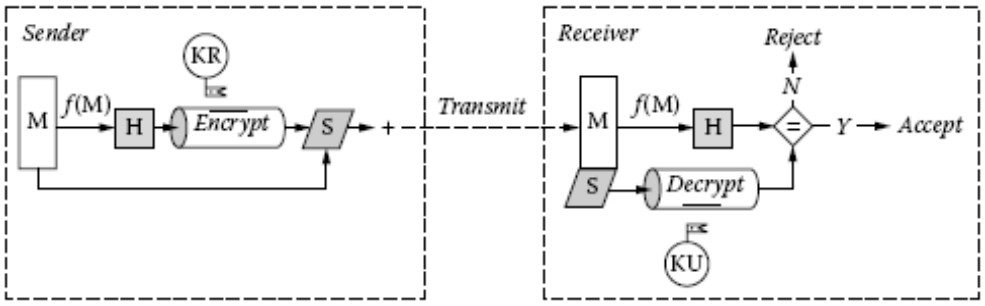

Fig.4.6. Message M is passed to the one-way hash-function *F*(*M*) to produce the hash value *H* [99]

A solution to this problem is found by adopting a broker. In terms of the proposed model, a broker is a special agent with the role of guarding the data traffic on the network. The broker is found through a negotiation process between the owner and consumer over a list of brokers obtained from the directory facilitator.

A digital signature is created by the broker, the following steps being performed:

1. The broker creates an asymmetric encryption key pair: B_R and B_U. B_U is then made known to the consumer and owner, identified through the directory facilitator.
2. The broker performs a scan on the mobile agent to be deployed (to its code and data and if necessary the state). If the scan is not successful the transaction is aborted.

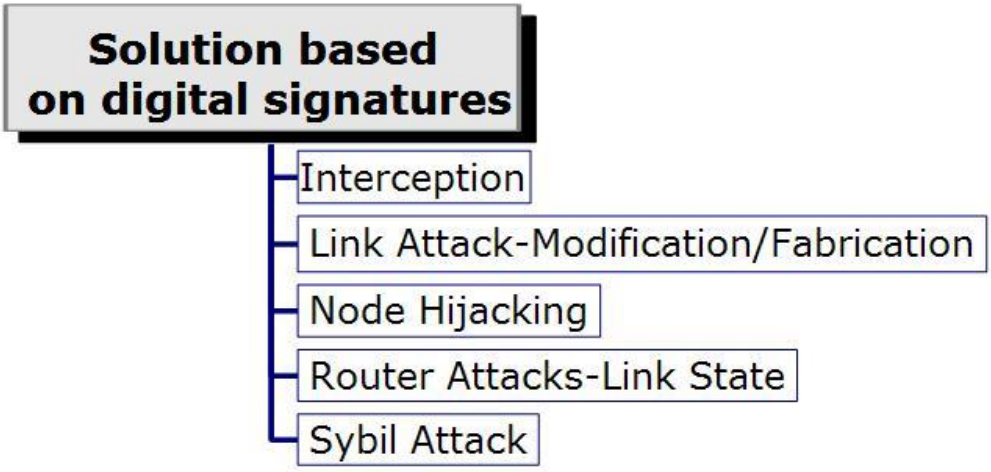

Fig.4.7. Attacks which can be addressed by digital signatures **[63]**

3. The broker creates the serialized code of the agent to be deployed, noted T_i.
4. The broker generates a hash code H of the serialized code T_i.
5. The broker produces a digital signature S by encrypting H with B_R.

Once this process is completed, the broker sends T_i and S back to the consumer because the consumer must transmit the agents, not the broker. Whenever necessary, the consumer transfers the agent, T_i, and the signature, S.

The owner then performs the following steps:

1. Decrypt the signature S with the public key B_U, thus recovering the hash code H.
2. The owner creates a new hash code H_1 of T_i. If $H = H_1$ then the node can accept the task with confidence that it was inspected by the broker. Otherwise, the task and associated hash value are discarded. If the task is accepted, then the hash value H is saved with the task in order to counter the attack.

We shall now explore how this algorithm can protect against the attacks mentioned before:

- Interception, Link Attack-Modification/Fabrication

  The attacker sees T_i and S. It does not alter any of them, but it can capture a sample of the serialized code and it can run it.
- Node Hijacking, Router Attacks-Link State

  If the hijacked node or router is the broker, the algorithm can be used in order to corrupt honest nodes. If one of the nodes is hijacked, and the broker is honest, the algorithm successfully detects an attack.
- Sybil Attack

  If the Sybil is the broker, the security is compromised. If the Sybil is one of the nodes, the attack can be detected.
- Sinkhole Attack, Wormhole Attack, Link Attack–Interruption, Link Attack-Replication

  These types of attacks are not to be solved by the digital signatures based algorithm [9]. The reasons for the last two ones were presented in their description

and in that case specific solutions were offered. The first two types of attacks are not detected due to the fact that they do not try to destroy data integrity, they only try to block or limit the bandwidth of the network.

Compared with the simple launch of an agent from one agent to another container, it only requires three more messages (one message in order to make B_U known, one message to upload the serialized agent, one message to send T_i and S back) and the computation time required by two applications of the hash (one at the broker, one at the consumer).

An important problem is broker's credibility. As previously shown, in case that the attacker is the broker, the entire network is compromised. We used a straightforward solution: the broker is the PC in the grid and there is only one broker in the whole network.

4.4.1. Experimental results

As mentioned in previous chapter, the platform on which the algorithm was tested runs the JADE middleware and had implemented a certain level of redundancy at sensor level. The coding/decoding was made using the presented algorithm. The hash coding was realized in two techniques:

- a dummy-algorithm that only returns the value and is useful in order to experimentally measure the complexity of the proposed algorithm excluding the complexity of the hash algorithm.
- a linear complexity type algorithm, in order to evaluate the impact of the hash algorithm to the overall run time. No message loss was presumed.

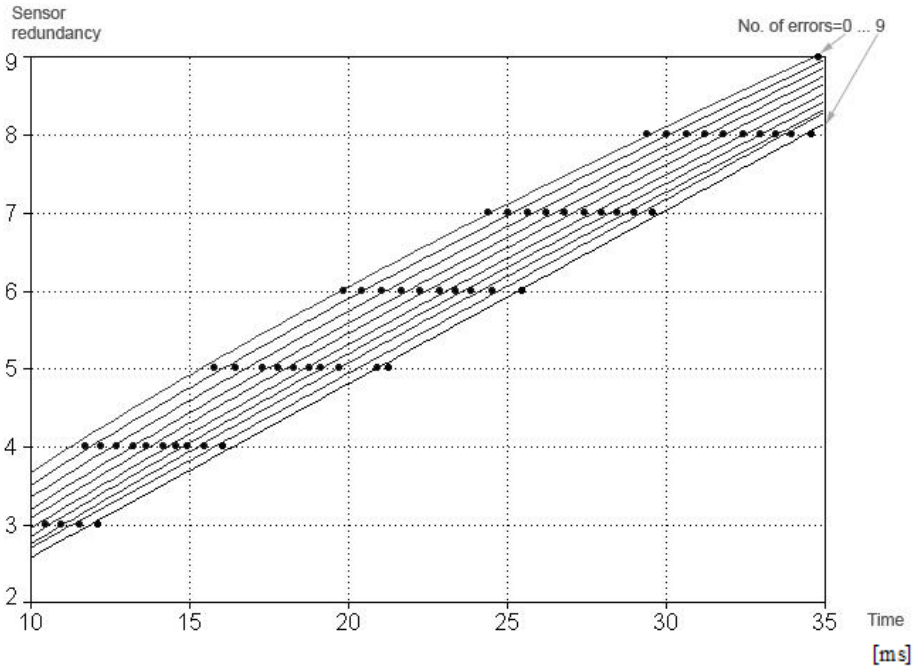

Fig.4.8. Fixed number of controllers (9), variable number of errors **[65]**

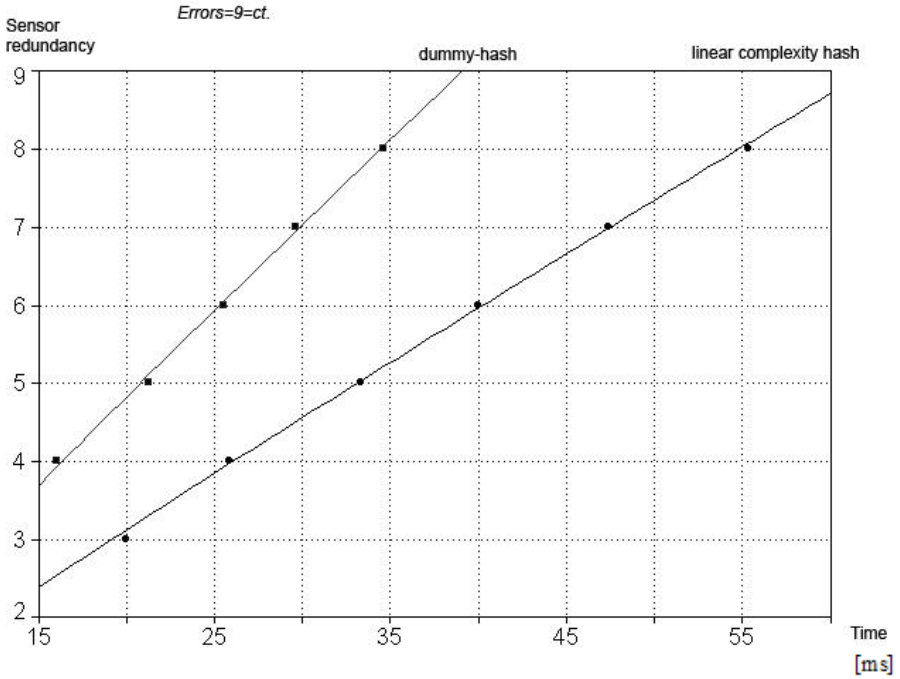

Fig.4.9. Fixed number of controllers (9), number of errors=9 **[65]**

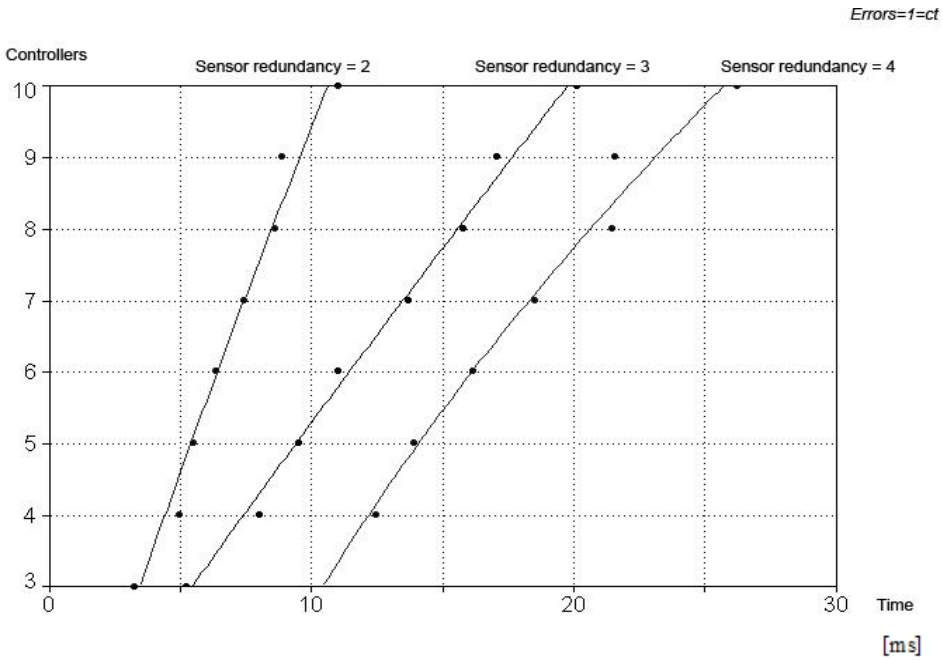

Fig.4.10. Fixed number of errors (1) **[65]**

In Fig.4.8 there can be noticed the variation of time with respect to sensor redundancy. The function obtained by this variation is

$$lny = a + blnx \tag{4.1}$$

In Fig.4.9 there is a comparison between the run up time of dummy hash function and the run up time of linear complexity hash, when there are a fixed number of errors. The general type of function describing this variation does not change. Only the value of *a* changes while *b* remains constant. In the last figure we see the variation of time when sensor redundancy and controller number vary and injected 1 error. The function best describing this is the power function.

4.5. Fault tolerance by means of error correction

As previously presented, a solution for addressing the threats to dependability and security of the proposed intelligent grid, would be a special agent called broker. Based on the proposed architecture of intelligent grid [88], a broker is a distinctive agent which role is to guard the data traffic on the network. The way to create a broker [65] is through a given protocol between the owner and consumer based on a list of brokers received from the directory facilitator. First of all, the role of a broker can be to detect a certain error in a message and to correct it. Another technique of attaining fault tolerance is that the role of broker is to detect the attack's pattern within a message, based on a targeted detection scheme. In other words, a certain attack's signature would be obtained by means of a dedicated scheme, as will be presented in the next section. These two different techniques will be studied on the following paragraphs as methods for attaining security and dependability in intelligent grid, the foundation of these method being the Generalized Linear Feedback Shift Register.

4.5.1. Solution based on GLFSR

The Generalized Linear Feedback Shift Register is a pattern generator with $n = (\delta \times m)$ outputs, being presented for the first time in [21]. Such a structure is designed over $GF(2^{\delta})$ and all its elements are part of $GF(2^{\delta})$. Even if the necessary components used for building such a GLFSR are adders, multipliers and storage elements, these components are not regular Galois field multipliers. More than this, it should be noticed that these components multiply the δ bit feedback input with a constant Φ_i. In Fig.4.11 is presented the structure of a Generalized LFSR. As a particular component, the adder is a set of δ XOR gates, in the same time the multiplier uses also XOR gates. The general form of the Generalized LFSR can be represented as

$$\Phi(x) = x^m + \Phi_{m-1}x_{m-1}+...+\Phi_1 x+\Phi_0 \tag{4.2}$$

What is interesting to notice in this case is that the Generalized LFSR has a total of m stages, namely D_0, D_1,..., D_{m-1}, with δ storage cells. At every shift, δ bits are being shifted from one stage to the next, while the feedback from the last stage, D_{m-1}, is sent to all the stages. The coefficients of the polynomial from (4.2) define the feedback connections specific to a GLFSR and are elements over $GF(2^{\delta})$. The

multiplied feedback input realized by an element Φ_k can be realized using XOR gates.

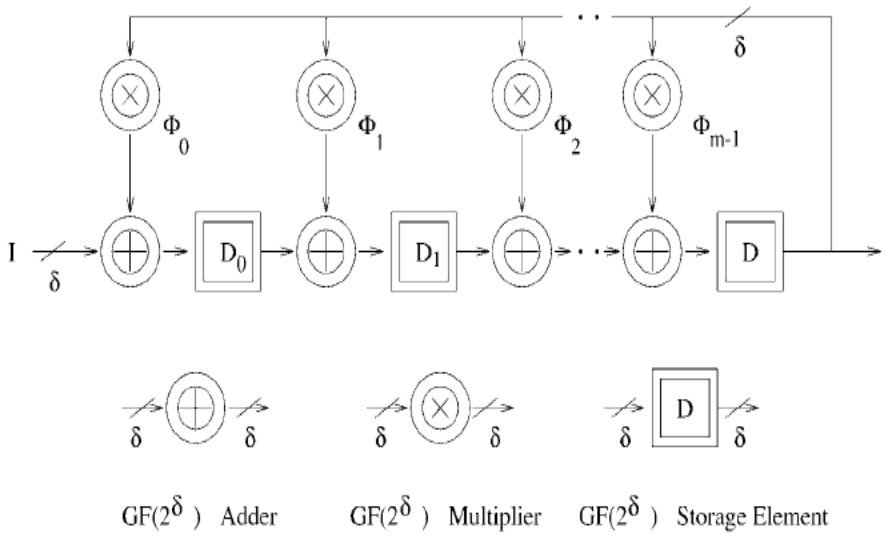

Fig.4.11. Structure of a GLFSR [54]

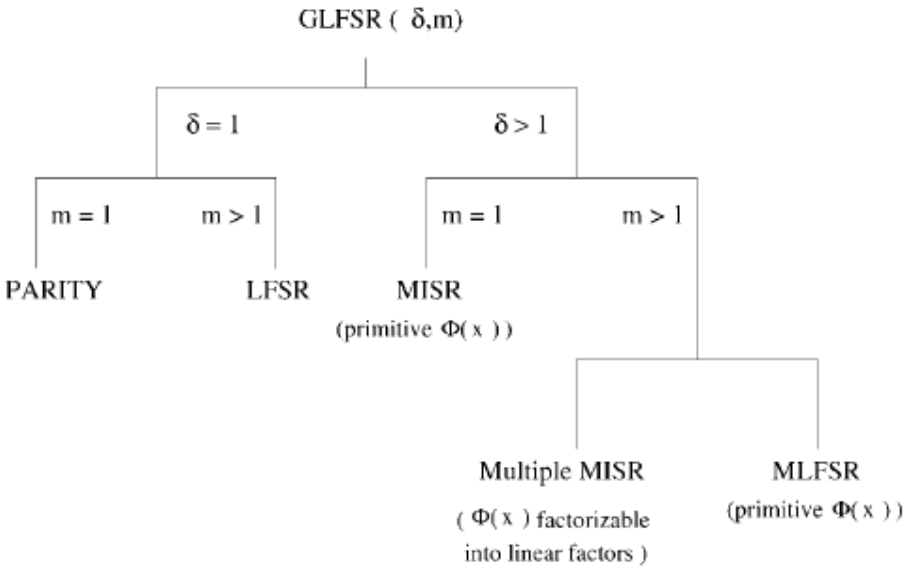

Fig.4.12. Different classes of GLFSR [54]

It has been demonstrated in [21] that a GLFSR with a primitive feedback polynomial (termed MLFSR), with $\delta > 1$ and $m > 1$, represents a structure which is a very effective pseudo-random pattern generator. GLFSR is a general structure [54] and all the other known structures like LIFSR, MISR (Fig.4.12) etc., are being special cases of GLFSR. When used to generate patterns, for a structure with n inputs, there can be m stages (like the one in relation (1)). The condition is that

each element has to belong to $GF(2^{\delta})$, where (m x δ) is at least or equal to n. In order to use a GLFSR as a pattern generator, it is required that a non-zero seed be loaded into the GLFSR and than clocked to produce the patterns. For the sake of simplicity, the GLFSR used in this paper has δ = 1, therefore is a LFSR.

4.5.2. Error correction in intelligent grid

In the same trend of design for reliability, the security of the architecture can be further improved by detecting and correcting the errors introduced by a possible attack via a transmission medium. The present solution can be used as an alternative, for example, to the case when the method presented in paragraph 4.4 and [65] is defeated. In this way, that after realizing that an attack has been produced on a certain exchanged message U(x), it is necessary to apply a correction to the compromised sequence. Such a correction can be realized by a method based on GLFSR. Without losing from generality, we can consider that the GLFSR has δ = 1, therefore is a LFSR. In order to adopt one of these methods, further comparisons are needed, presented in the last section of this chapter.

This method is taking into discussion a simplified model of a coded system as presented in Fig.4.13.

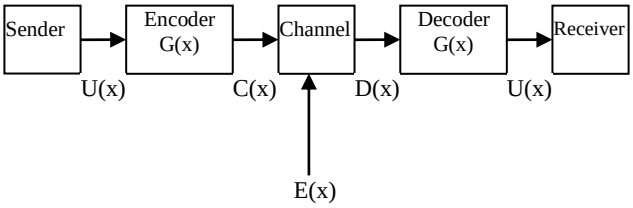

Fig.4.13. Simplified model for a coded system **[62]**

In terms of the intelligent grid architecture, the problem can be stated as following: if a certain agent called *Sender* intends to send a message, U(x) to another agent called *Receiver*, the message can be compromised by an attack. The idea is to find a method such as the compromised message to be corrected on-the-fly before arriving to the *Receiver* agent. A solution to this problem can be presented as follows:

1. Encode U(x) with a feedback polynomial G(x) [2], based on a GLFSR with δ = 1. The result is called C(x).
2. C(x) is sent via a transmission medium to the receiver.
3. C(x) is affected by an attack E(x) such as the result D(x) contains errors.
4. Build a Hamming matrix H_B starting from the equations resulted from the encoded LFSR scheme. Detect and correct the errors from D(x) using a feedback polynomial G(x) [78] and the appropriate column from H_B.
5. Send U(x) to the Receiver agent.

The problem can also be viewed from the owner-consumer paradigm [65]. In this case the broker agent can be responsible for encoding U(x), detecting and correcting the errors from the affected message. In order to encode U(x) with a feedback polynomial G(x), there can be used a dividing procedure, as presented in

$$C(x) = x^k U(x) + R(x) \quad (4.3),$$

where the maximum grade of U(x) is t-1, the grade of G(x) is k and the grade of R(x) is k-1. If we take into account that the target error to be detected and corrected is the singular error, it has been previously demonstrated [2, 78] that the most powerful feedback polynomial used for single error correction, double error detection is an indiscrete feedback polynomial.

The correcting and decoding part is established by building a Hamming matrix H_B from the equations of the LFSR used for decoding. It should be mentioned that both on the transmission and reception, the LFSR is constructed based on feedback polynomial G(x). From the Hamming matrix, the last column is used to build the logical part for the correction of the altered bit. Following next, a practical example will be presented in order to illustrate the above theoretical steps.

Example.

Let be U(x) the message to be sent between the agents, $U(x) = x^{10} + x^8 + x^7 + x^6 + x^4 + x + 1$.

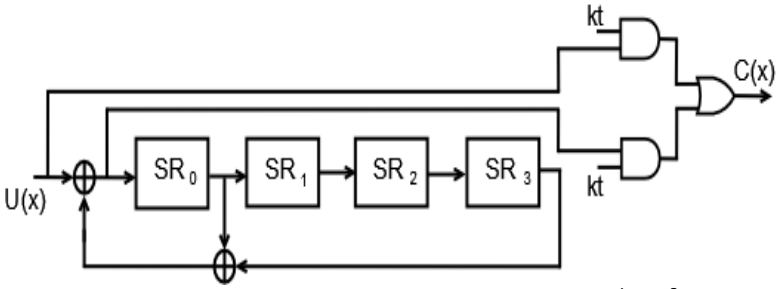

Fig.4.14. LFSR scheme for feedback polynomial G(x) = $x^4 + x^3 + 1$ **[62]**

The feedback polynomial used for encoding, decoding and correction is G(x) = $x^4 + x^3 + 1$.

Table 4.2. Encoding Sequence **[62]**

U(x)	**SR₀**	**SR₁**	**SR₂**	**SR₃**	**C(x)**
	0	0	0	0	
1	1	0	0	0	1
0	1	1	0	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	0	1	1
0	0	1	1	0	0
1	1	0	1	1	1
0	0	1	0	1	0
0	1	0	1	0	0
1	0	1	0	1	1
1	0	0	1	0	1
0	0	0	0	1	0
0	1	0	0	0	1
0	1	1	0	0	0
0	1	1	1	0	0

The LFSR-based encoding scheme used for transmission is presented in Fig.4.14. Based on this scheme is constructed the sequence for encoding the message to be sent, U(x).

Table 4.3. Correction of the affected bit **[62]**

D(x)	SR_0	SR_1	SR_2	SR_3	C(x)
	0	0	0	0	
1	1	0	0	0	1
0	1	1	0	0	0
1	0	1	1	0	1
1	1	0	1	1	1
1	1	1	0	1	1
0	0	1	1	0	0
1	1	0	1	1	1
0	0	1	0	1	0
0	1	0	1	0	0
1	0	1	0	1	1
1	0	0	1	0	1
0	**1**	1	0	0	1	0
1	1	1	0	0	1
0	1	1	1	0	0
0	1	1	1	1	0
0	0	1	1	1	1
0	1	0	1	1	0
0	0	1	0	1	1
0	1	0	1	0	1
0	1	1	0	1	1
0	0	1	1	0	0
0	0	0	1	1	1
0	1	0	0	1	0
0	0	1	0	0	0
0	0	0	1	0	1
0	0	0	0	1	1
0	1	0	0	0	**0**
0	1	1	0	0	1
0	1	1	1	0	0

The sequence of encoding U(x) is presented in Table 4.2. In Table 4.3 is presented the sequence used in order to decode D(x), which is affected by an error.

$$RD_0(\tau+1) = RD_0(\tau) \oplus RD_3(\tau) \oplus U(x) \tag{4.4}$$

$$H_B = \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \end{bmatrix} \begin{matrix} \text{SR0} \\ \text{SR1} \\ \text{SR2} \\ \text{SR3} \end{matrix} \quad (4.5)$$

The correction has been performed building a Hamming matrix. This matrix is constructed starting from relation (4.4). The last column from the Hamming matrix offers the combination for correcting the affected bit (relation (4.5)).

4.5.3. Experimental results

The proposed middleware [90] for intelligent grid is JADE, the present experimental studies taking into account the proposed algorithm with respect to the redundancy of the sensors.

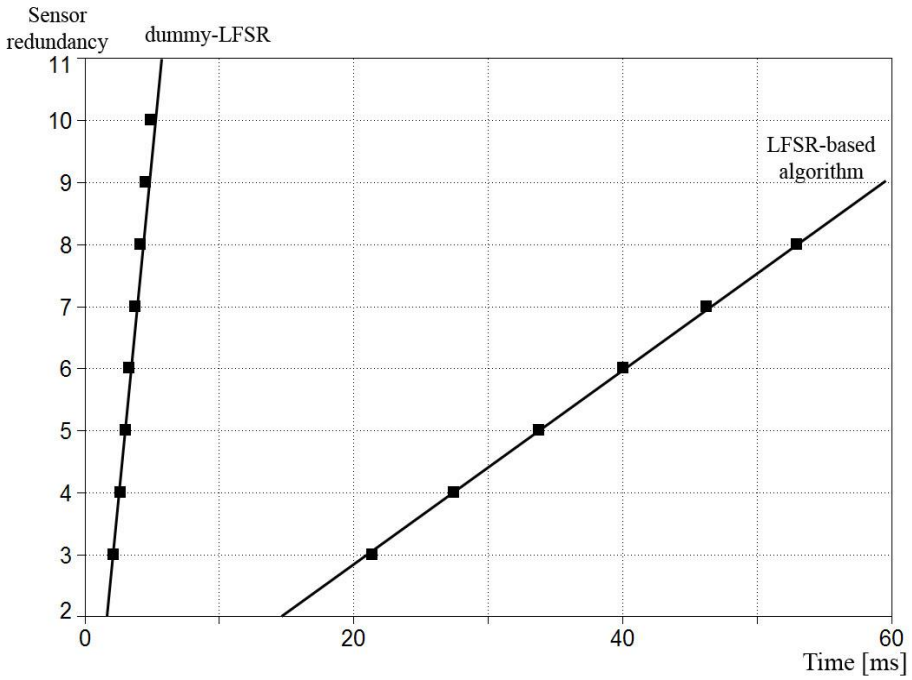

Fig.4.15. Fixed number of controllers **[62]**

A first simulation round is depicted in Fig.4.15 in order to establish the time overhead introduced by this correction method, first of all, we implemented a dummy-LFSR algorithm that only returns a value. Secondly, it was implemented the LFSR algorithm for encoding, decoding and correction of errors in order to establish the impact of this algorithm on the overall time. There are a fixed number of errors and no message loss was presumed. The comparable values of these two algorithms are presented in Fig.4.15. In Fig.4.16, the number of errors read from the architecture's sensors is increased from 0 to 4 errors. The function best describing this variation is

$$y = \mathrm{a} + bx + \mathrm{c}x\ln x + \mathrm{d}x^2 + \mathrm{e}^{\frac{\mathrm{x}}{\ln \mathrm{x}}} \tag{4.5}$$

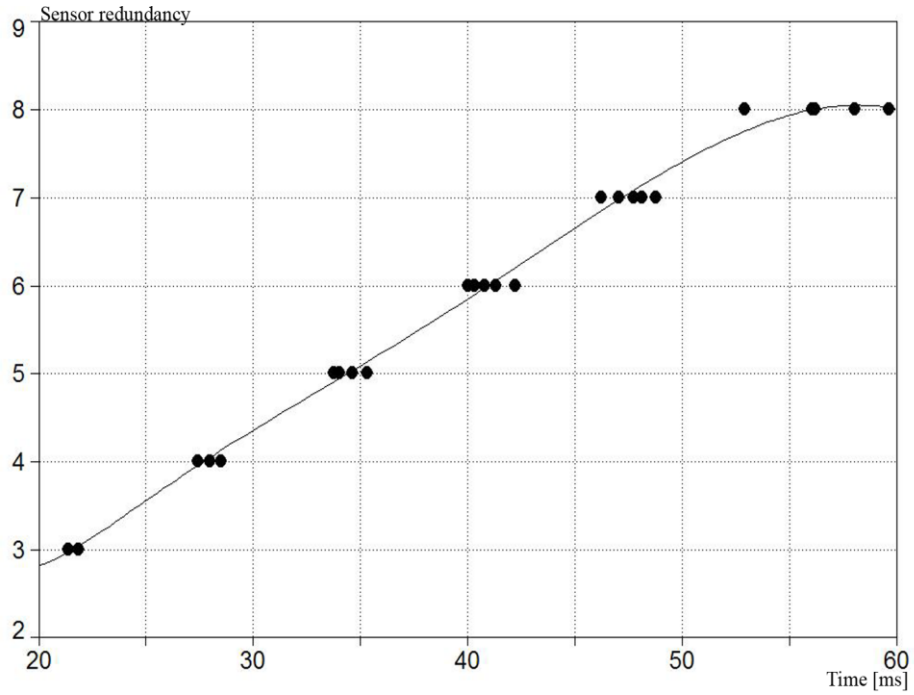

Fig.4.16. Variable number of errors, fixed number of controllers (9) **[62]**

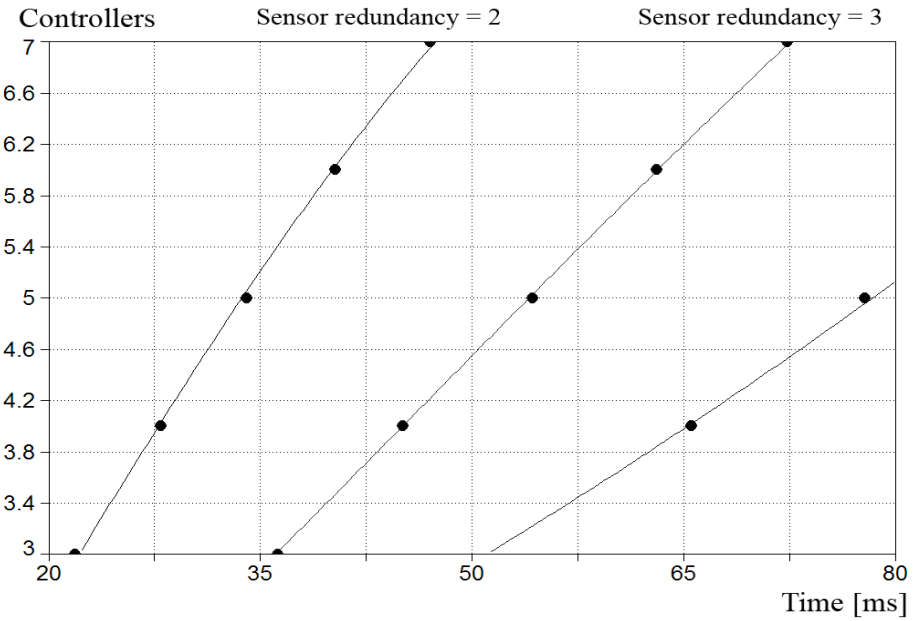

Fig.4.17. Variable number of controllers, fixed number of errors (1) **[62]**

In order to adopt the most appropriate function we had to consider and evaluate the maximum r^2 measure of diversification. This measure ranged from 0.9949 to 0.990 for a number of errors belonging to a spectrum of 0 to 4 errors. In Fig.4.16 is depicted the situation when the parameters of sensor redundancy are varied and the number of controllers is increased. The function obtained by this variation is

$$y^{-1} = a + b\frac{\ln x}{x^2} \tag{4.6}$$

In relation (4.6) it should be noticed that *a* remains constant, while the allure of the function is influenced by *b*'s coefficient.

4.6. Intrusion detection in intelligent grid

4.6.1. Necessity of intrusion detection

This subchapter addresses the problem of detecting different types of attacks in the context of the architecture based on intelligent agents and called intelligent grid, as a response to the drawbacks of the classical model of ambient intelligence.

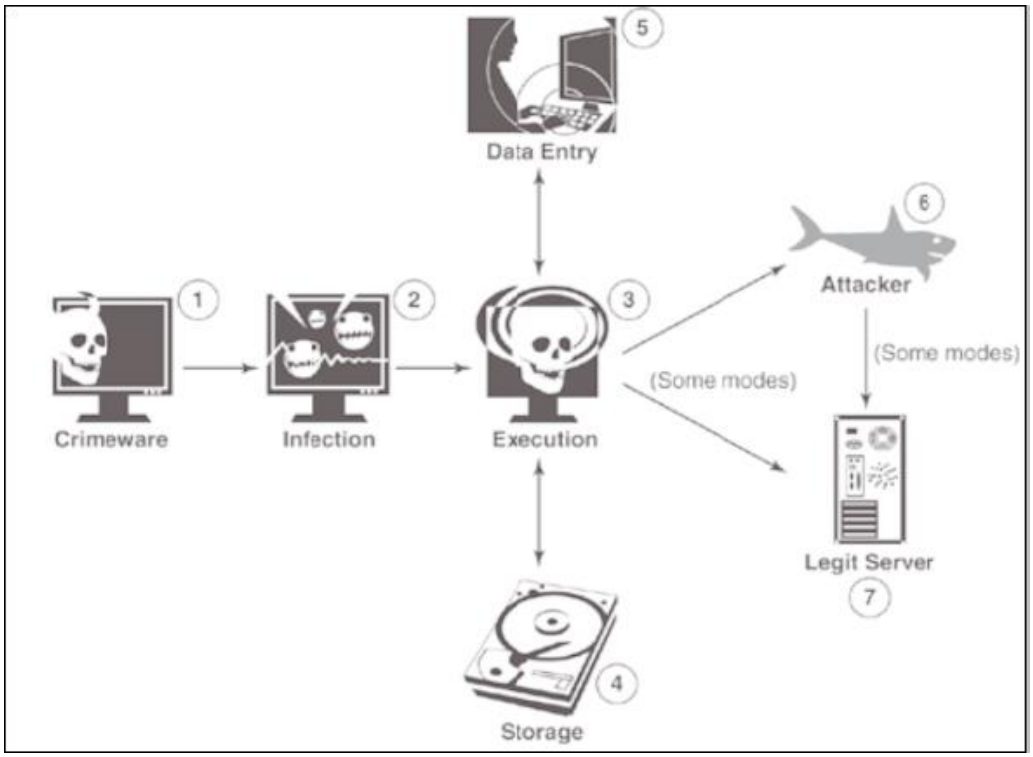

Fig.4.18. The stages of a typical crimeware attack [51]. (1) the crimeware is distributed, (2) breaking into a precise computing platform, and (3) executes. From this point the malware can adopt different behaviours depending on the nature of the specific crimeware instance. For example, the crimeware instance may (4) scan the user's hard drive for sensitive information or (5) intercept the user's keystrokes. In other behaviors, the crimeware instance transmits the information it collected (6) directly to the attacker or the information can be transmitted indirectly to the attacker through an otherwise (7) legitimate server that is being misused. There are attacks in which the information will be sent to (6) the attacker before it is passed on to (7) a legitimate server.

It is argued [51] that the days in which the authors of malicious code were interested in notoriety are gone, the focus of the attackers being shifted to the online threat landscape. Therefore a system networked to the online stage is a potential system for being hijacked. The motive of such a strategy is that they could potentially make serious money from the online activities, giving the fact that more and more people are conducting online transactions. A classical malicious code is now called criminal [51]. This trend has given rise to a new form of malicious software—namely, *crimeware*. According to the twelfth edition of the Symantec Internet Security Threat Report (ISTR), 46% of malicious code that propagated during the first half of 2007 did so over the standard protocol for mail transmission over the Internet (SMTP) [83]. The propagation of such a crimeware script and the specific behaviors applied to a certain system are presented in Fig.4.18.

The process of communication between agents is based on messages. Whenever there is a message exchange between agents, a malicious attack can occur and the message can be corrupted. The intelligent grid architecture, by its nature is prone to such vulnerabilities: different networks are being mixed; the networks are coupled to the internet and so on. State-of-the-art solution are based on intelligent agents [48, 90] because in this way the security concerns can be shifted towards networks itself rather than being host based. The security mechanisms can evolve into network-based and distributed approaches in order to deal to maintain a scalable solution and in the same time a heterogeneous open platform [48].

The idea presented in this subchapter is to identify peculiar attacks, each attack being identified by certain signature. The method implied for this task offers a tailored detection scheme for each particular attack.

4.6.2. Construction of detection schemes

The methods discussed for the process of detecting the attacks is based on the synthesis of a combinational block which transforms pseudo-random code words into deterministic test patterns, identified as the attack’s signature. In order to efficiently test for these test patterns, in other words, for signatures, incorporating Built-in-Self-Test (BIST) methods becomes inevitable. There have been different approaches [60] so as to find some trade-offs between the criteria for evaluating such BIST-based methods. Such criteria are the *fault coverage*, *test time* and the *area overhead*. It should be mentioned that the last criteria becomes in the case of intelligent grid the *complexity* of the method. High fault coverage means a long test time or a high complexity. The pseudo-random testing has been demonstrated [54, 60] to be the most efficient way of establishing the simplest trade-off between these criteria.

In [54] is presented a method (for the sake of simplicity, let’s call it Chatterjee-Pradhan method) of developing pattern generators which can detect

hard-to-detect faults, based on a relatively modest number of patterns. This method was developed in order to test specific faults from a circuit under test.

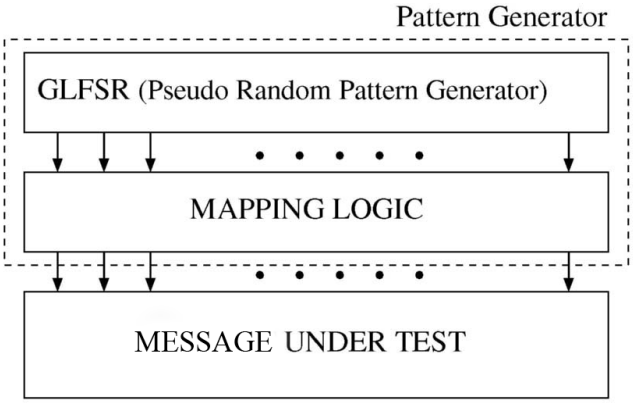

Fig.4.19. The process of detecting an attack's signature **[61]**

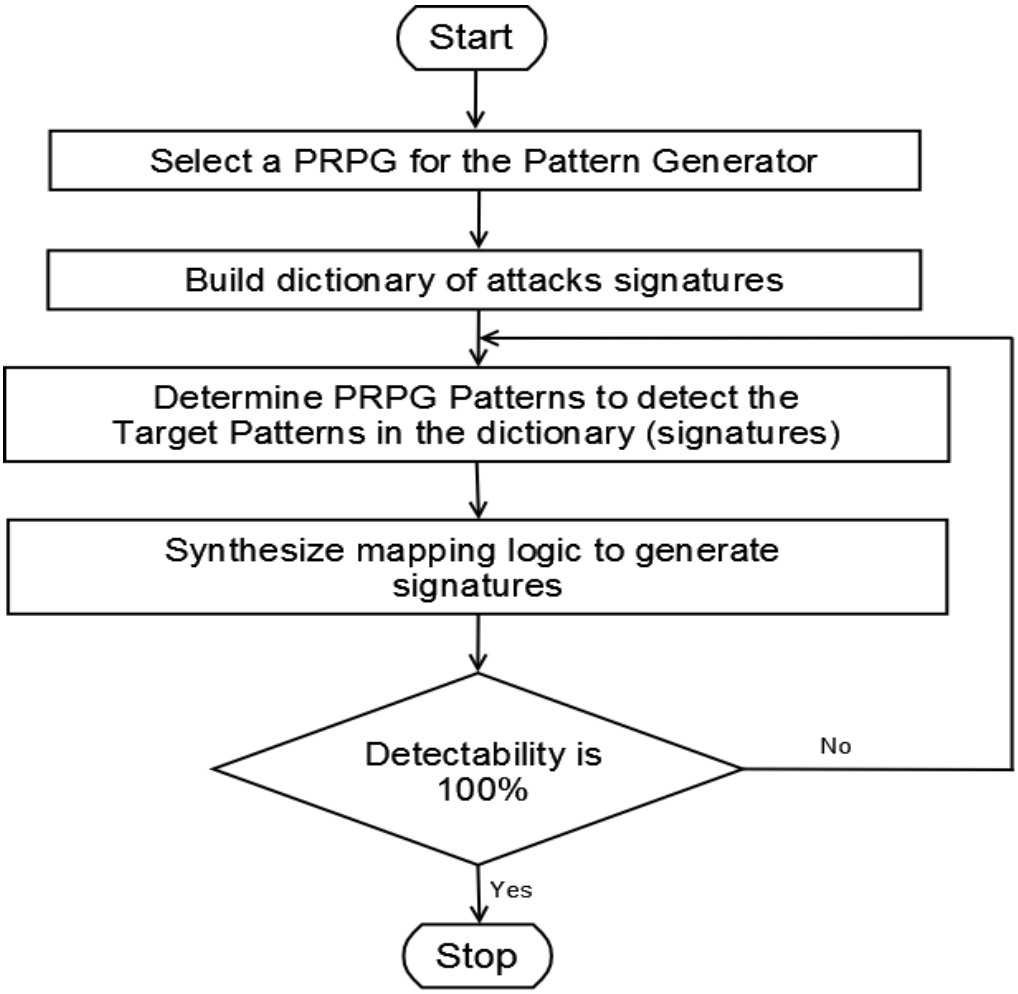

Fig.4.20. Design flow for the attacks detection **[61]**

Table 4.4. Dictionary of attacks signatures

Attack Type	Signature
Interception	FD01
	FC01
Router Attack-Link State	C0AF
Node Hijacking	2E2E
	2F2E

Table 4.5. Output transformations for interception, node hijacking, link state attacks, $G_1(x)$

Hex value	Input Vectors																Output Values(Target Patterns)															Hex value
	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	
FB21	1	1	1	1	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1	FD01
F531	1	1	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	FC01
C0EF	1	1	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	0	0	0	0	0	0	1	0	1	0	1	1	1	1	C0AF
02F7	0	0	0	0	0	0	1	0	1	1	1	1	0	1	1	1	0	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	2E2E
2F70	0	0	1	0	1	1	1	1	0	1	1	1	0	0	0	0	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	0	2F2E

Table 4.6. Output transformations for interception, node hijacking, link state attacks, $G_2(x)$

Hex value	Input Vectors																Output Values(Target Patterns)																Hex value
	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	
FDE2	1	1	1	1	1	1	0	1	1	1	1	0	0	0	1	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1	FD01
3C01	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	FC01
C01A	1	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	1	0	0	0	0	0	0	1	0	1	0	1	1	1	1	C0AF
6F01	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	0	2F2E
DE02	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	2E2E

Table 4.7. Output transformations for interception, node hijacking, link state attacks, $G_3(x)$

Hex value	Input Vectors																Output Values(Target Patterns)																Hex value
	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	
FD46	1	1	1	1	1	1	0	1	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1	FD01
1C4C	0	0	0	1	1	1	0	0	0	1	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	FC01
C0C5	1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	0	1	0	1	1	1	1	C0AF
2F5B	0	0	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	0	2F2E
5EB6	0	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	2E2E

Starting from the Chatterjee-Pradhan approach, a specific method for intelligent grid can be disclosed [61], as depicted in Fig.4.19, but in the case of intelligent grid architecture, there is a necessity to verify that each message exchanged in the communication process between the nodes is not malicious in some way. To be more specific, it is required to identify if a message is infected by a certain attack. This can be achieved by a detection process of the attack's signature.

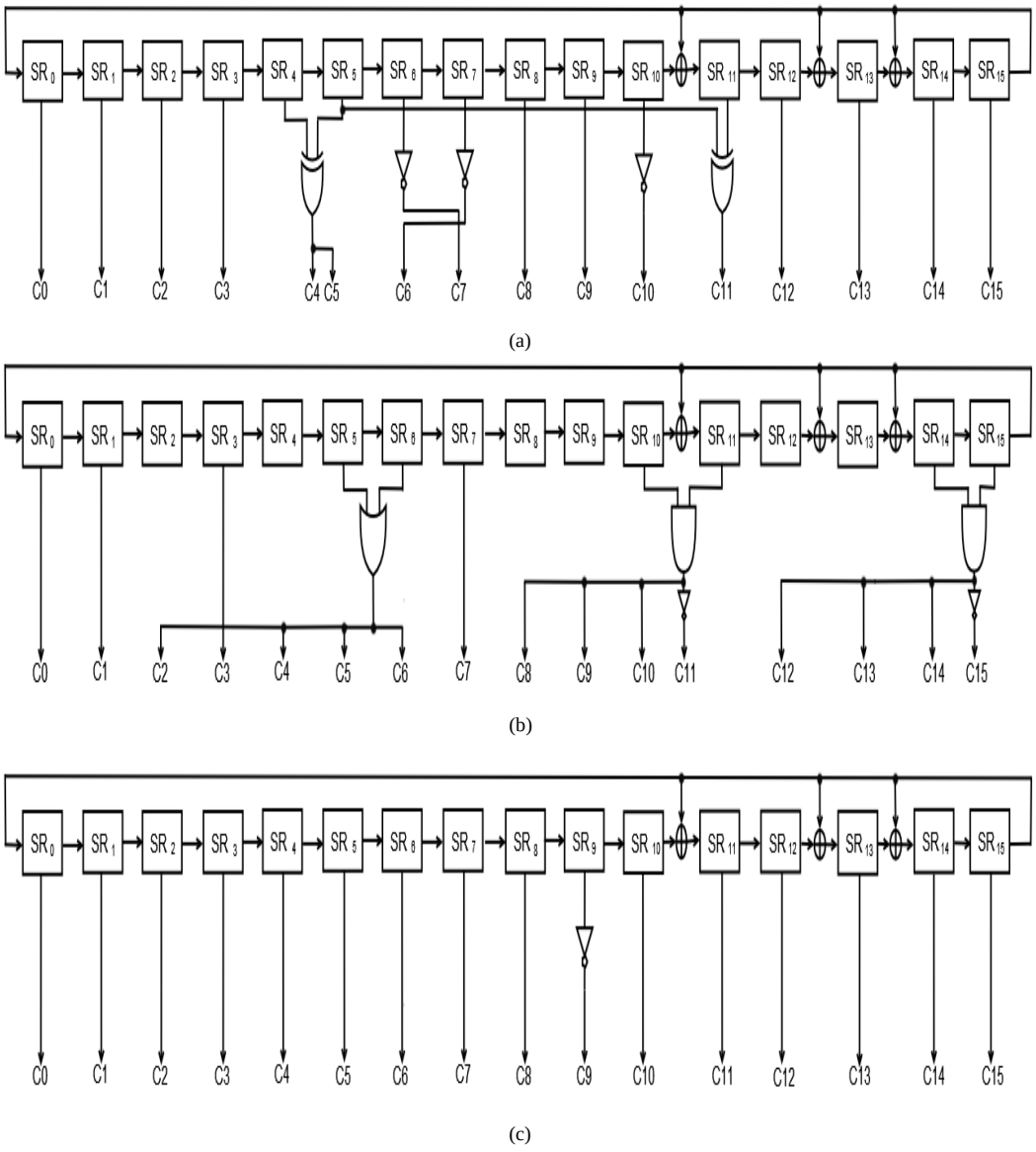

Fig.4.21. Detection scheme for (a) Interception attack; (b) Node Hijacking attack; (c) Link State attack, $G_1(x) = x^{16} + x^{14} + x^{13} + x^{11} + 1$

In Fig.4.19 it is presented how the outputs of the GLFSR are being transformed by the mapping logic into test vectors (attack's signatures) for the message under test (MUT). The GLFSR is loaded initially with a seed from which the patterns for the mapping logic will be generated. In [54] it is stated that using GLFSRs as pseudorandom pattern generator is a way to provide better coverage of majority of faults. As stated before, in the case of intelligent grid, the faults are aimed to be taken from a digital signatures dictionary of certain attacks, as depicted in Table 4.4. This particular dictionary was build based on [86].

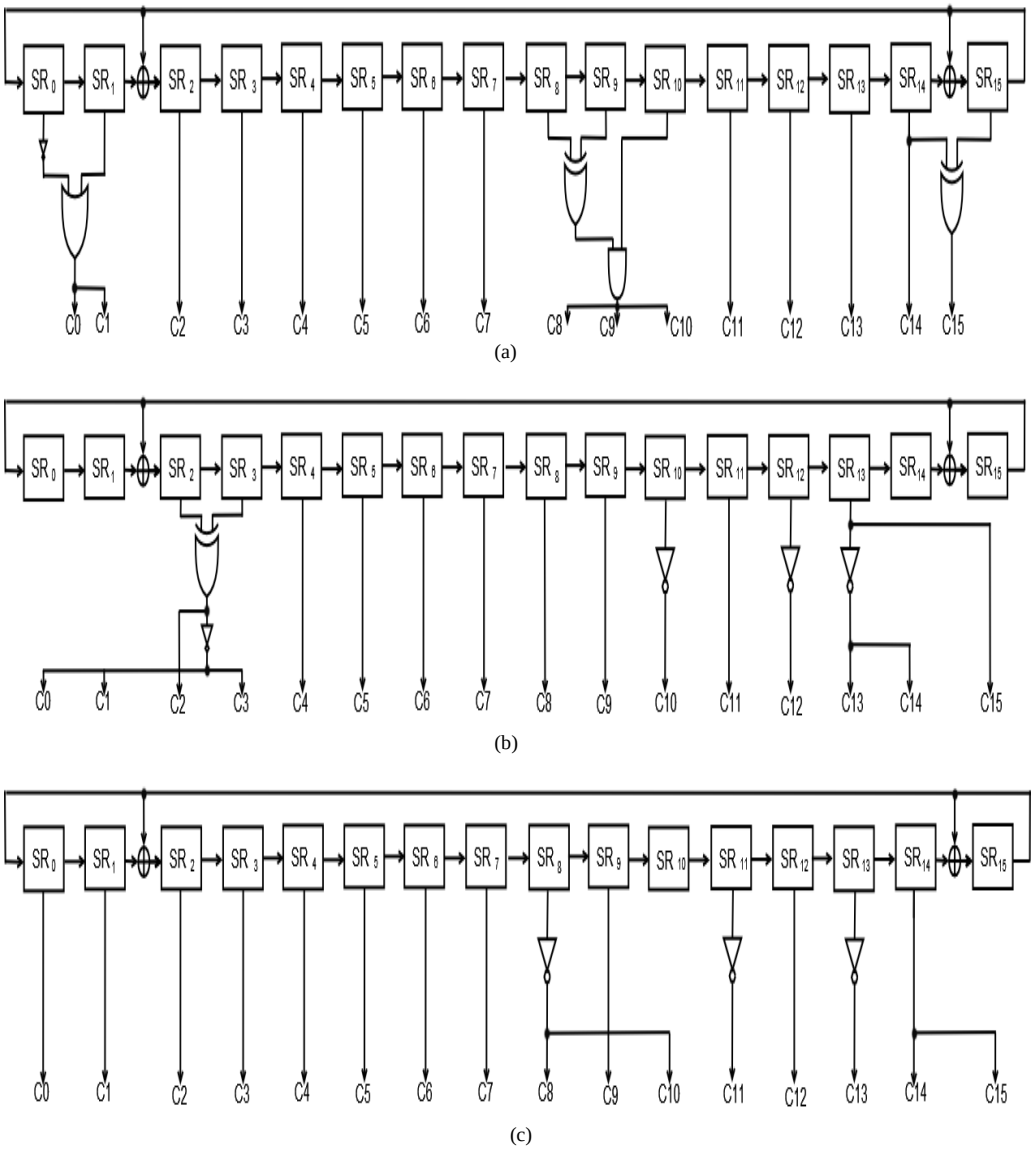

Fig.4.22. Detection scheme for (a) Interception attack; (b) Node Hijacking attack; (c) Link State attack, $G_2(x) = x^{16} + x^{15} + x^2 + 1$ **[61]**

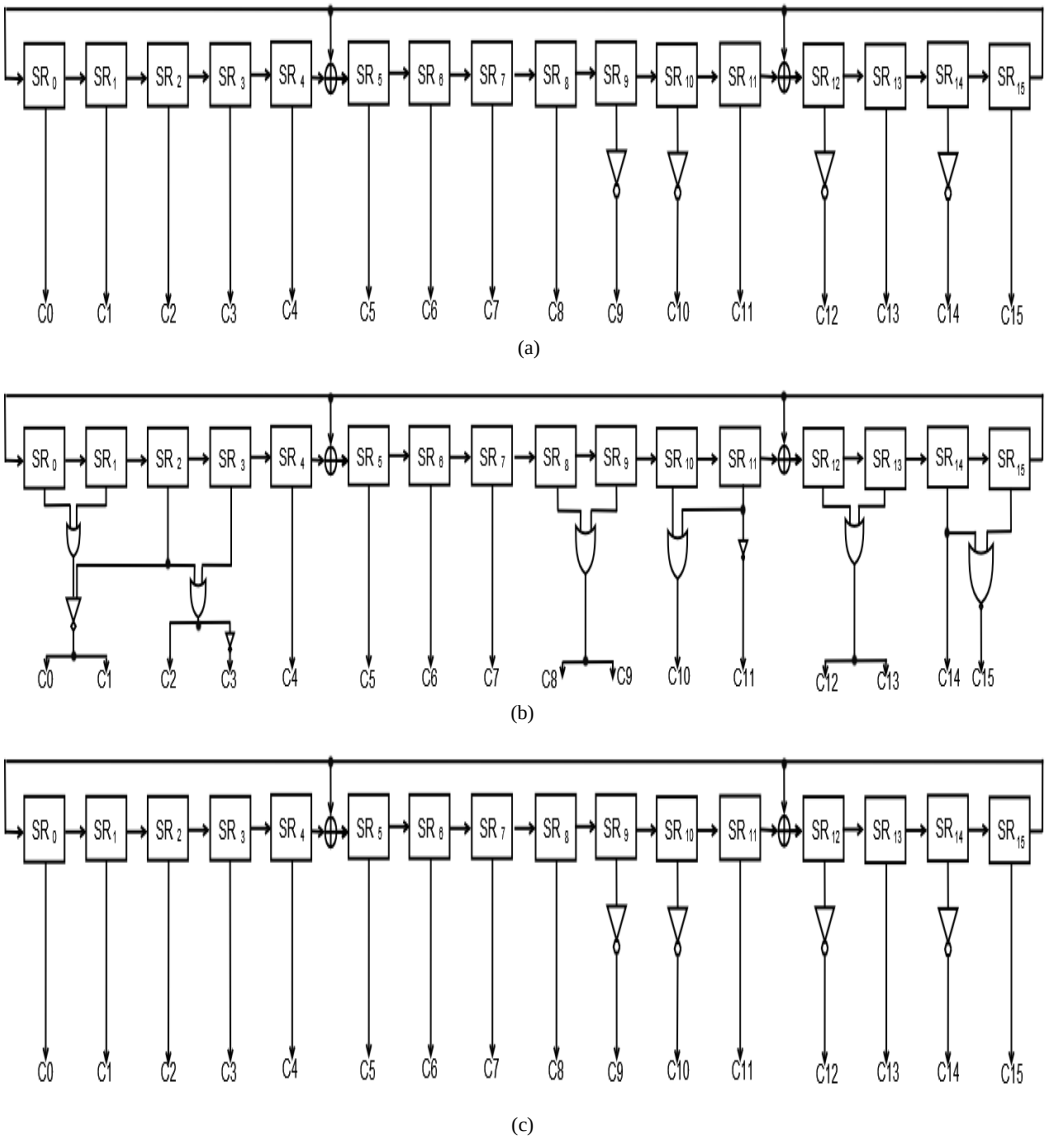

Fig.4.23. Detection scheme for (a) Interception attack; (b) Node Hijacking attack; (c) Link State attack, $G_3(x) = x^{16} + x^{12} + x^5 + 1$

Given a MUT, the Chatterjee-Pradhan method can be rephrased for detecting attacks in intelligent grid by the following design procedure [61]:

1. Select a GLFSR functioning as a PRPG engine for the composite pattern generator
2. Build a signature attack dictionary
3. Determine a minimal set of PRPG patterns which test the determined fault list (Target Patterns).

4. For the signatures of each attack, design an efficient combinational circuit with the PRPG so that the combined pattern generator generates all the signatures for a certain attack.

The Chaterjee-Pradhan method treats its similar step 3 of this method by dividing the process of determining the set of target patterns into two sub-methods. First of all, it generates input indexes for the unit under test and secondly will try to find a match between target patterns to PRPG patterns. The complexity of such algorithms is $O(k.n^3)$, respectively $O(|T|^2(n^3 + |P|.n))$ [54]. We propose to determine the appropriate combinations from step 3 by adopting the following algorithm. Each PRPG pattern will be divided in a number of *m* number of bits. This group will be searched for in the desired fault, namely the attack's signature. If the group is found in the signature, it will be provided at the output. Otherwise, for the unfounded groups, an appropriate way of minimizing will be searched for so as all the outputs to be completed. Such a method has a complexity of $O(N.\log N^{Nk})$.

```
findAMatch( Targets, PRPG vectors )
BEGIN
        Matching M = 0
        Divide the PRPG vector in m number of bits
        For PRPGvectors ( i ), i = 0..n
        For targetPatterns( j ), j = 0..n
                Search the pattern of length m in targetPatterns
                If found, update M;
        Choose the PRPG vectors based on the maximum number for
        founded patterns
END
```

The last step from the design procedure can be stated as the following problem: being specified an initial seed to the PRPG and a constrained test pattern length, design an efficient combinational logic area which aims at providing the expected target within the specified length. The result of the design procedure is the detection of a dedicated signature, namely a particular attack to the intelligent grid architecture. In order to complete with the last step, a process of minimization should be employed so as to obtain the outputs that offer the bits of the signature. If we take into consideration exemplifying this method as presented in Table 4.5, 4.6, 4.7 it can be observed that there is a row matching of the target patterns so as to obtain the desired attack's signature. Further on, another important observation is that this method, based on row matching, provides only one detection scheme for *each* signature. In other words, each signature from the dictionary of attacks is detected by a different detection scheme.

Considering a refinement of the proposed method, we should take into consideration obtaining a unique detection scheme for all the entries of the dictionary of attacks. We propose to address this goal by adopting a column approach for matching of the PRPG patterns into target patterns, namely faults. The minimization process will be completed in this case when those columns that cannot be matched will be minimized so as all the desired outputs to be completed. These particular outputs can be obtained from the column values of the input patterns. In Fig.4.24, Fig.4.25, Fig.4.26 the column-matching technique is detailed taking in consideration three different feedback polynomials. The resulted detection schemes, one for all the entries from the dictionary are presented in Fig.4.27, Fig.4.28, Fig.4.29.

	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	
FB21	1	1	1	1	1	0	1	1	0	0	1	0	0	0	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	0	1	FD01
F531	1	1	1	1	0	1	0	1	0	0	1	1	0	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	1	FC01
C0EF	1	1	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1	1	0	0	0	0	0	1	0	1	0	1	1	1	1	1	C0AF
02F7	0	0	0	0	0	0	1	0	1	1	1	1	0	1	1	1	0	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	2E2E
2F70	0	0	1	0	1	1	1	1	0	1	1	1	0	0	0	0	0	1	0	0	1	1	1	1	0	1	0	1	1	1	0	0	2F2E

$C_0 = X_0$
$C_1 = X_1$
$C_3 = C_{10} = C_{14} = X_3$
$C_2 = C_4 = C_5 = X_{12}'$
$C_6 = X_6$
$C_7 = X_4$
$C_8 = X_{12}$
$C_9 = C_{11} = X_{10}'$
$C_{12} = C_{13} = X_6 \oplus X_9 + X_6 X_{15}'$

Fig.4.24. Column matching method for $G_1(x)$

	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	
FDE2	1	1	1	1	1	1	0	1	1	1	1	0	0	0	1	0	1	1	1	1	1	1	0	1	0	0	0	0	1	1	0	1	FD01
3C01	0	0	1	1	1	1	0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0	1	1	FC01
C01A	1	1	0	0	0	0	0	0	0	0	0	1	1	0	1	0	1	0	0	0	0	0	0	0	1	0	1	0	1	1	1	1	C0AF
6F01	0	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	0	2F2E
DE02	1	1	0	1	1	1	1	0	0	0	0	0	0	0	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	2E2E

$C_0 = C_1 = C_{15} = X_6'$

$C_2 = C_8 = X_5$

$C_3 = X_2X_3$

$C_4 = C_5 = X_4$

$C_6 = X_6$

$C_7 = X_7$

$C_9 = C_{11} = X_{13}$

$C_{10} = C_{12} = C_{13} = C_{14} = (X_2X_3)'$

Fig.4.25. Column matching method for $G_2(x)$

Hex value	Input Vectors																Output Values(Target Patterns)															Hex value
	X_0	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}	X_{11}	X_{12}	X_{13}	X_{14}	X_{15}	C_0	C_1	C_2	C_3	C_4	C_5	C_6	C_7	C_8	C_9	C_{10}	C_{11}	C_{12}	C_{13}	C_{14}	C_{15}	
FD46	1	1	1	1	1	1	0	1	0	1	0	0	0	1	1	0	1	1	1	1	1	1	0	1	0	0	0	0	0	0	0	1	FD01
1C4C	0	0	0	1	1	1	0	0	0	1	0	0	1	1	0	0	1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	1	FC01
C0C5	1	1	0	0	0	0	0	0	1	1	0	0	0	1	0	1	1	1	0	0	0	0	0	0	1	0	1	0	1	1	1	1	C0AF
2F5B	0	0	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	0	1	0	1	1	1	1	0	0	1	0	1	1	1	0	2F2E
5EB6	0	1	0	1	1	1	1	0	1	0	1	1	0	1	1	0	0	0	1	0	1	1	1	0	0	0	1	0	1	1	1	0	2E2E

$C_0 = C_1 = C_{15} = X_6'$
$C_2 = X_4$
$C_3 = X_2X_3$
$C_4 = C_5 = X_0'$
$C_6 = X_{11}$
$C_7 = X_{13}'$
$C_8 = X_4'$
$C_9 = C_{11} = X_7X_{10}$
$C_{10} = C_{12} = C_{13} = C_{14} = (X_3X_9)'$

Fig.4.26. Column matching method for $G_3(x)$

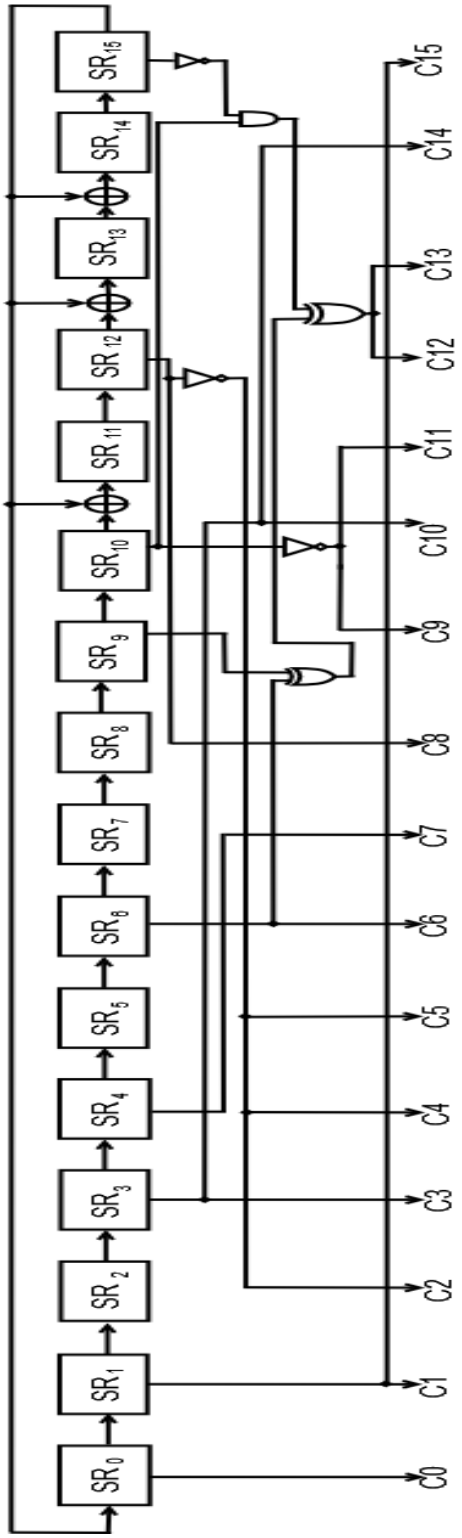

Fig.4.27. Detection scheme for the entire dictionary based on $G_1(x)$

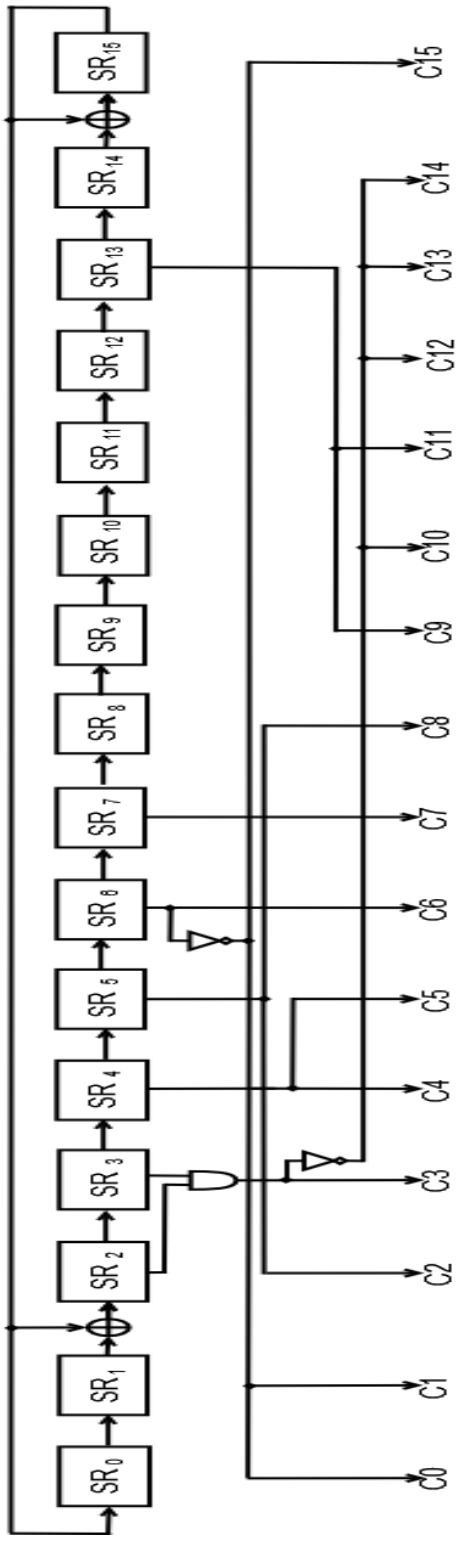

Fig.4.28. Detection scheme for the entire dictionary based on $G_2(x)$

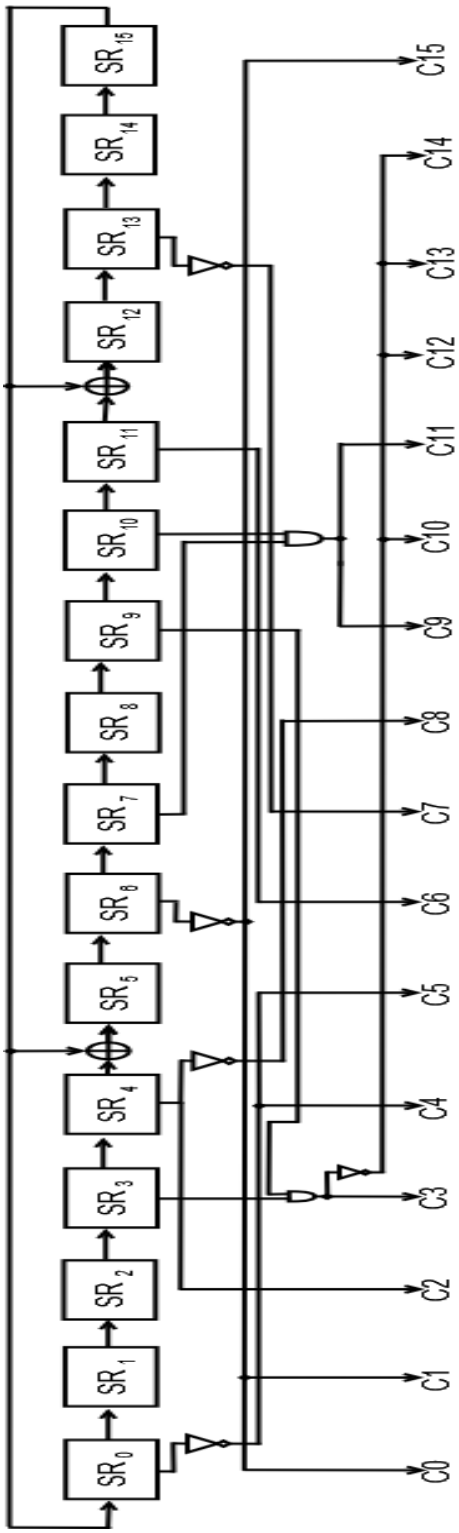

Fig.4.29. Detection scheme for the entire dictionary based on $G_3(x)$

The method can be therefore reformulated (Fig.4.30) so as to obtain one detection scheme for all the signatures of the dictionary. The improvement of the initial proposed method is taking into consideration the fact that a column matching technique is employed. The complete method would have the following steps:

1. Select a GLFSR functioning as a PRPG engine for the composite pattern generator
2. Build a signature attack dictionary
3. Determine a minimal set of PRPG patterns which test the determined fault list (Target Patterns).
4. Based on a column matching process obtain the Target Patterns.
5. For the signatures of each attack, design an efficient combinational circuit with the PRPG so that the combined pattern generator generates all the signatures for a certain attack.

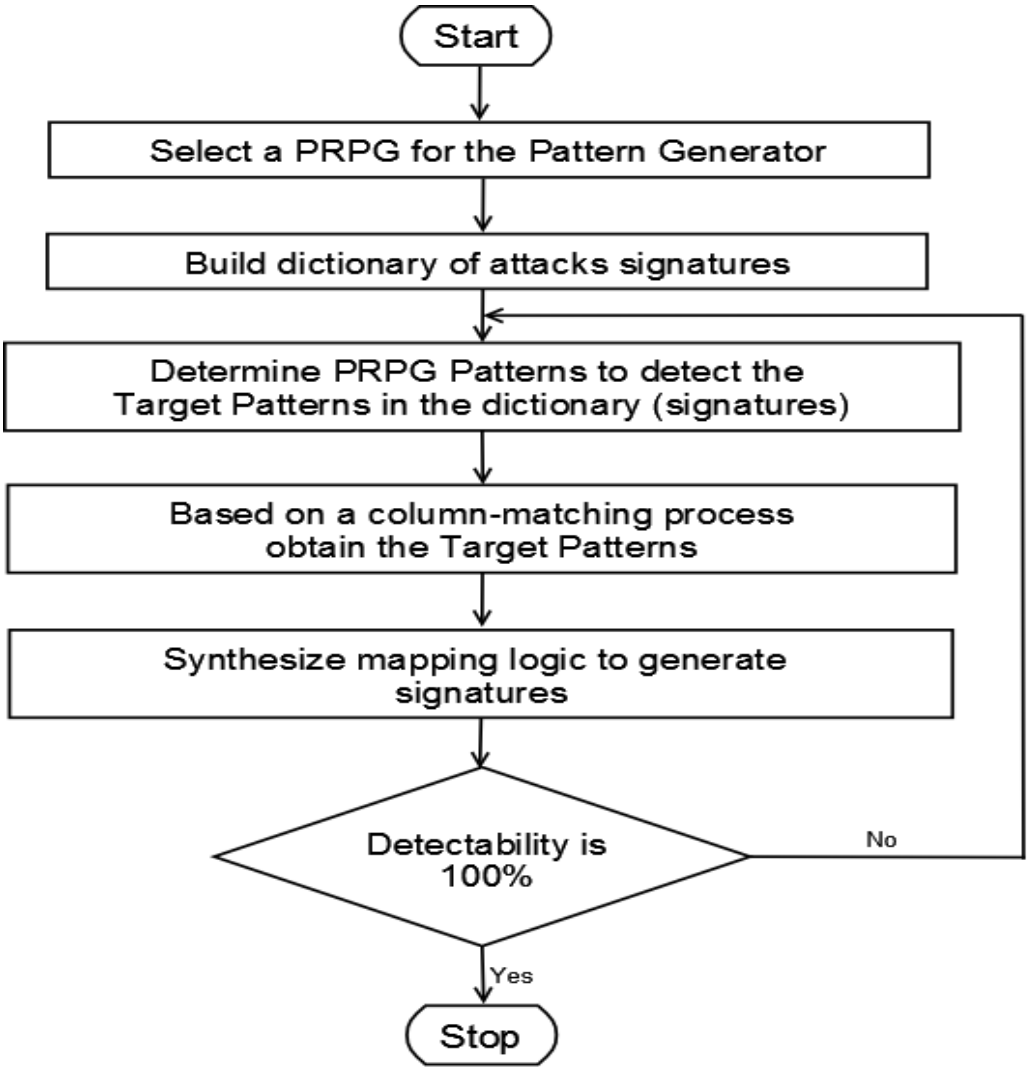

Fig.4.29. Design flow for the attacks detection invested with column matching technique

4.6.3. Experimental results

The idea that has been envisioned was to offer a targeted detection scheme for a certain attack. The first part of this process was to test which feedback polynomial would be most appropriate to be used in order to obtain the lowest test application time. In other words, time detection of the attacks to be the most convenient.

The method presented previously was tested to the broker node with different feedback polynomials. Without loosing from generality, the feedback polynomials used in order to generate the target patterns are $G_1(x) = x^{16} + x^{14} + x^{13} + x^{11} + 1$, $G_2(x) = x^{16} + x^{15} + x^2 + 1$ (Cyclic Redundancy Check 16), $G_3(x) = x^{16} + x^{12} + x^5 + 1$ (Cyclic Redundancy Check 16 CCITT). The last two feedback polynomials, namely CRC16 and CRC16CCITT are among the most used and standardized polynomials due to their capacity of error correction and pseudo random pattern generators [109]. Each of these feedback polynomials was used as a PRPG engine for obtaining all the combinations used as inputs for the mapping logic. In order to facilitate the process of obtaining all the combinations, a testbench was used, namely LFSRTestbench [47].

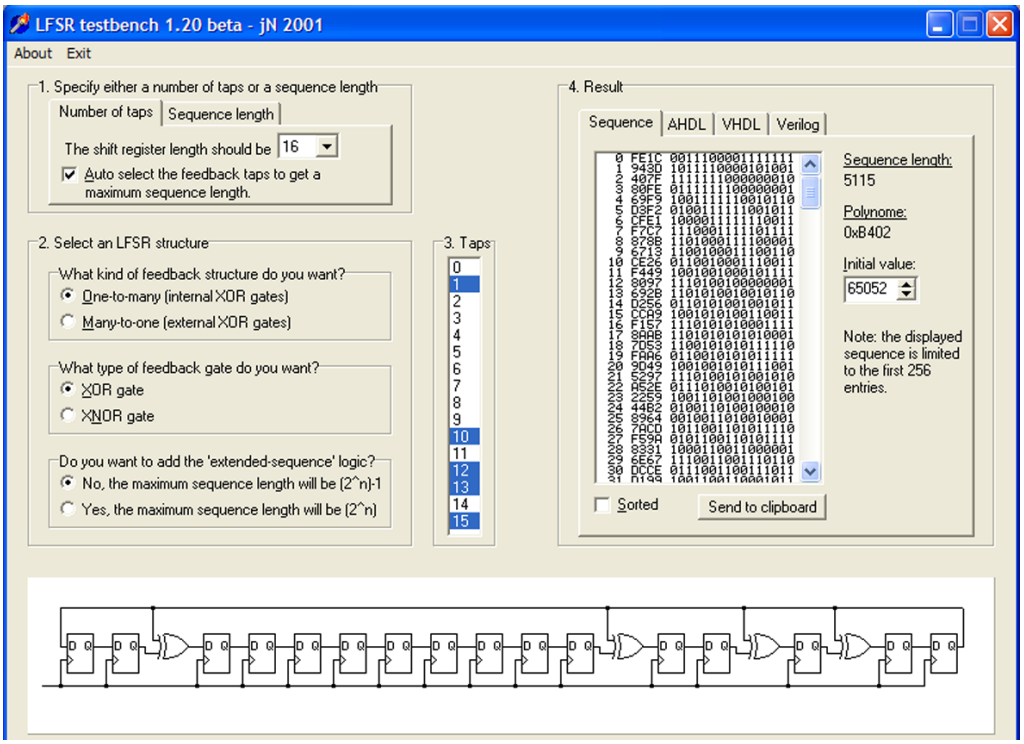

Fig.4.31. LFSRTestbench environment

The next step was to build a dedicated mapping logic which would provide the attack's signature. This partial goal was to be aimed for each feedback polynomial in order to establish which of these would be more appropriate to be used in order to obtain the dedicated detection scheme. First of all, it was used the construction method for the detection scheme presented in Fig.4.20. We used a row

matching technique in this case. In Table 4.5, 4.6, 4.7 are presented output transformations for each attack: interception, node hijacking, link state. The following step of the proposed method is presented in Fig.4.22, Fig.4.23, Fig.4.24. In other words, it can be followed how a certain target pattern will provide by means of mapping logic the desired output combination which is the attack's signature. It should be again underlined that for each attack's signature, there is a unique detection scheme. The improved method of constructing detection scheme is presented in Fig.4.30 and employs a column-matching approach. The matching step is detailed in Fig.4.24, 4.25, 4.26. What is particular for this method is that the obtain detection schemes based on the used PRPG can be used to detect all the signatures from the dictionary.

Table 4.8. Comparison between the results obtained with different feedback polynomials **[61]**

Tap	Initial seed	Number of iterations to obtain the combinations	Test Application Time [ms]	Fault coverage
$G_1(x)=x^{16}+x^{14}+x^{13}+x^{11}+1$	FE1C	65	8.671	3
$G_2(x)=x^{16}+x^{15}+x^2+1$	FFDC	15	2.297	3
$G_3(x)=x^{16}+x^{12}+x^5+1$	639C	50	7.063	3

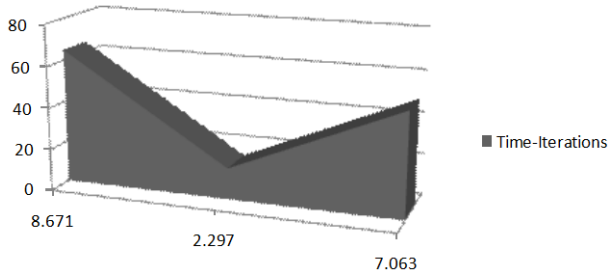

Fig.4.32. Variation of time with number of iterations **[61]**

The outputs of GLFSR are being transformed by the logic part into attack's signatures. For each attack type from Table 4.4 and every feedback polynomial, it was build a separate detection scheme. In other words, the broker was capable of detecting three different types of attacks on the intelligent grid network. The advantage of the dictionary attacks and dedicated scheme idea is that the detectability [54] rate of an attack being maximum. The detectability of a fault it is defined "*in terms of the fraction of all possible test vectors that can detect the particular fault*", in our case a particular attack's signature. Another particular problem that we faced was the establishment of specific metrics. In Table 4.8 the experimental results have been centralized according to peculiar metrics for this type of solution [5, 54], such as the number of iterations used to obtain the input combinations to the mapping logic, test application time and fault coverage. It can be seen that the number of iterations used in order to obtain all the combinations used for achieving the attack's signature, exponentially increases with the test application time, as it can be analyzed in Fig.4.32.

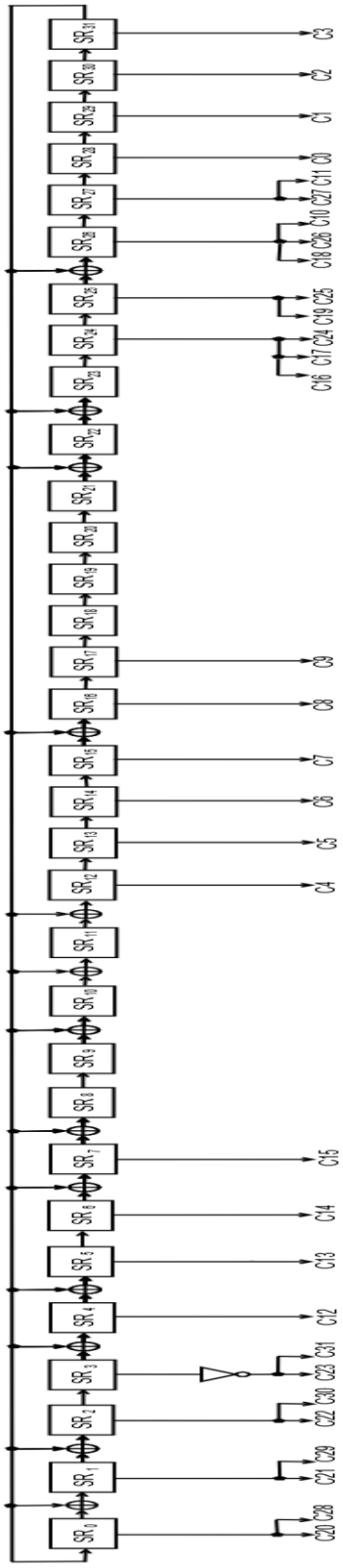

Fig.4.33. Detection scheme for 32-bit signature

Table 4.9. Output transformations for node hijacking,
$G_4(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ **[64]**

HEX Value	Input Vector								Output Values (Target Patterns)								HEX Value
FFDC0000	1111	1111	0101	0000	1001	0100	0010	1100	1100	0000	1010	1111	0010	1110	0010	1110	C0AF2E2E

Table 4.10. Result for 32-bit detection scheme **[64]**

Tap	Initial seed	Number of iterations to obtain the combinations	Test Application Time [ms]	Fault coverage
$G_4(x)$	FFDC0000	5	9.859	1

In Table 4.8 the *Initial seed* column has different values because otherwise, for an equal seed for all the feedback polynomials, the detectability would be practical 0% that cause being that no target pattern would be generated. The *Fault coverage* column has the same value for each feedback polynomial because the dictionary has three different attacks. The experimental results show that $G_2(x)$ is the most optimum feedback polynomial for generating dedicated schemes for attacks' detection, taken into consideration the values of number of iteration and the test application time.

4.6.4. The case of 32-bit detection scheme

A special case of detection scheme construction has been encountered for node hijacking attack because the signature of this type of attack is on 32 bits. As it can be observe in Table 4.9, the signature is C0AF2E2E. The methodology of constructing the detection scheme is the one presented in the previous sections. The employed feedback polynomial is $G_4(x) = x^{32} + x^{26} + x^{23} + x^{22} + x^{16} + x^{12} + x^{11} + x^{10} + x^8 + x^7 + x^5 + x^4 + x^2 + x + 1$ [7]. The resulted scheme is offered in Fig.4.33. In Table 4.10 it can be noted that the fault coverage has value 1 because in this case is only one attack that was targeted to identify. As an observation it should be said that the test application is rather high, but the number of iterations to obtain the necessary signature has a very significant value, compared to the number of iterations obtained in the case of 16-bit signatures.

4.7. Conclusions

The present chapter addresses the problem of security in a non-traditional parallel architecture called intelligent grid, by presenting a method for correcting the errors [62] introduced by possible attacks. A detailed example was presented in order to illustrate the modality in which this algorithm can be used in this kind of system. Previously, the fault tolerance issue was focused by the method of digital signatures based on hash function [65]. Such approach can be used in order to realize the authentication of the messages in the proposed architecture based on intelligent grid. From the experimental results we can see that the present algorithm increases the overhead time, but in the same time offers a fault-free technique in order to correct the errors that can appear when the messages are sent between the intelligent agents.

Following next we intended to build a dedicated signature dictionary which contains signatures of different attacks. Based on this dictionary, dedicated detection schemes will be constructed for different attacks and in different simulation conditions. The obtained results are specific detection schemes, targeted to a spectrum of attacks [61]. This method was than improved by adopting column matching technique so as the obtained result was a detection scheme for all the attacks in the dictionary. The experimental part revealed that a certain feedback polynomial is the most appropriate to be used in order to detect the attacks from the developed attacks dictionary. A detection method has been provided specific for the case of intelligent grid architecture. Each attack has associated a signature. Based on the proposed detection scheme, each signature is detected when appears in a deteriorated message.

5. PERFORMANCE METRICS FOR INFORMATION SECURITY IN INTELLIGENT GRID

”You can’t control what you can’t measure.” Tom DeMarco [110]

In the previous chapters it has been proposed a model called intelligent grid in order to address the shortcomings of the existent embient intelligence. It has been demonstrated [61, 62, 65] that security and dependability are complex problems because of the distributed nature of such a collection of networks. Reliability is another main problem, because unreliability is inherent to the disappearing electronics concept. Therefore a design for reliability emphasizes as a method for attaining a secure non-traditional system [88, 90] such as intelligent grid. Also, different techniques have been provided so as to improve the security and reliability of intelligent grid. The first step was to classify the threats on such networks and provide a threats model.

In order to realize the authentication of the sender, a solution based on digital signatures has been adopted based on hash functions. In terms of owner-consumer paradigm, a possible attack can be one of impersonating the consumer. In this case, the execution of a malicious mobile agent can be realized after a previous transfer on the consumer container. A solution to such a case is to adopt a broker. In the case of intelligent grid this is represented by a distinctive agent targeted to guarding the data traffic on the network. Another technique for achieving fault tolerance in intelligent grid was to construct dedicated detection schemes for different intrusions. Such schemes aim at detecting those attacks that can be identified by a signature.

The following problem to be address in order to achieve a fault tolerant, reliable system is to measure in quantifiable terms, the improvement achieved by investing in the presented security methods. In other words, specific metrics should be employed to determine the effectiveness of security investments and strategies.

There are two categories of information security metrics, namely qualitative metrics and quantitative measures. The first category use subjective evaluations of risk, such as low, medium, high [37] and are useful for establishing a security project’s progress. Even so, these metrics cannot be employed in order to make significant risk-management decisions. Risk analysis and risk management support themselves by means of quantitative performance metrics that provide cost-benefit analyses and return-on-investment (ROI) estimations. In other words such metrics can provide the degree of compliance with some security requirements. These requirements can be formulated as the number or percent of systems certified and also those that can measure the real effectiveness of security controls. Such quantitative metrics enable peculiar defensible and awareness [22] management decisions concerning information security investments and strategies.

In order to measure the obtained improvements towards the security and reliability of an intelligent system (such as the proposed architecture intelligent grid), different measuring techniques are expected to be presented. However, it has been demonstrated [37] that statistical data regarding the threats, the numbers of

attacks, the consequences of such attacks and the action of threats, all these necessary to calculate values for security information metrics, are still not available. The effect is that the capacity to make use of security performance metrics has provided leisurely advancement. In the same time, it is argued [6, 20] that it is essential to employ risk management approaches based on metrics in order to manage information security in a given system.

5.1. Information assurance

A reliable program for information assurance can be measured in terms of the malicious attacks that can be avoided and in the same time by the implied losses of these attacks prevented of taking place. It is a question of establishing the effectiveness of investment in a program for information assurance. In [45] it is argued that expected loss offers a convenient metric in order to establish if it guarantees an investment in information security. The expected loss can be computed in terms of the security investment. In this regard, let be E_{IB} the investment benefit, defined like in equation (5.1).

$$E_{IB} = (\text{Expected loss before investment}) - (\text{Expected loss after investment}) \quad (5.1)$$

In order to obtain a confident investment, the condition is that $E_{IB} > 0$. The net benefit of a security investment can be computed as in equation (5.2).

$$E_{IBnet} = (\text{Expected loss before investment}) - (\text{Expected loss after investment}) - \text{Investment} \quad (5.2)$$

The same condition is applied in this case, namely $E_{IBnet} > 0$ in order to warrants the security investment. In [37] it is stated that these are products of probabilities and consequences because are expectations of investments. Therefore the probability of a successful attack offers the loss implied by a successful attack. The expected loss before investment and expected loss after investment can be computed in terms of probabilities as in equation (5.3). In this set of equations p_0 and p_I are the probabilities of a successful attack before and after a security investment. It is interesting to notice that $v(t)$ is a function which measures a successful attack's total economic consequences. The final form of E_{IBnet} is presented in equation (5.4).

$$E_0 = p_0 \times v(t), \quad E_I = p_I \times v(t) \quad (5.3)$$

$$E_{IBnet} = E_0 - E_I - I = [p_0 \times v(t)] - [p_I \times v(t)] - I \;\; (p_0 - p_I) \times v(t) - I \quad (5.4)$$

The following problem is to establish the probabilities from the last equation. In this regard, the medical communities of biostatisticians have proposed different techniques and methodologies which can be used in order to measure the effectiveness of security countermeasures. The basic feature of such methodologies is the study of different groups upon which different drugs have been administered. A plethora of relevant factors are employed such as age, weight, genetic background and so forth. In the case of information infrastructures different types of systems, applications, users' preferences and so on should be taken into account.

In [37] it is presented that the biostatistics approaches can be channelized in a manner that can provide the possibility to measure countermeasures' effectiveness and also to develop performance metrics. The methodology which presents the most promising results is *failure-time analysis*. This method can be used to study a minimum of two groups of systems. One of these groups will have no security enhancements, while the other groups are invested with different proposed enhancements in information security. Such particular technique is called random controlled trial. These randomized trials can provide conclusions regarding investments and security effectiveness and also the relative contributions of security countermeasures can be computed [34]. Another facility of randomized controlled trials is that they offer a quantitative approach for establishing and investigating the relationships among security policies, methodologies, technologies and the losses that their use seek to avoid.

5.1.1. Relative risk

For the case of intelligent grid architecture the main concerns appear from intentional malicious attacks. In this case, their actions are a threat to the reliability of the system.

In the case of a thorough analysis, the attacks are expected to occur from individuals or different groups. A typical methodology for such a situation [34, 37] is to consider two substructures of the information infrastructure, which are differentiated by two distinct collections of security countermeasures. The probability of system failure of the first group is p_0. The characteristic of this group is that it doesn't have any security investment in new security technology, while the second group, which probability is p_I, has such an investment. A specific design used to compare two collections with different characteristics is called the *risk matrix*. The independent samples of systems are noted n_1 and n_2, representing the unenhanced systems and enhanced systems. A set from each of n_1 and n_2 represents the number of systems who have failed in each group. Let these two sets be x_1 and x_2.

	Unenhanced sample	Enhanced sample	
Failure	x_1	x_2	M_1
Survival	$n_1 - x_1$	$n_2 - x_2$	M_2
	n_1	n_2	N

Fig.5.1. Risk matrix. $M_1 = x_1 + x_2$, $M_2 = (n_1 - x_1)+(n_2 - x_2)$, $N = n_1 + n_2$ **[66]**

From this risk matrix, it can be concluded that x_1 of n_1 unenhanced systems fail and x_2 of n_2 enhanced systems also fail. The sum of the first row provides the

total number of failures, while the sum of the values from the second row offers the total number of survivals.

5.1.2. Relative risk's measures

Different measures of relative risk can be derived from failure-time data. Each measure is a function of the probabilities of the positive response in the two collections [34]. Each function will indicate a variance which takes place from the null hypothesis H_0: $p_0 = p_I$. This variance occurs because it can be associated a link between the affiliation to a collection and the probability of system failure.

A first metric implied is the risk difference, *RD*, defined as the algebraic difference between the probabilities of system failure in the two groups, being equal with zero under the null hypothesis. When calculating the difference in the sample proportions the risk difference is obtained. Each of the terms from the algebraic difference is the natural estimator for p_0 and p_I.

$$\hat{RD} = \hat{p}_0 - \hat{p}_1 \tag{5.5}$$

$$\hat{p}_0 = \frac{x_0}{n_0}, \quad \hat{p}_1 = \frac{x_1}{n_1} \tag{5.6}$$

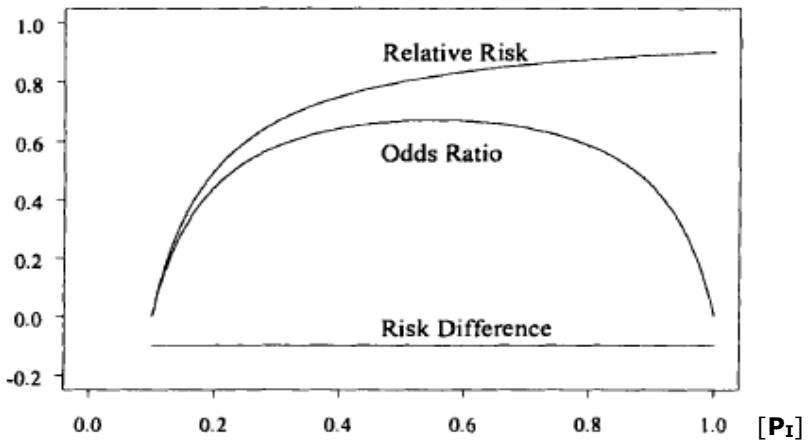

Fig. 5.2. Odds ratio and relative risk over a range of values for P_I for a fixed risk difference of -0.1 [34]

The relative risk (risk ratio), *RR*, is the ratio of the two probabilities, $RR = p_0/p_I$, being equal with zero under the null hypothesis. The estimation of RR is done using the ratio of the sample proportions. Whenever the relative risk is higher than one, the failure is more likely in the first group and when less than one, the failure is more likely in the second group. Similarly, if the relative risk is one, the risk of system failure is equal in both groups.

$$\hat{RR} = \frac{\hat{p}_0}{\hat{p}_1} \tag{5.7}$$

The odds ratio, *OR*, is the ratio of the odds of failure in the two collections. In order to estimate *OR* it is necessary to calculate relation (5.8). The estimation is given similar to RR with respect to value one.

$$\hat{OR} = \frac{\hat{p}_0 /(1-\hat{p}_0)}{\hat{p}_1 /(1-\hat{p}_1)} \tag{5.8}$$

In order to notice the relation between these three metrics, Fig.5.2 is presented. It displays the values of the odds ratio and relative risk over a range of values for P_I. The risk difference is constant at value -0.1. It can be noticed that while the P_I increases, so is the relative risk monotonically increasing toward the value of 1.0. It indicates proportionately smaller risk reduction. If P_0 tends to 0 and P_I tends to 1, the odds ratio *OR* is approximately 0. If P_I increases, the odds ratio increases toward the null value. What is interesting to notice is that the relative risk continues to increase as P_I tends to 1.

Following next, we propose to introduce different other metrics in order to obtain the entire domain as being the real line and also to provide a more precise granularity of the risk management and security improvement by enhancing a system with security techniques.

5.1.3. Risk distribution. Odds distribution

As it can be depicted from Table 5.1, both the relative risk and odds ratio have a domain of the positive real line and one for the null value. Because the domains are not symmetric about their null values, it is useful to introduce two other metrics, let's call them risk distribution and odds distribution, obtained by a log transformation. The result is that the domain becomes the real line.

$$\hat{L}_{RR} = \log R\hat{R} = \log \frac{x_0 n_1}{x_1 n_0} \tag{5.9}$$

$$\hat{L}_{OR} = \log \hat{OR} = \log \frac{\hat{p}_0 /(1-\hat{p}_0)}{\hat{p}_1 /(1-\hat{p}_1)} = \log \frac{x_0 (n_1 - x_1)}{x_1 (n_0 - x_0)} \tag{5.10}$$

Table 5.1. Measures for relative risk **[66]**

Risk metric	Expression	Domain	Null Value
Risk difference (RD)	p_o-p_I	[-1,1]	0
Relative risk (RR)	p_0/p_I	$(0,\infty)$	1
log RR (L_{RR})	log(p_o)-log(p_I)	$(-\infty, \infty)$	0
Odds ratio (OR)	$\frac{p_0 /(1-p_0)}{p_I /(1-p_I)}$	$(0,\infty)$	1
log OR (L_{OR})	$\log\frac{p_0}{1-p_0} - \log\frac{p_I}{1-p_I}$	$(-\infty, \infty)$	0

The importance of introducing these two metrics is that the confidence intervals are bounded with the domain of the parameters that are measured [66].

A useful metric is named attributable risk fraction, being defined as in equation (5.11). Its estimation is presented in equation (5.12).

$$AR = \frac{p_0 - p_I}{p_I} = RR - 1 \tag{5.11}$$

$$\hat{AR} = \hat{RR} - 1 = \frac{x_0 n_1}{x_1 n_0} - 1 \tag{5.12}$$

In defining this metric, $RR = p_0/p_I$ is the relative risk of failure among exposed versus non-exposed groups. Attributable risk fraction is a measure of the partial or fractional increase in the risk of failure when, for example, a given system is attacked by a certain type of threat. However such a metric as *AR* it doesn't provide an account of the prevalence of the risk in group and is presenting partial results compared to a necessity of an overall perspective of a certain type of attack.

In order to extend the results of *AR*, population attributable risk fraction can be employed. This metric is defined as the proportion of all cases of the failure in the group that are attributable to exposure to the risk factors, such as threats. In other words this measure addresses the question "What fraction of systems failures could be avoided if a certain risk (attack) could be completely eliminated in the group?" Practically this measure responds to the problem of estimating the impact of developing certain measurements of avoiding a peculiar attack. The methodology for this case is to consider a group of *N* systems such that the fraction exposed to a risk in the sample is expected to reflect the fraction exposed in the entire number of systems. In order to calculate *PAR*, in the general set of systems consider $a_1 = P(E)$ to be that fraction which is exposed to the risk factor. The population attributable risk is the proportion of all cases of the failure that would be prevented if the exposure to the risk factor (threat) were eliminated in the overall number of systems.

$$PAR = \frac{a_1(RR-1)}{1 + a_1(RR-1)} \tag{5.13}$$

In order to estimate PAR is necessary to calculate

$$\hat{PAR} = \frac{\hat{a}_1(\hat{RR}-1)}{1 + \hat{a}_1(\hat{RR}-1)} \tag{5.14}$$

$$\hat{a}_1 = \frac{n_1}{N} \tag{5.15}$$

$$\hat{RR} = \frac{\hat{p}_0}{\hat{p}_1} \tag{5.16}$$

$$\hat{PAR} = \frac{n_1^2 x_0 - x_1 n_0 n_1 N}{x_1 n_0 N + n_1^2 x_0 - x_1 n_0 n_1 N} \tag{5.17}$$

The final form for estimating PAR is presented in equation (5.17).

5.2. Case study. Intelligent grid

Intelligent grid consists of different networks combined in order to provide enhanced functionalities. Therefore, computational networks, but also sensor networks and multimedia networks are put together so as to offer particular gratifications that can not be offered if one of these networks is not taken into consideration in the architecture. In this case the risk matrix can be defined so as to point each particular network. For such a mixed system, we propose to use a risk matrix as defined in Table 5.2. Based on the risk matrix, the metrics for intelligent grid case can be defined. In the case of the sample proportions, it will take into consideration each subgroup that can fail from every single network.

The probabilities of the two groups, namely the one which is not invested with any security improvements and the one enhanced with security investments are presented in equations (5.18) and (5.19).

$$\hat{p}_{IG}^{0} = \frac{x_{1,3,5}^{0}}{n_{1,3,5}^{0}} \tag{5.18}$$

Table 5.2. Risk matrix for intelligent grid

	Unenhanced sample	Enhanced sample	
Failure PC	x_1	x_2	$x_1 + x_2$
Survival PC	n_1-x_1	n_2-x_2	$(n_1$-$x_1)$+$(n_1$-$x_1)$
Failure SN	x_3	x_4	$x_3 + x_4$
Survival SN	n_3-x_3	n_4-x_4	$(n_3$-$x_3)$+$(n_4$-$x_4)$
Failure MN	x_5	x_6	$x_5 + x_6$
Survival MN	n_5-x_5	n_6-x_6	$(n_5$-$x_5)$+$(n_6$-$x_6)$
	n_1+ n_3+ n_5	n_2+ n_4+ n_6	N = n_1+n_2+n_3+n_4+n_5+n_6

$$\hat{p}_{IG}^{1} = \frac{x_{2,4,6}^{1}}{n_{2,4,6}^{1}} \tag{5.19}$$

Once these two probabilities are defined, the following step is to determine the risk difference. This can be defined like in equations (5.20) and (5.21). It can be noted, that each term which appear in the probabilities is determined based in the entries of Table 5.2.

$$\hat{RD}_{IG} = \hat{p}_{IG}^{0} - \hat{p}_{IG}^{1} \tag{5.20}$$

$$\hat{RD}_{IG} = \frac{x^0_{1,3,5}}{n^0_{1,3,5}} - \frac{x^1_{2,4,6}}{n^1_{2,4,6}} \tag{5.21}$$

The relative risk for intelligent grid can be determined by calculating the relations from equations (5.22) and its final form from (5.23).

$$\hat{RR}_{IG} = \frac{\hat{p}^0_{IG}}{\hat{p}^1_{IG}} \tag{5.22}$$

$$\hat{RR}_{IG} = \frac{x^0_{1,3,5} * n^1_{2,4,6}}{x^1_{2,4,6} * n^0_{1,3,5}} \tag{5.23}$$

The odds ratio is presented in equation (5.24). The risk distribution and odds distribution are obtained when a logarithmic transformation is applied so as the domain becomes the real line.

$$\hat{OR}_{IG} = \frac{\hat{p}^0_{IG} /(1-\hat{p}^0_{IG})}{\hat{p}^1_{IG} /(1-\hat{p}^1_{IG})} \tag{5.24}$$

The estimation of attributable risk fraction is presented in relation (5.25). The first term of the equation can be computed starting from equation (5.22), namely the relative risk estimator for intelligent grid.

$$\hat{AR}_{IG} = \hat{RR}_{IG} - 1 = \frac{x^0_{1,3,5} * n^1_{2,4,6}}{x^1_{2,4,6} * n^0_{1,3,5}} - 1 \tag{5.25}$$

$$\hat{PAR}_{IG} = \frac{(n^1_{2,4,6})^2 * x^0_{1,3,5} - x^1_{2,4,6} * n^0_{1,3,5} * n^1_{2,4,6} * N}{x^1_{2,4,6} * n^0_{1,3,5} * N + (n^1_{2,4,6})^2 * x^0_{1,3,5} - x^1_{2,4,6} * n^0_{1,3,5} * n^1_{2,4,6} * N} \tag{5.26}$$

Finally, the estimator for population attributable risk in intelligent grid is presented in equation (5.26). In computing this final form it was taken into consideration the fact that population attributable risk is calculated with respect to relative risk.

5.2.1. Practical results

Intelligent grid is an architecture where different devices such as sensors, DSPs, PCs etc. are interacting in order to create an area meant to be controlled. Different attacks can threat the reliability, security and dependability of such architecture. Therefore certain techniques are to be adopted as presented in chapter 4 in order to avoid the effects of these attacks. By implementing such security techniques, we talk about an enhanced system.

Table 5.3. Risk matrix for general computing systems [37]

	Unenhanced sample	Enhanced sample	
Failure	86	63	149
Survival	14	37	51
	100	100	200

One of the components that form the intelligent grid is the general computing systems. Without losing from generality we can calculate the improvement obtained by enhancing the general computing systems with the proposed techniques taking into consideration the benchmark results obtained in [37]. In this case, two groups of systems were observed in the conditions of one population being enhanced with security mechanisms, while the other one functioning with a minimum of security mechanisms.

In others words, based on the proposed metrics we can determine the improvement obtained by using security techniques from the general computing systems point of view. In Table 5.3 it can be observed that 86 of 100 unenhanced systems have failed during the test period, while adopting security measures, only 63 of 100 enhanced systems have failed. Regarding the estimation of risk difference, we observe that the difference 0.86 – 0.63 = 0.23 shows that the excess probability of failure is 23. It can be concluded very interesting that the number of surviving systems will increase by a percentage of 23 if the particular security improvement is being applied. If we calculate the relative risk we can notice that is equal with 0.86/0.63 = 1.365 which demonstrates that the total number of safe systems increases by a percentage of 36.5.

In order to calculate the odds ratio, first of all it is to be calculated the odds of the enhanced sample: 0.63/0.37 = 1.702. Secondly, we obtain the odds for the unenhanced sample as being: 0.86/0.14 = 2.324. The odds ratio is 2.324/1.702 = 1.365. This result can be interpreted that the odds of improvement is increased by 36.5%.

The attributable risk fraction is 0.365 which means that the proportionate risk increases by a percentage of 3.65 when exposed to a threat. Further on, if we consider that 60% of this population was attacked by a certain type of attackers which can be identified by their signatures [65], the population attributable risk is 0.95. This means that 95% of the malfunctioning of the general computing elements may be attributable to the specific types of attacks.

5.3. Conclusions

Even from the first chapter it has been stressed out that, in order to obtain a secured system different factors are required like requirements, policy, and mechanisms. More than this, a separate phase of testing the effectiveness is required and also the integration of policies and mechanism into daily operations. The metrics for these last two constraints *"concentrate on measuring effectiveness and efficiency of implemented security controls and the impact of these controls on the organization's mission. These metrics concentrate on the evidence and results of testing and integration. Instead of measuring the percentage of approved security plans, these metrics concentrate on validating whether security controls, described in the security plans, are effective in protecting the organization's assets"* [111]. Such measures of relative risk of system failure have been presented and also

extended in this chapter. As Dow Williamson presented in [22], the security measures are presented at the management decisions level. In this regard, the measures of relative risk are efficiency and effectiveness metrics let mangers to decide how implemented security procedures will extend the lifetime of their assets in a reliable fashion for the entire lifecycle. The security metrics when combined with a value function $v(t)$ [37] for all the information assets that are possessed by an information infrastructure will illustrate the impact that the improvements will have at the total value of the information infrastructure. Therefore the offered metrics have wide-range applicability in terms of information technology. The question in this point was how to test a certain security improvement, the aim of design for testability being addressed. Based on metrics, the security of a general system can be tested and decisions can be recommended.

Particularly, this chapter showed a modality to apply specific metrics in order to establish the improvement obtained in an intelligent grid network after operating certain methods for enhancement the reliability and security of the architecture.

6. CONCLUSIONS

6.1. Thesis impact and contributions

The great challenge of this thesis was to study and provide different techniques so as to improve the existent ambient intelligence model with respect to the problems encountered in such a networked architecture. We have identified a solution to these challenges in the form of intelligent grid. The distributed support of second wave was used in order to emulate this architecture belonging to the third wave of computing.

The security of such a model poses a great challenge due to the fact that different networks are mixing in order to provide such an environment. In addition, these networks are generally deployed and then left unattended. All these aspects joined together make it unfeasible to directly apply the traditional security mechanisms. Therefore, there was a need to analyze and better understand the security requirements of these networks.

In order to perform a relevant simulation based assessment of architecture reliability, very accurate fault models must be realized. A rigorous study and analysis of threats has been proposed. Furthermore, the errors occurrence models are very important for the reliability assessment, thus an analysis of them is also required. The expanding profile of intelligent grid requires a reliable security system, capable of responding to any attack on resources. This was achieved by developing a comprehensive threat model. Without such a model security designers might concentrate their efforts on some threats while leaving the system vulnerable to others.

Different methods of attaining fault tolerance have been presented and discussed: redundancy and digital signatures based on hash functions. By using these two approaches, the increasing execution time is justified by the gain in the overall reliability and security of the system. In chapter four it has been presented a technique for detecting errors at message level exchanged by the agents. These errors were introduced in a malicious approach by crimeware scripts. A detailed example has been provided in order to illustrate how the correction takes place. In the last part of the chapter experimental results were presented. First of all, the technique was compared with dummy-algorithm in order to illustrate time application time. Specific remarks have been offered in order to illustrate the differences between the time execution of this algorithm and the one based on digital signatures.

In the second part of chapter four were provided dedicated detection schemes for different attacks on the proposed architecture. Another research line for this chapter was to establish which feedback polynomial is optimum to be used in order to build the before mentioned detection schemes. In this regard certain metrics have been used. It should be said also that to each attack is associated a digital signature based on which the theft identity is recognized. The improvements on detection algorithm, made us to construct a detection scheme for all the attacks from a signatures' dictionary, in a more optimal complexity fashion.

This report presents several contributions.

- First of all this report offers an overview of the current trends in networked systems and addresses the idea of an intelligent grid. We proposed this architecture as a *solution to the existing problems of ambient intelligence*. By transferring concepts from the existing models into this new *intelligent grid concept*, an architecture based on intelligent agents emerges. Using such a level of abstraction, many tasks specific to this mixed type of networks, become easy to implement. Therefore the problems of power consumption, portability, scalability, configurability and security, existent in ambient intelligence can be faced.
- However, given the unattended nature of ambient intelligent networks, they are vulnerable to a number of security attacks which could substantially degrade the performance of the network. Therefore, we offered *an up-to-date threats model* for the proposed architecture. Errors occurrence models are very important for the reliability assessment, thus an analysis of them was provided.
- In order to provide a dependable, secure and reliable system, the subset of *fault tolerance goal* has been addressed through methods of redundancy (in order to address the reliability and security) and digital signatures based on hash functions (as means for authentication at message exchange). Another implemented technique was that one of correcting the affected messages in intelligent grid. Their appliance was followed and certain conclusions were drawn.
- The presented report offers *an improved algorithm for constructing detection schemes by adopting a column matching approach*, and in the same time it offers *dedicated detection schemes* for different attacks to the intelligent grid architecture. These attacks are identified by their signatures, collected in a dictionary of attacks. The *improvement* proposed to the initial algorithm was made in order to construct a *single detection scheme for all the attacks*. The complexity of the proposed algorithm is superior to the one of the initial algorithm. *Such an algorithm can be further used in order to test different circuits (Circuits Under Test)*.
- Another contribution is the identifying of a certain feedback polynomial which is optimum for constructing the detection schemes. In order to come to the conclusion that this feedback polynomial is the optimum one, certain metrics have been used. By analyzing the data obtained from these specific metrics a conclusion regarding the feedback polynomial was presented.
- In order to aim at attaining a design for testability, different metrics were proposed, which *appliance doesn't limit to the case of system's security*, but can be used in other computer science areas.
- This report proposes an architecture that can address the concept of *sensor grid*. In this case specific problems of sensor grids such as routing, aggregation, querying diverse sensor network data can be address from the point of intelligent agents and dedicated middleware solutions.
- This report followed the standard steps to be implemented in order to obtain a dependable and secured architecture (Fig.6.1). The

above mentioned contributions are linking to different levels from the methodology of attaing a dependable and secured architecture.

This thesis is supported by the following published papers:

- **R. Bogdan**, V. Ancusa, M. Vladutiu**,** "Fault Tolerance Issues in Non-Traditional Grids Implemented with Intelligent Agents", Proceedings of the International Conference on Computer and Electrical Engineering (ICCEE08), Thailand, ISBN 978-0-7695-3504-3, pp. 912-917, 2008 (IEEE, ISI rank)

- **R. Bogdan**, M. Vladutiu, "Intrusions Detection in Intelligent Agent-Based Non-traditional Grids", Proceedings of the International Conference on Education Technology and Computer (ICETC09), Singapore, ISBN 978-0-7695-3609-5, pp. 116-122, 2009 (IEEE, ISI rank)

- **R. Bogdan**, M. Vladutiu, "Providing Security in Intelligent Agent-Based Grids by Means of Error Correction", Proceedings of the International Conference on Future Networks (ICFN 2009), Thailand, ISBN 978-1-4244-3579-1, pp. 233-239, 2009 (IEEE, ISI rank)

- **R. Bogdan**, V. Ancusa, M. Vladutiu, "Performance Metrics for Information Security in Intelligent Grid", Proceedings of the International Conference on Machine Learning and Computing (ICMLC09), Australia, 2009 (ISI rank)

- V. Ancusa, **R. Bogdan**, M. Vladutiu, "Redundancy at Link Level for Non-Traditional Grids Implemented with Intelligent Agents", Proceedings of the 4th International Conference on Networked Computing and Advanced Information Management (NCM08), South Korea, ISBN 978-0-7695-3322-3, Vol. 1, pp. 597-603, 2008 (IEEE, ISI rank)

- V. Ancusa, **R. Bogdan**, M. Vladutiu, "Discussing Redundancy Issues in Intelligent Agent-Based Non-traditional Grids", Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES08), Croatia, Vol. LNAI 5178, pp. 297-395, 2008 (ISI rank)

Additional accepted papers with relevance to the thesis:

- V. Ancusa, **R. Bogdan**, M. Vladutiu, "Discussing the Intelligent Agent Approach in Non-traditional Grids", Proceedings of the International Multi-Conference on Engineering and Technological Innovation, Florida, USA, ISBN 978-1-934272-46-6, Vol. I, pp. 87-92, 2008 (ISI rank)

- **R. Bogdan**, V. Ancusa, M. Vladutiu, "Possible Threats in an Intelligent Sensor Grid" , Proceedings of the 8th International Conference on Technical Informatics (CONTI 2008), Timisoara, Romania, ISSN 1844-539X, Vol. 2, pp. 51-57, 2008

- V. Ancusa, **R. Bogdan**, L. Susan, M. Vladutiu, "A Customized Population Screening Method for Osteoporosis and Osteoarthrosis" , Proceedings of the

8th International Conference on Technical Informatics (CONTI 2008), Timisoara, Romania, ISSN 1844-539X, Vol. 2, pp. 171-175, 2008

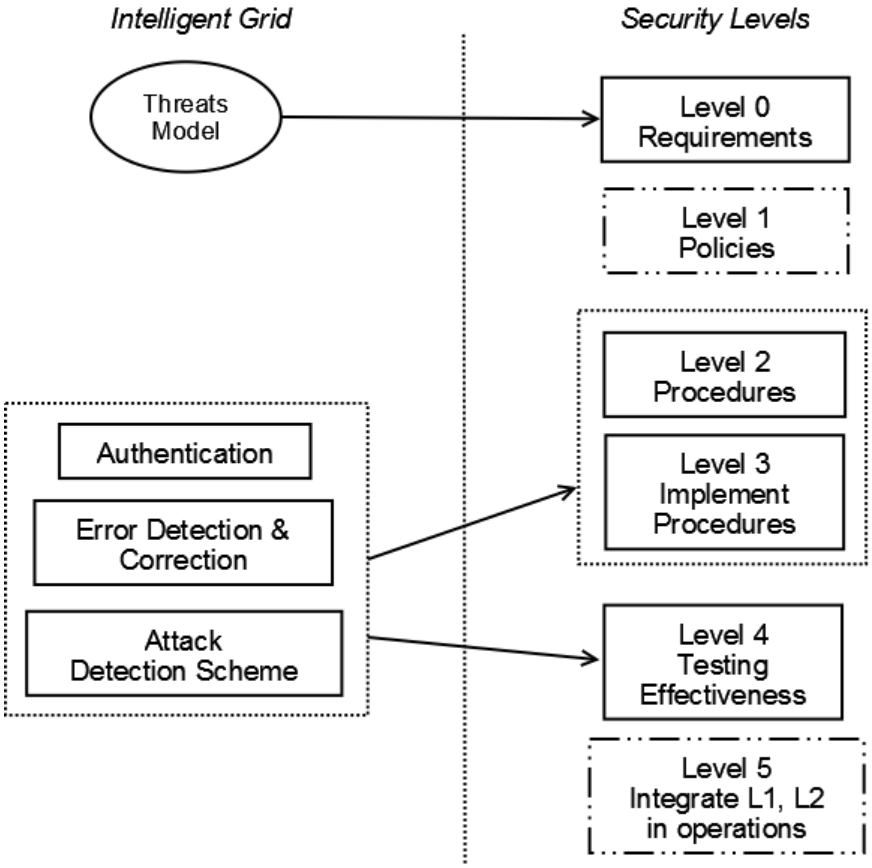

Fig.6.1. Parallel between security levels in intelligent grid and the standard security levels

Two Ph.D. reports were presented in the Computer Science and Engineering Department, Politehnica University of Timisoara:

- **R. Bogdan**, PhD Report 1, Politehnica University of Timisoara, July, 2008
- **R. Bogdan**, PhD Report 2, Politehnica University of Timisoara, February, 2009

This Ph.D. program was partially funded by **CNCSIS grant TD 54/2008**.

6.2. Future work

As it has been presented in the above subchapter, this thesis aimed at following the steps to implement a secure architecture. In this regard Fig.6.1 is relevant. This architecture, entitled intelligent grid, emerges as a solution to the existent problems identified in ambient intelligence and the methods and proposed

solutions were adopted and offered in order to follow the steps of obtaining a secured and dependable system.

The following steps to be addressed are:

- establishing what security policies should be adopted, for example in the matter of message communication or the acceptance of other devices to be part of the intelligent grid
- integrating all the levels into the final security level, namely having a completely functional secured intelligent grid
- extend the methodology of evaluating the security enhancement to the other components of intelligent grid so as to provide risk predictions for the entire complex network
- in the matter of security improvements, these directions are related to the possibility of researching a method of determining an optimum way of selecting the best feedback polynomial from a very large number of feedback polynomials; in this regard primitive feedback polynomials techniques are to be employed (e.g. MISR)
- another future direction is related to constantly improving an Intrusion Detection System with state-of-the-art signatures and attacks.

REFERENCES

[1] A. Boulis, C.C. Han, and M. B. Srivastava, ”Design and Implementation of a Framework for Programmable and Efficient Sensor Networks”, MobiSys 2003, San Franscisco, USA, 2003

[2] A. Neubauer, J. Freudenberger and V. Kuhn, Coding Theory, John Wiley & Sons, ISBN 978-0-470-02861-2, 2007

[3] Ahmar Abbas, Grid Computing: A Practical Guide to Technology and Applications, ISBN:1584502762, Charles River Media, 2004

[4] Alexandru Coman, Mario A. Nascimento and Jorg Sandera, ”Exploiting Redundancy in Sensor Networks for Energy Efficient Processing of Spatiotemporal Region Queries”, ACM Proceedings of the Conference on Information and Knowledge Management, Bremen, Germany, 2005

[5] Algirdas Avizienis, Jean-Claude Laprie, Brian Randell and Carl E. Landwehr, ”Basic Concepts and Taxonomy of Dependable and Secure Computing”, IEEE Transactions on Dependable and Secure Computing, Vol. 1, No. 1, pp. 11-33, 2004

[6] Andrew Jaquith, Security Metrics: Replacing Fear, Uncertainly and Doubt, Addison-Wesley Professional, ISBN 8-0-321-34998-9, 2007

[7] Andrew S. Tanenbaum, Computer Networks, Fourth Edition, Prentice Hall, ISBN 0-13-066102-3, 2003

[8] Anirban Chakrabarti, A. Damodaran and S. Sengupta, ”Grid Computing Security. A Taxonomy”, IEEE Security & Privacy, Vol. 6, Nr. 1, 2008

[9] Anirban Chakrabarti, Grid Computing Security, Springer-Verlag, ISBN 978-3-540-44492-3, 2007

[10] Barry Johnson, The Design and Analysis of Fault Tolerant Digital Systems, Addison Wesley, ISBN 978-0201075700, 1989

[11] Bill Gates, The Disappearing Computer. [Online]. http://www.microsoft.com/presspass/ofnote/11-02worldin2003.mspx, 2003

[12] Bogdan Carbunar, Ananth Grama, Jan Vitek and Octavian Carbunar, ”Coverage Preserving Redundancy Elimination in Sensor Networks”, IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, pp. 377-386, 2004

[13] Brian Halla, ”How the PC Will Disappear”, IEEE Computer, Vol. 31, No. 12, pp. 134-136, 1998

[14] C. C. Shen, C. Srisathapornphat and C. Jaikaeo, ”Sensor Information Networking Architecture and Applications”, IEEE Personal Communications, Vol. 8, No. 4, pp. 52–59, 2001

[15] Carl E. Landwehr, ”Cybersecurity and Artificial Intelligence”, IEEE Security & Privacy, Vol. 6, Nr. 5, 2008

[16] Chee-Yee Chong and Srikanta P. Kumar, ” Sensor Networks: Evolution, Opportunities,and Challenges”, Proceedings of the IEEE, Vol. 91, No. 8, 2003

[17] Chen-Khong Tham and Rajkumar Buyya, ”SensorGrid: Integrating Sensor Networks and Grid Computing”, CSI Communications, pp. 24-29, July 2005

[18] D. McKinney, ”New Hurdles for Vulnerability Disclosure”, IEEE Security & Privacy, Vol. 6, No. 2, pp. 76-78, 2008

[19] D. Petcu, ”Arhitecturi si tehnologii grid”, Eubeea, pp. 118-165, 2006

[20] Debra S. Herrmann, Complete Guide to Security and Privacy Metrics, Auerbach Publications, ISBN 0-8493-5402-1, 2007

[21] Dhiraj K. Pradhan and Sandeep K. Gupta, ”A New Framework for Designing and Analyzing BIST Techniques and Zero Aliasing Compression”, IEEE Transactions on Computers, Vol. 40, No. 6, pp. 743-763, 1991

[22] Dow Williamson, Annual Security Awareness Certification, SCIPP International. [Online]. http://www.brighttalk.com/channels/2098/view, 2009

[23] Eric M.Yeatman, Paul D. Mitcheson and Andrew S.Holmes, ”Micro-Engineered Devices for Motion Energy Harvesting”, International Electron Devices Meeting (IEDM 2007), pp. 375-378, USA, 2007

[24] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood, ”Developing multi-agent systems with JADE”, John Wiley& Sons, ISBN: 978-0-470-05747-6 (HB), 2007

[25] Fred B.Schneider, "Paradigms for distributed programming. Distributed system methods and tools for specification ", LNCS, Vol. 190, pp. 343-430, 1985

[26] Fred Boekhorst, ”Ambient Intelligence, the Next Paradigm for Consumer Electronics: How will it Affect Silicon?”, IEEE International Solid-State Circuits Conference (ISSCC 2002), Vol. 1, pp. 28-31, 2002

[27] G. Koch, ”Discovering Multi-Core: Extending the Benefits of Moore’s Law”, Technology Intel Magazine, 2005

[28] Gordon Bell, "Bell's Law for the Birth and Death of Computer Classes", Communications of the ACM, Vol. 51, No. 1, pp. 86-94, 2008

[29] Haowen Chan and Adrian Perrig, "Security and Privacy in Sensor Networks", IEEE Computer, Vol. 36, No. 10, pp. 103-105, 2003

[30] I. Foster and C. Kesselman, The Grid: Blueprint for a Future Computing Infrastructure, Morgan Kaufmann Publishers, ISBN 1-55860-475-8, 1998

[31] Ian Foster and Adriana Iamnitchi, "On Death, Taxes, and the Convergence of Peer-to-Peer and Grid Computing", Peer-to-Peer Systems II, LNCS, Springer Berlin, pp. 118-128, ISBN 978-3-540-40724-9, 2003

[32] Ian Foster, "What is the Grid? A Three Point Checklist", Argonne National Laboratory & University of Chicago, 2002

[33] Ian Foster, C. Kesselman and S. Tuecke, "The Anatomy of the Grid: Enabling Scalable Virtual Organizations", International J. Supercomputer Applications, Vol. 15, No. 3, 2001

[34] J. M. Lachin, Biostatistical Methods: The Assessment of Relative Risks, John Wiley & Sons, ISBN 0-471-36996-9, 2000

[35] J. Pikoulas, W. Buchanan, M. Mannion and K. Triantafyllopoulos, "An Intelligent Agent Security Intrusion System", 9th Annual IEEE International Conference and Workshop on the Engineering of Computer-Based Systems (ECBS 2002), pp. 94-99, 2002

[36] J. Rabaey, F. Burghardt, D. Steingart, M. Seeman and P. Wright, "Energy Harvesting - A Systems Perspective", International Electron Devices Meeting (IEDM 2007), pp. 363-366, USA, 2007

[37] J.C.H. Ryan and D.J. Ryan, "Performance Metrics for Information Security Risk Management", IEEE Security & Privacy, Vol. 6, Nr. 5, pp. 38-44, 2008

[38] Jan M. Rabaey, "Wireless Sensor and Consumer Multimedia Networks – A Story of Converging Trajectories?", IEEE Consumer Communications & Networking Conference (CCNC 2005), 2005

[39] Jeff Shneidman, Peter Pietzuch, Jonathan Ledlie, Mema Roussopoulos, Margo Seltzer and Matt Welsh, "Hourglass: An Infrastructure for Connecting Sensor Networks and Applications", Harvard Technical Report TR2104, Harvard University. [Online]. http://hourglass.eecs.harvard.edu, 2004

[40] Jennifer M. Schopf and Bill Nitzberg, "Grids: The Top Ten Questions", Scientific Programming, special issue on Grid Computing, Vol. 10, No. 2, pp. 103 – 111, 2002

[41] Jiancheng Ni, Zhishu Li, Zhonghe Gao and Jirong Sun, "Threats Analysis and Prevention for Grid and Web Service Security", Proceedings of the Eight

International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, Vol. 3, pp. 526 – 531, 2007

[42] K. Zhang, "Efficient Protocols for Signing Routing Messages", Proceedings of the Symposium on Network and Distributed System Security (NDSS), San Diego (CA), pp. 127-137, 1998

[43] Kay Romer, "Programming Paradigms and Middleware for Sensor Networks", GI/ITG Fachgespräch Sensornetze, Karlsruhe, 2004

[44] Klaus Krauter, Rajkumar Buyya and Muthucumaru Maheswaran, "A taxonomy and survey of grid resource management systems for distributed computing", Software - Practice And Experience, No. 32, pp. 135–164, 2002

[45] L. A. Gordon and M.P. Loeb, "The Economics of Information Security Investment", ACM Transactions on Information and System Security, Vol. 5, No. 4, pp. 438-457, 2002

[46] Leslie Lamport, Marshall Pease and Robert Shostak, "Reaching Agreement in the Presence of Faults", Journal of the Association for Computing Machinery, 1980

[47] LFSRTestbench, LFSR testbench. [Online]. http://users.ece.gatech.edu/~hamblen/book/LFSR.html, 2008

[48] M. Benattou and K. Tamine, "Intelligent Agents for Distributed Intrusion Detection System", Proceedings of World Academy of Science, Engineering and Technology, Vol. 6, pp. 190-193, 2005

[49] M. Humphrey, M. R. Thompson and K. R. Jackson, "Security for Grids", Proceedings of the IEEE, Vol. 93, No. 3, pp. 644-652, 2005

[50] M. Shao, S. Zhu, W. Zhang and G. Cao, "pDCS: Security and Privacy Support for Data-Centric Sensor Networks", 26th Annual IEEE Conference on Computer Communications (INFOCOM'07), 2007

[51] Markus Jakobsson and Zulfikar Ramzan, Crimeware: Understanding New Attacks and Defenses, Addison Wesley Professional, ISBN 978-0-321-50195-0, 2008

[52] Matt Bishop, " What Is Computer Security?", IEEE Security & Privacy, Vol. 1, No. 1, pp. 67-69, 2003

[53] Mitchel M. Waldrop, "Grid Computing", Technology Review, No. 3, 2002

[54] Mitrajit Chatterjee and Dhiraj K. Pradhan, "A BIST Pattern Generator Design for Near-Perfect Fault Coverage", IEEE Transactions on Computers, Vol. 52, No. 12, pp. 1543-1557, 2003

[55] OE Magazine, Optoelectronic Markets. [Online].

http://oemagazine.com/fromTheMagazine/jan05/busspot.html, 2005

[56] P. Bonnet, J. E. Gehrke, and P. Seshadri, "Querying the PhysicalWorld", IEEE Personal Communications, Vol. 7, Nr. 5, pp. 10–15, 2000

[57] P. Levis and D. Culler, "Mate: A Tiny Virtual Machine for Sensor Networks", International Conference on Architectural Support for Programming Languages and Operating Systems, ASPLOS X, San Jose, USA, 2002

[58] P. Townend and J.Xu, "Dependability in Grids", IEEE Distributed Systems Online, Vol. 6, No. 12, 2005

[59] P. Townend, J. Xu, "Fault Tolerance within a Grid Environment", Proceedings of AHM Conference. [Online]. http://www.nesc.ac.uk/events/ahm2003/AHMCD/pdf/063.pdf, 2003

[60] Petr Fiser, Hana Kubatova, "Multiple-Vector Column-Matching BIST Design Method", Proceedings of the IEEE Design and Diagnostics of Electronic Circuits and Systems (DDECS'06), pp. 266-271, 2006

[61] **R. Bogdan**, M. Vladutiu, "Intrusions Detection in Intelligent Agent-Based Non-traditional Grids", Proceedings of the International Conference on Education Technology and Computer (ICETC09), Singapore, ISBN 978-0-7695-3609-5, pp. 116-122, 2009

[62] **R. Bogdan**, M. Vladutiu, "Providing Security in Intelligent Agent-Based Grids by Means of Error Correction", Proceedings of the International Conference on Future Networks (ICFN 2009), Thailand, ISBN 978-1-4244-3579-1, pp. 233-239, 2009

[63] **R. Bogdan**, PhD Report 1, Politehnica University of Timisoara, July, 2008

[64] **R. Bogdan**, PhD Report 2, Politehnica University of Timisoara, February, 2009

[65] **R. Bogdan**, V. Ancusa, M. Vladutiu**,** "Fault Tolerance Issues in Non-Traditional Grids Implemented with Intelligent Agents", Proceedings of the International Conference on Computer and Electrical Engineering (ICCEE08), Thailand, ISBN 978-0-7695-3504-3, pp. 912-917, 2008

[66] **R. Bogdan**, V. Ancusa, M. Vladutiu, "Performance Metrics for Information Security in Intelligent Grid", Proceedings of the International Conference on Machine Learning and Computing (ICMLC09), Australia, ISBN 978-1-84626-018-6, 2009

[67] **R. Bogdan**, V. Ancusa, M. Vladutiu, "Possible Threats in an Intelligent Sensor Grid" , Proceedings of the 8th International Conference on Technical Informatics (CONTI 2008), Timisoara, Romania, ISSN 1844-539X, Vol. 2, pp. 51-57, 2008

[68] R. D. Schlichting, "Dependability and the Grid. Issues and Challenges", Proceedings of the International Conference on Dependable Systems and

Networks (DSN'02), pp. 263-273, Washington DC, USA, 2002

[69] R. Marculescu, J. Rabaey, A. Sangiovanni-Vincentelli, "Is "Network" the Next "Big Idea" in Design?", Proceedings of the Design, Automation and Test in Europe, DATE '06, Vol. 1, pp. 1–3, 2006

[70] R. Swanson, "Future Developments in Silicon Solar Cells", International Electron Devices Meeting (IEDM 2007), pp. 359-362, USA, 2007

[71] R.Venkatasubramanian, Cynthia Watkins, David Stokes, John Posthill and Chris Caylor, "Energy Harvesting for Electronics with Thermoelectric Devices using Nanoscale Materials", International Electron Devices Meeting (IEDM 2007), pp. 367-370, USA, 2007

[72] Rodica Tirtea, Integration at middleware level of fault tolerance for distributed systems, Katholieke Universiteit Leuven, PhD Thesis, 2005

[73] S. Li, S. H. Son and J. A. Stankovic, "Event Detection Services Using Data Service Middleware in Distributed Sensor Networks", Information Processing in Sensor Networks (IPSN '03), Palo Alto, USA, 2003

[74] S. Pai, S. Bermudez, S. Wicker, M. Meingast, T. Roosta, S. Sastry and D. Mulligan, "Transactional Confidentiality in Sensor Networks", IEEE Security & Privacy, Vol. 6, No. 4, pp. 28-35, 2008

[75] S. R. Madden, M. J. Franklin, J. M. Hellerstein and W. Hong, "TAG: a Tiny Aggregation Service for Ad-Hoc Sensor Networks", 5th Symposium on Operating System Design and Implementation (OSDI 2002), Boston, USA, 2002

[76] Shashank Khanvilkar, Faisal Bashir, Dan Schonfeld and Ashfaq Khokhar, Multimedia Networks and Communication - The Electrical Engineering Handbook, Wai Chen - Academic Press, ISBN 0-12-170960-4, 2004

[77] Sheng-bo Xu, Jeroen Doumen, Henk van Tilborg, "On the Security of Digital Signature Schemes Based on Error-Correcting Codes", Designs, Codes and Cryptography, Springer, Vol. 28, No. 2, pp. 187-199, 2003

[78] Shu Lin and D. Costello Jr., Error Control Coding, 2nd Edition, Prentice-Hall, ISBN 9780130426727, 2004

[79] Sujoy Basu, Sameer Adhikari, Raj Kumar, Yong Yan, Roland Hochmuth and Bruce E. Blaho, "mmGrid: Distributed Resource Management Infrastructure for Multimedia Applications", 17th International Parallel and Distributed Processing Symposium, Nice, France, 2003

[80] Syed Naqvi and Michel Riguidel, "Grid Security Services Simulator (G3S) — A Simulation Tool for the Design and Analysis of Grid Security Solutions", Proceedings of the First International Conference on e-Science and Grid Computing (e-Science'05), pp. 421-428, 2005

[81] Syed Naqvi, "Grid Security – Principles and Practices", Riga, Latvia, 2007

[82] Syed Naqvi, Philippe Massonet and Alvaro Arenas, "Pragmatic Security Analysis of the Grid - A Requirements Engineering Perspective", Proceedings of the 4th International Conference on Information Systems Security, 2006

[83] Symantec, Symantec Internet Security Threat Report Edition XII. [Online]. http://www.symantec.com/threatreport, 2008

[84] T. Lewis, Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation, Wiley-Interscience, ISBN 978-0-471-78628-3, 2006

[85] T. Roosta, S. Shieh and S. Sastry, "Taxonomy of Security Attacks on Sensor Networks", First IEEE International Conference on System Integration and Reliability Improvements, Hanoi, Vietnam, 2006

[86] The Honeynet Project, Honeynet Project Challenges. [Online]. http://project.honeynet.org/, 2008

[87] V. Ancusa, **R. Bogdan**, L. Susan, M. Vladutiu, "A Customized Population Screening Method for Osteoporosis and Osteoarthrosis" , Proceedings of the 8th International Conference on Technical Informatics (CONTI 2008), Timisoara, Romania, ISSN 1844-539X, Vol. 2, pp. 171-175, 2008

[88] V. Ancusa, **R. Bogdan**, M. Vladutiu, "Discussing Redundancy Issues in Intelligent Agent-Based Non-traditional Grids", Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES08), Croatia, Vol. LNAI 5178, pp. 297-395, 2008

[89] V. Ancusa, **R. Bogdan**, M. Vladutiu, "Discussing the Intelligent Agent Approach in Non-traditional Grids", Proceedings of the International Multi-Conference on Engineering and Technological Innovation, Florida, USA, ISBN 978-1-934272-46-6, Vol. I, pp. 87-92, 2008

[90] V. Ancusa, **R. Bogdan**, M. Vladutiu, "Redundancy at Link Level for Non-Traditional Grids Implemented with Intelligent Agents", Proceedings of the 4th International Conference on Networked Computing and Advanced Information Management (NCM08), South Korea, ISBN 978-0-7695-3322-3, Vol. 1, pp. 597-603, 2008

[91] V. Welch, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman and S. Tuecke, "Security for Grid Services", Proceedings of the Twelfth International Symposium on High Performance Distributed Computing (HPDC-12), IEEE Press, pp. 48-57, 2003

[92] W. Hasselbring and R. Reussner, "Toward Trustworthy Software Systems", Computer, pp. 91-92, April, 2006

[93] W. Kinsner, "Compression and its metrics for multimedia", Proceedings of

First IEEE International Conference on Cognitive Informatic, pp. 107-121, 2002

[94] W. Weber, C. Braun, J. Dienstuhl, R. Glaser, Y. Gsottberger, B. Knoll, C. Lauterbach, D. Leitner, M. X. Shi, M. Schnell, D. Savio, G. Stromberg and M. Verbeck, "Disappearing electronics and the return of the physical world", IEEE International Symposium on VLSI Technology (VLSI-TSA-Tech), pp. 45-48, 2005

[95] W. Weber, J.M. Rabaey and E. Aerts, Ambient Intelligence, Springer-Verlag, ISBN 978-3-540-23867-6, 2005

[96] William Hoarau, Sébastien Tixeuil, Luis Silva, "Fault-Injection and Dependability Benchmarking for Grid Computing Middleware", Integrated Research in GRID Computing - CoreGRID Integration Workshop, Springer-Verlag, ISBN 978-0-387-47656-8, USA, 2007

[97] Y. Demchenko, L. Gommans, C. Laat, B. Oudenaarde, "Web services and grid security vulnerabilities and threats analysis and model", Proceedings of the 6th IEEE/ACM International Workshop on Grid Computing, pp. 262-267, 2005

[98] Y. Qian, J. Joshi, D. Tipper and P. Krishnamurty, Information Assurance. Dependability and Security in Networked Systems, Morgan Kaufmann, ISBN 978-0-12-373566-9, 2008

[99] Y. Xiao, Security in distributed, grid, mobile, and pervasive computing, Auerbach Publications, ISBN 0-8493-7921-0, 2007

[100] Y.C. Hu, A. Perrig and D.B. Johnson, "Wormhole detection in wireless ad hoc networks", Technical Report TR01-384, Department of Computer Science, Rice University, 2002

[101] Zhong Lin Wang, "From Nanogenerators to Nano-Piezotronics", International Electron Devices Meeting (IEDM 2007), pp. 367-374, USA, 2007

[102] (2005) International Technology Roadmap for Semiconductors. [Online]. http://www.itrs.net/Links/2005ITRS/ERD2005.pdf

[103] (2007) International Technology Roadmap for Semiconductors. [Online]. http://www.itrs.net/Links/2005ITRS/ERD2005.pdf

[104] S. Feuerstack, M. Blumendorf, G. Lehmann, and S. Albayrak, "Seamless Home Services", Proceedings of the First International Conference on Ambient Intelligence Development (AmID'06), pp. 1-11, 2006

[105] M. N. Huhns, L. M. Stephens, "Multiagent Systems and Societies of Agents". [Online]. http://citeseerx.ist.psu.edu/, 1999

[106] R. Guerraoui, A. Schiper, "The Generic Consensus Service", IEEE Transactions on Software Engineering, Vol. 27, No. 1, 2001

[107] V. Ancusa, ”Problema consensului in calcul tolerant la erori”, Politehnica University of Timisoara, 2009

[108] Fred B. Schneider, ”Accountability for Perfection”, IEEE Security & Privacy, Vol. 7, Nr. 2, pp. 3-4, 2009

[109] T. R. N. Rao, E. Fujiwara, ”Error-Control Coding for Computer Systems”, Prentice-Hall International, 1989

[110] T. DeMarco, ”Controlling Software Projects: Management, Measurement & Estimation”, Yourdon Press, 1982

[111] M. Swanson, N. Bartol, J. Sabato, J. Hash, L. Graffo, ”Security Metrics Guide for Information Technology Systems: Special Publication 800-55”, US National Institute of Standards and Technology, pp. 1-5, 2003

[112] Ross Anderson, ”Security Engineering”, Wiley, pp. 25-32, 2008

[113] D. E. Geer Jr., D. G. Conway, ”Security Is a Subset of Reliability”, IEEE Security & Privacy, Vol. 6, Nr. 6, 2008

[114] M. Zhivich, R. K. Cunningham, ”The Real Cost of Software Errors”, IEEE Security & Privacy, Vol. 7, Nr. 2, pp. 87-90, 2009

LIST OF PUBLICATIONS

Conference proceedings

- **R. Bogdan**, V. Ancusa, M. Vladutiu**,** ”Fault Tolerance Issues in Non-Traditional Grids Implemented with Intelligent Agents”, Proceedings of the International Conference on Computer and Electrical Engineering (ICCEE08), Thailand, ISBN 978-0-7695-3504-3, pp. 912-917, 2008 (IEEE, ISI rank)

- **R. Bogdan**, M. Vladutiu, ”Intrusions Detection in Intelligent Agent-Based Non-traditional Grids”, Proceedings of the International Conference on Education Technology and Computer (ICETC09), Singapore, ISBN 978-0-7695-3609-5, pp. 116-122, 2009 (IEEE, ISI rank)

- **R. Bogdan**, M. Vladutiu, ”Providing Security in Intelligent Agent-Based Grids by Means of Error Correction”, Proceedings of the International Conference on Future Networks (ICFN 2009), Thailand, ISBN 978-1-4244-3579-1, pp. 233-239, 2009 (IEEE, ISI rank)

- **R. Bogdan**, V. Ancusa, M. Vladutiu, ”Performance Metrics for Information Security in Intelligent Grid”, Proceedings of the International Conference on Machine Learning and Computing (ICMLC09), Australia, ISBN 978-1-84626-018-6, 2009 (ISI rank)

- **R. Bogdan**, V. Ancusa, M. Vladutiu, ”Possible Threats in an Intelligent Sensor Grid” , Proceedings of the 8th International Conference on Technical Informatics (CONTI 2008), Timisoara, Romania, ISSN 1844-539X, Vol. 2, pp. 51-57, 2008

- V. Ancusa, **R. Bogdan**, M. Vladutiu, ”Redundancy at Link Level for Non-Traditional Grids Implemented with Intelligent Agents”, Proceedings of the 4th International Conference on Networked Computing and Advanced Information Management (NCM08), South Korea, ISBN 978-0-7695-3322-3, Vol. 1, pp. 597-603, 2008 (IEEE, ISI rank)

- V. Ancusa, **R. Bogdan**, M. Vladutiu, ”Discussing Redundancy Issues in Intelligent Agent-Based Non-traditional Grids”, Proceedings of the 12th International Conference on Knowledge-Based Intelligent Information and Engineering Systems (KES08), Croatia, Vol. LNAI 5178, pp. 297-395, 2008 (ISI rank)

- V. Ancusa, **R. Bogdan**, M. Vladutiu, ”Discussing the Intelligent Agent Approach in Non-traditional Grids”, Proceedings of the International Multi-Conference on Engineering and Technological Innovation, Florida, USA, ISBN 978-1-934272-46-6, Vol. I, pp. 87-92, 2008 (ISI rank)

- V. Ancusa, **R. Bogdan**, L. Susan, M. Vladutiu, ”A Customized Population Screening Method for Osteoporosis and Osteoarthrosis” , Proceedings of the 8th International Conference on Technical Informatics (CONTI 2008), Timisoara, Romania, ISSN 1844-539X, Vol. 2, pp. 171-175, 2008