Wireless Communications: Security Management Against Cloned Cellular Phones

Mirela Sechi Moretti Annoni Notare
Fernando Augusto da Silva Cruz
Bernardo Gonçalves Riso
Carlos Becker Westphall

Federal University of Santa Catarina (UFSC) - Technological Center (CTC) - Network and Management Laboratory (LRG)
P.O. Box 476, Florianópolis SC, Brazil, ZIP 88040 970, +55(48)331-9739, Fax (48)331-9770
www.lrg.ufsc.br, {mirela, cruz, riso, westphal}@lrg.ufsc.br

Abstract: **This work presents the development of a distributed security management system for telecommunication networks. The system consists in reducing the use of cloned mobile telephones (same both number and series of a genuine phone – a perfect copy) using three main techniques: (1) An ISO Formal Technique (LOTOS) is used to specify and validate the system through the Eucalyptus software employment. The validation process includes exhaustive and interactive simulations, testing and verifications in order to guarantee the correctness of the system; (2) A Pattern Recognition Technique is used to classify the telephone users into classes. From this classification it is easier to identify if a call does not correspond to the patterns of a specific user, and thus identify whether a nongenuine caller made the call. MatLab software was employed to implement the classification algorithms; and (3) Distributed Object Technique is used for the implementation of this distributed system (i.e., manager and agents), with CORBA support, considering TMN and ATM technologies, by VisiBroker and JDK software employment.**

Keywords: Distributed Network Management, Telecommunication Security, Formal Description Technique, Pattern Recognition, CORBA.

1. INTRODUCTION

Considering the five areas of network management, i.e., configuration, failures, performance, accounting and security, the last area has deserved prominence in the current studies and constitutes the scope of this work. The security management service is responsible for providing a safe environment for both the operation and management of resources in a domain [14]. This work seeks to augment the security in telecommunication networks, avoiding frauds of cloned mobile phones. In order to program a nongenuine mobile cloned phone in such a way as to debit calls from a genuine mobile phone, one only needs to buy a piece of portable radio equipment called a scanner, which registers the frequency in which mobile phones operate in its immediate surroundings. The person committing the fraud may, for example, park his car around a shopping center, jot down various frequencies, transfer the data to clones and then pass them on to whoever may be interested.

The present work makes use of **formal description techniques** (FDT) to specify, validate and translate from specification code to implementation code. The specification is made in stepped refinements, using automatic tools to verify each refinement. LOTOS (Language of Temporal Ordering Specification) is the FDT used by the Eucalyptus Toolbox 2.3/CADP/Aldébaran 97b-19 Liege employment. In addition, **pattern recognition techniques** are used to classify the telephone users into classes according to their usage logs. Such logs contain the relevant characteristics for every call made by the user. From this classification it is easier to identify if a call does not correspond to the patterns of a specific user, and thus, identify whether a nongenuine caller made the call. As a consequence, the immediate identification of a fraud (instead of at the moment of receiving the monthly bill) will reduce losses for both users and carriers. We are convinced that the distributed systems which make use of this classified data base can uncover fraud with greater ease than conventional systems, when a call is outside of the standard pattern of a particular user, that is, when a possible fraud occurs. Pattern Recognition techniques are used by the MatLab tool employment. With this software, neural network algorithms (such as k-means, p-nearest neighbor and gauss) are implemented. Moreover, due to the characteristics of the telecommunication networks, CORBA and Java technologies are considered too.

This work is subject of the first author PhD Thesis. This paper is structured in the following manner. In Section 2 the formal description technique employment for specification and validation is presented. In Section 3 the pattern recognition technique usage for users classification is shown. In section 4 some relevant aspects of C++ and Java languages with CORBA support used for implementation are described. In Section 5 the conclusions are discussed, and, in addition, future works are listed. And finally, in Section 6 the bibliographical references are listed.

2. USING FORMAL DESCRIPTION TECHNIQUES

This work uses the LOTOS [1] Formal Description Technique, an ISO and actual standard which can describe both abstract data types and behavior, to enhance rigor in the procedures to obtain specification, validation (simulations, testing, verifications) and automatic translation from LOTOS code to C code. The simulations include exhaustive and interactive issues. Testing includes searches of event sequences, for instance. Verification includes deadlocks absence proofs as well as equivalence proofs of refined specifications. In this work, attention is given to the behavior aspects but do not include the abstract data types description. This position is justified due the great number of works related with data types (using ASN.1/GDMO – Abstract Syntax Notation One/Guidelines for the Definition of Managed Objects, for instance) and the lack of works related with the behavior formal description.

A. The SSCC Most Abstract Specification

Initially, in the highest abstraction level, the Security System to avoid the Cloning of Cellular - SSCC [13] (part of the SSTCC System – Security System for Telecommunication against Cloning and to avoid debtors) can be observed as a black box, with two communication gates (gate `mail` and gate `phone`), to send messages to the users.

The gate mail is used by the SSCC to send alarms of possible frauds to the user by surface mail. The gate phone allows the SSCC to use the mobile phone to send the same alarm. The specific advantage of sending alarms by phone is the immediate notification, the specific advantage of mail alarm is security. The SSCC system is always active, designating an infinite range of behavior, and suggests a LOTOS specification with the noexit functionality:

```
specification SSCC[mail,phone]:noexit
behaviour SSCC[mail,phone] where
   process SSCC[mail,phone]:noexit:=
     mail;(phone;SSCC[mail,phone]    []    i;SSCC[mail,phone])
endproc
endspec
```

The behavior of the SSCC system is defined by the process SSCC, that can execute an action in the gate mail, to send an alarm by surface mail (we consider this action always possible); the sequence is followed by a nondeterministic choice with two options. The first option is related to the alarm sent by mobile phone, in the gate phone (this action is not always possible) and, following, the process SSCC is called recursively in order to treat another case. Because the first option may not be successful - phones do not work properly, out of their area, for instance, after a period of time, an internal action i occurs (not observed) and the process SSCC is called recursively.

The most abstract specification of the SSCC system corresponds to a formalization of the user requirements of this system. It is the basis for future refinements of the project.

B. SSCC Specification Refinement

The SSCC system can be detailed in order to consider two of its most important components: the Manager (represented by the MANAGER process) and the Managed Sites Set (represented by the SITES_SET process). This refined conception is identified by SSCC_REF.

The process SITES_SET uses the gate notif to send notifications to the MANAGER process. This, in turn, after receiving a notification (by the gate notif) sends alarms to the users (by the gates phone and mail). SSCC_REF identifies the LOTOS specification of this refined conception.

```
specification SSCC_REF[mail,phone]:noexit
behaviour SSCC_REF[mail,phone]: where
   process SSCC_REF[mail,phone]:noexit:=
     hide notif in  SITES_SET[notif]   ||[notif]||
     MANAGER[notif,mail,phone]    where
     process SITES_SET[notif]:noexit:=...endproc
     process MANAGER[notif,mail,phone]:noexit:...endproc
   endproc
endspec
```

The behaviour of the SITES_SET process can be specified in LOTOS as follows: notif;SITES_SET[notif]. In this manner, an infinite succession of notifications can be made. The MANAGER process can be specified in LOTOS as follows: notif;mail;(phone;MANAGER[notif,mail,phone] [] i;MANAGER[notif,mail,phone]). And then, it sends alarms to the users after receiving notification.

The SITES_SET e MANAGER processes are combined with the general composition operator (|[…]|) usage. In this combination it is shown that both processes share all events that occur in the notif gate. The hide…in operator usage hides the notif internal gate, allowing us to compare the SSCC specification with the SSCC_REF specification; proving (by observational equivalence) that the last one is a correct refinement of the first.

C. Refinement of the SITES_SET Process

The SITES_SET process (Managed Sites Set) includes several instances of the same managed site model. Each of these instances corresponds to a LOTOS process that communicates with a MANAGER process (System Manager) through the notif gate. See the Fig. 2.1.

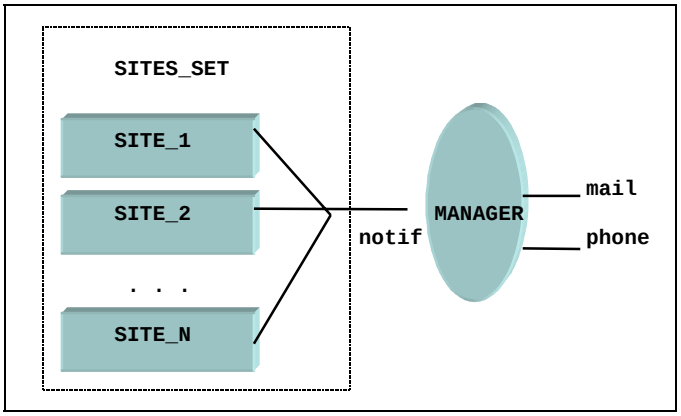

Fig. 2.1 - Detail of the SITES_SET process.

Consider that each managed site acts alone in sending the possible frauds alarms to the Manager, we can then use the independent composition operator (|||) to combine them, obtaining the following LOTOS representation: SITE_1[notif] ||| SITE_2[notif] ||| . . . ||| SITE_N[notif]. Each one of these managed sites constitutes a distributed agent.

D. Detail of a Managed Site

The adopted model for the managed sites conception considers them as three main elements: a Management Agent, a Reference Baseline and a File with the Telephone Calls. The SITE_J process represents a typical management site, with its three main elements. The LOTOS formal specification of this architecture can be presented as follows:

```
process SITE_J[notif]:noexit:=
 hide base_j,file_j in
(BASELINE_J[base_j] |||  CALLS_FILE_J[file_j])
|[base_j,file_j[| AGENT_J[base_j,file_j,notif] where ...
endproc
```

The hide … in operator employed in the detailed specification of the SITE_J process allows us to compare this specification with another more abstract, of the same site, that does not employ this operator. The BASELINE_J and CALLS_FILE_J processes act independently. Considering them in a set, these two processes share events, in the gates base_j and file_j, with the AGENT_J process. The BASELINE_J process can be run in an infinite sequence of events on its gate base_j: base_j; BASELINE[base_j]. In a

similar mode, the CALLS_FILE_J process can perform more actions on the file_j gate: file_j; CALLS_FILE[file_j]

The BASELINE_J and CALLS_FILE processes do not have complex behaviors. However, the AGENT_J process, can demonstrate more complex behavior: after checking the CALLS_FILE (through an action in the file_j gate), the AGENT_J verifies the certified user characteristics through access to the BASELINE_J (through an event in the file_j gate). Then, a nondeterministic choice with two options appears: this nondeterministic choice is internally solved by the AGENT_J process:

```
file_j;base_j;(i;AGENT_J[base_j,file_j,notif]      []
i;notif;AGENT_J[base_j,file_j,notif])
```

In the first option, the internal event i indicate that nothing abnormal has been detected. In that case, the AGENT_J process is called recursively. The second option represents the AGENT_J behavior when there is possibility of fraud.

E. SSCC Complete Refinement

The complete refined specification was validated and translated from LOTOS to C code, using the Eucalyptus Toolbox (see Section F hereafter). This C code will be helpful for the C and Java implementation presented in the Section 4.

F. Results of the Validation Using Eucalyptus Toolbox

The Eucalyptus toolbox [2] is a graphical user-interface (GUI) based on X-Windows for formal specification validation. The functions of the Eucalyptus toolbox include tools for:

- *analysis* - contains front-end tools performing lexical, syntatic, and static semantics analysis;
- *simulation* - the toolbox supports various forms of simulation;
- *exhaustive verification* - the toolbox allows the generation of the LTS (Labelled Transition Systems) corresponding to a LOTOS description. These LTSs can be analyzed and verified in several ways;
- *compositional verification* - due to the well-known state explosion problem, exhaustive generation of LTSs is not always possible. The Eucalyptus toolbox allows us to divide a LOTOS description into parallel processes;
- *graph drawing* - the toolbox contains several tools to display the LTSs, including PostScript representation;
- *test generation* - from the LOTOS descriptions, one can automatically generate test sequences, which will be used to assess the conformity of real implementations with respect to the original descriptions;
- *trace analysis* - the Eucalyptus toolbox allows us to verify whether a given trace (execution sequence) can be obtained from a LOTOS description); and,
- *code generation* - there are compilers to translate LOTOS types and process definitions into C code that can be executed and/or embedded in application programs.

The most expected result is to obtain the correction proof between refinements (such from SSCC specification to SSCC_DET specification). This procedure uses the CADP tool (Caesar Aldébaran Development Package) [4] included in the Eucalyptus toolbox for the generation of two automata: SSCC.AUT and SSCC_DET.AUT. (See its below, in FromState-Event-ToState format.) Using the observational equivalence option, the ´TRUE´ result is achieved. The ´TRUE´ result states that the processes under consideration are observational equivalent, i.e. the refinement is proved correct, proving that each refinement made is equivalent to the previous specification.

SSCC.AUT

```
des (0, 3, 2)
(0, MAIL, 1)
(1, PHONE, 0)
(1, i, 0)
```

SSCC_DET.AUT

```
des (0, 154, 49)
(0, i, 1)
(0, i, 2)
(1, i, 3)
...
(45, i, 33)
(45, i, 48)
(45, i, 18)
(46, PHONE, 19)
(46, i, 36)
(46, i, 48)
(46, i, 19)
(47, MAIL, 48)
(48, PHONE, 25)
(48, i, 25)
```

We consider the validation process composed by simulations, test and verifications. Both, simulations (were made exhaustive and interactive simulations) and testing (find events sequences, for instance) are able to found and identify errors – but do not prove correctness. For other hand, verifications go ahead, giving the correctness proof.

3. USING PATTERN RECOGNITION FOR CLASSIFICATION

This section involves the algorithm construction (see Section A), the algorithm implementation (see Section B) and the results of the classification (see Section C).

A. Algorithm Construction

The construction of an RBF (Radial Basis Function) in its more basic form involves three layers, whose output nodes form a linear combination of the Radial Basis Function (kernel) calculated by the nodes of the hidden layer.

The Radial Basis Function in the hidden layer produces a response for the input stimulus standard), that is, it produces response different from zero only when the input standard is within a small region located in the input space. The input is made from the source nodes (sensorial units). Each activation function requires a center and a numeric parameter. A function which can be used as activation is the Gauss function, while this network can be used to make decisions of maximum hood, determining which of the various centers is most similar to the input vector. With a given input vector, the output of a simple node will be Y= f(x-c) where, for example, the function can be taken as

$$f = (\mathbf{x} - \mathbf{c}) = \frac{1}{(2\pi)^{n/2}\sigma_1.\sigma_2...\sigma_n} \exp\left\{-\frac{1}{2}\sum_{j=1}^{n}\left(\frac{x_j - c_j}{\sigma_j}\right)^2\right\} \qquad (1)$$

The values σ_1, σ_2…. σ_n, j=[1,n] are used in the same manner as in the normal probability distribution to determine the scalar dispersion in each direction. Another common variation in the Radial Basis Function is to increase its functionality using the Mahalanobis distance in the Gaussian function. The previous equation becomes:

$$f = (\mathbf{x} - \mathbf{c}) = \frac{1}{(2\pi)^{n/2}|K|^{1/2}} \exp\left\{-\frac{1}{2}(x-c)^T K^{-1}(x-c)\right\} \qquad (2)$$

where K^{-1} is the inverse of the X co-variance matrix, associated with the node of the hidden C layer. Given n-vectors (input data) of p-samples, representing p-classes, the network may be initiated with the knowledge of the centers (locations of the samples). If the J-th vector sample is represented, then the weight matrix C can be defined as: $C = [c_1\ c_2\ ...c_3]^T$ so that the weights of the hidden layer in the j node are composed of the center vector. The output layer is a weight sum of the outputs of the hidden layer. When presenting an input vector for the network, the network implements

$$y = W \cdot f(\|x - c\|) \qquad (3)$$

where *f* represents the functional output vector of the hidden layer, and C the corresponding center vector. After supplying some data with desired results, the weights W can be determined using LMS training algorithm interactively and noninteractively, as techniques of the descendant and pseudo inverse gradient, respectively. The learning in the intermediate (hidden) layer is executed using the nonsupervised method, typically a cluster algorithm, a heuristic cluster algorithm, or an algorithm supervised to find the centers (C nodes in the hidden layer). The most common algorithm employed to determine the centers (which are the connections between the input layer and the intermediate layer) is the Lloyd or K-means algorithm. Some studies also have employed supervised learning to find the centers, and self-organized learning of the centers or the minimum Orthogonal Least Squares algorithm. A simple way of determining the σ^2 variation parameter for the Gaussian functions is to make them equal to the median distance between all the training data

$$\sigma_j^2 = \frac{1}{M_j} \sum_{x \in \Theta_j} (x - c)^T (x - c) \qquad (4)$$

where Θ_j is the group of training patterns grouped in the center of the cluster Cj, and Mj is the number of standards in Θ. Another way of choosing the parameter σ^2 is to calculate the distances between the centers in each dimension and use some percentage of this distance for the scale factor. In this manner, the p-nearest neighbor algorithm has been used. The reasons that have led us to study the application and use of the RNA approach for classification are: (1) an RNA has the intrinsic capacity of learning input data and to generalize; (2) the network is nonparametric and makes more delicate suppositions regarding the distribution of input data than the static traditional methods (Baysian); and (3) an RNA is capable of creating decision boundaries which are highly nonlinear in the space of characteristics. Beyond this, these attributes are not unique for the RNAs used for classification.

In summary, in order to conduct the classification from the existing data base, an artificial neural network was used, built from a radial base function (Gaussian), known in literature as RBF with use of the clustering algorithm (k-means) that, for this work, was shown to be very efficient. The architecture of the radial basis function network consists of an entry layer, a hidden layer and an output layer. We believe that the algorithms used are efficient, though we are currently researching the implementation of possibly more efficient algorithms to improve the system.

B. Algorithm Implementation

The following, steps to reach a solution to the proposed problem are described. Step 1: at the first level of connections of a radial base network, one must first of all identify the number of neurons of the hidden layer; Step 2: next the centers c_j are found (j=1,…..,M) which make-up the base of an M-dimensional space. Step 3: for each presented input standard for the network, the $\| x_i - c_j \|$ is sent as a parameter to the radial basis (Gaussian) function which describes the level of classification of the input patterns. The output of the hidden layer makes-up a G matrix, which serves as a basis for the calculation of the weight ***W*** (connections for an output layer), following the formula: $W = G^{\square} t$ where $G^{\square}$ is a pseudo-inverse of the G matrix given by: $G^{\square} = (G^{7\square} G + \lambda G_0)^{\square\square} . G^{7}$ and ***t*** is the matrix which contains the group of training data. In the last step, the output is calculated as a sum of the activated neurons (excited neurons) of a hidden layer.

The K-means and P-nearest neighbor algorithms were used for obtaining the centers and radiuses of each cluster and variance between the centers, respectively. The Gauss function was used for obtaining the output of a hidden layer (centers data, input standards and radii) and a linear function (denoted purelin) contained in the neural Toolbox. This Toolbox is an addendum to the Matlab software [10], where various implemented functions are available for use in the neural network project. The source code has a `.m` extension and executes the classification of users through the K-means, P-nearest neighbor and Gauss algorithms [3, 7]. Initially was used the well-known Copenhagen data (up to 4000 observations classified in 7 clusters). Currently, in news investigations, we are testing with real data from Telefônica Celular (Carrier at South Brazil) containing up to 53000 calls.

C. Results of the Classification

The best classification using this pattern recognition technique was obtained using 110 neurons in the hidden layer, giving an error rate equal to 4,2027% (see Table I).

TABLE I - NUMBER OF NEURONS IN THE HIDDEN LAYER AND RESPECTIVE ERROR.

Neurons of the hidden layer	error rate
50	5.0185
107	4.3758
100	4.4252
110	4.2027
111	4.2027
127	4.3511
137	out of memory
150	out of memory

This number is satisfactory, because the used literature [11], which uses other algorithms with this same data showed similar or greater error rates (5,4 using Back Propagation algorithms, for instance). The algorithms used here have demonstrated reasonable performance.

This classified data make-up the Data Base used in the implementation of the Mobile Phone Security Management System presented as follows (see Section 4). In the implementation, every call is compared with this Data Base in order to identify a possible fraud, i.e., a call which does not match with that the pattern of the client.

4. IMPLEMENTATION

The implementation of the security system for mobile phone usage (which uses the classified data as described above is being conducted by steps, both with CORBA distribution support. The first one using C++ (employing Visigenic VisiBroker IDL Compiler for C++, version 3.0) and the second one using Java JDK1.2 (employing Visibroker 3.1 for Java). The second alternative becomes easier and faster.

The ODP/OMG **CORBA** (Open Distributed Processing/Object Management Group Common Object Request Broker Architecture) is a management technology for distributed objects. It provides a basic structure for distribution and has demonstrated its ability for the support of important functions required in telecommunication and services management networks [9]. The frauds in telecommunication may reach a global scale, therefore, it is necessary to produce a system under an independent code architecture. In addition, it is consider ATM. ATM (Asynchronous Transfer Mode) is a broad band transmission technology that, with the reliability of the modern facilities, provides a rapid package commuter. There is a group of CORBA services which provides distributed management of ATM networks [12]

5. CONCLUSIONS

Safety and Security are two reliability properties of a system. A 'safe' system provides protection against errors of trusted users, while 'secure' system protects against errors introduced by nontrusted users [15]. A comprehensive network security plan must encompass all the elements that make up the network and provide important services: Access (authorized users), Confidentiality, (information remains private), Authentication (sender is who he claims to be), Integrity (message has not been modified in transit) and Nonrepudiation (originator cannot deny that he sent the message) [6].

The present work seeks to contribute towards the reduction of losses or damages, for clients as well as for telecommunication carriers, through the implementation of an anti-fraud system, which avoids the cloning of mobile phones. Brazil is likely to have 6 million mobile phones by December 1998 – in December 1994 there were merely 700,000. In Brasilia, the Federal District will have 320,000, i.e., one for every 6 in habitants (a high density when compared to France or Spain, for instance).

Currently, the frauds attain 2% of the profits of Brazilian telecommunication carriers, that is, from one to one and half million dollars each month. The fraud calls include Sex, traffic, drug and weapon (Colombia and Libia calls, for instance). In Santa Catarina, in May1999, a client received a bill totaling up to 25,000 dollars. Telesp, the São Paulo carrier, for example, in 1997 had already lost ≈ 18 million dollars [5]. It is important to highlight, that even the digital mobile telephones have already been cloned – in April 1998.

The system presented in this work employs a classification algorithm of high reliability, uses efficient technologies and provides an automatic character in the process of contacting the user (to verify the origin of a suspected call, after setting off the alarm of discrepancy between a given call and the associated pattern of the user). This service is similar to the emission of warnings and messages of the automatic wake-up call type.

At this time, we can say that the techniques employed were very useful. The LOTOS ISO standard helps with a rigorous validation process. The automata generation was especially important in revealing all possible execution paths. The C code generated from LOTOS code is being investigated, in order to make the development faster. The CORBA support for the distribution of codes (both C and Java) have been very satisfactory. The method used for the classification of the carrier clients, which includes mainly the Gauss algorithm, proved to be efficient and reliable with the use of the MatLab software.

As a continuation of this study, the intention is to reduce the obtained error rates, by employing the Orthogonal Least Squares algorithm. This is a Gram-Schmidt orthogonalization process, which guarantees that each new column added to the design matrix of a growing subset is orthogonal to all previous columns. This simplifies the process of obtaining the sum-squared-error, which makes the algorithm more efficient. In addition, similarity investigations between the C code generated with CADP tool and C/Java code generated under a CORBA distribution support will be made, in the newest version of Eucalyptus ToolSet.

REFERENCES

[1] E. Brinksma. IS 8807 – *LOTOS – Language of Temporal Ordering Specifications*, 1988.

[2] G. Chen, J Rixon, Q. KONG. *Integration CORBA and Java for ATM Connection Management*. DSOM'97. Sydney, Australia, pp. 104-117, 1997.

[3] R. Duda, P.E. Hart. *Pattern Classification and Scene Analysis*. John Wiley & Sons, 1973.

[4] H. Garavel. *CADP Manual*. INRIA, France, 1996.

[5] D. N. Gonçalves. *Eyes on the Bill – The growing number of victims of the cloned mobile phone*. (in Portuguese) Veja Magazine, p. 86, 08/10/1997.

[6] P. Dowd; J.T. McHenry. "Network Security: It's time to take it Seriously". IEEE Computer Magazine, Vol31, N9, Sept98, pp.24-28.

[7] R. A. Johnson, D. Wichern. *Applied Multivariate Statistical Analysis*, Prentice-Hall, Inc., New Jersey. 1982.

[8] G. McGraw, E. Felten. *Java Security*. Ed. Wiley, 1997.

[9] G. Pavlou. *From Protocol-based to Distributed Object-based Management Architectures*. DSOM 97. Sydney, Australia, pp. 25-40, 1997.

[10] *The Student Edition Of Matlab* – For Ms-DOS Personal Computers: The Problem-Solving Tool for Engineers, Mathematicians, and Scientists. The Math Works Inc., Prentice-Hall, Englewood Cliffs, NJ 08632, 1992.

[11] J. L. Todesco. *Pattern Recognition using Artificial Neuronal Network with a Radial Basis Function: na application for a human chromosome classification*. PhD Thesis. UFSC/PEPS. Florianopolis, 1995.

[12] L. H. Hauw, Z. Canela, F. Voyer. *A CORBA-Based TMN Prototype With Web Access*. DSOM´97, Sydney, Australia, pp. 81-93, 1997.

[13] M. S. M. A. Notare, F. A. S. Cruz, J.B.M. Sobral, J.B.M. Alves, B. G. Riso, C. B. Westphall. *Distributed Management in the Security Area for Cloned Mobile Phones*. IEEE DSOM'98. University of Delaware, Newark, Delaware, USA, 26-28/10/1998.

[14] W. Stallings. *Network and Internetwork Security – Principles and Practice*. IEEE Press. Prentice-Hall. IEEE, ISBN: 0-7803-1107-8, p. 462, 1995.

[15] D.S. Alexander; W.A. Arbaugh; A,D. Keromytis; J.M. Smith. "Safety and Security of Programmable Networks Infrastructures". IEEE Communications Magazine. Vol. 36. N10, Oct98. Pp. 84-92.