A SECURITY SIMULATION MODEL FOR LARGE SCALE DISTRIBUTED SYSTEMS

Ciprian Dobre, Florina Constantin, Florin Pop and Valentin Cristea
Department of Computer Science
University POLITEHNICA of Bucharest
Spl. Independentei, 313, Bucharest
Romania
E-mails: florina.constantin@cti.pub.ro, {ciprian.dobre, florin.pop, valentin.cristea}@cs.pub.ro

KEYWORDS
Security, modeling and simulation, evaluation, large scale distributed systems.

ABSTRACT

Today there is a growing interest for large scale distributed systems, both from academia and the industrial environment. If until recently the research in this area mainly focused on developing operational infrastructures, currently many applications have some additional needs. Among these, security represents a hot research topic. In this paper we present a simulation model suitable for evaluating methods and techniques designed to increase security in large distributed systems. The model has the characteristics needed to develop a wide range of security scenarios, being able to assess from solutions to secure data transfers to various mechanisms to assess the access management in a distributed system. The model was implemented as an extension of the MONARC simulator for distributed systems. We present experimental results demonstrating its capabilities to correctly model security solutions for large scale distributed systems, and to pinpoint likely security problems in the simulated environments.

1. INTRODUCTION

Modeling and simulation were seen for a long time as viable solutions to develop new algorithms and technologies and to enable the enhancement of large-scale distributed systems, where analytical validations are prohibited by the scale of the encountered problems. The use of discrete-event simulators in the design and development of large scale distributed systems is appealing due to their efficiency and scalability.
Together with the extension of the application domains, new requirements have emerged for large scale distributed systems; among these requirements, security is needed by more and more modern distributed applications, not only by the critical ones. Most times the resources of such systems are located in different geographically dispersed administrative domains. The evaluation of such solutions is usually done by implementing them in real-world environments. Such an approach, however, implies costs. Also, it is hard to make general remarks on the validity of a particular solution based on the observations made in a particular study case.
In this paper we present a security model that allows the analysis of security-dependent experiments, where possible problems can occur in any simulated component. The use of modeling and simulation is appealing because it allows a greater flexibility in evaluating security solutions for distributed systems.
The model was implemented as an extension of the MONARC simulator. This extension allows the user to correctly describe security solutions currently used in many real-world distributed environments (such as GSI, PKI, SSL, cryptographic solutions, etc.). In addition, the implementation includes already-available simulated security attacks. It allows the addition of detection mechanisms for such attacks, by providing simulation mechanisms for message encryption or authentication and authorization. The modularity and extensibility also allows the user to easily add new capabilities or components for custom experiments.
The rest of the paper is structured as follows. Section 2 gives a description of the work related to the work presented in this paper. In Section 3 we present the security model. Section 4 presents implementation details of the extension added to the MONARC simulator. In Section 5 we present experimental results that demonstrate the capabilities of the security model. Finally, in Section 6 we give conclusions and present future work.

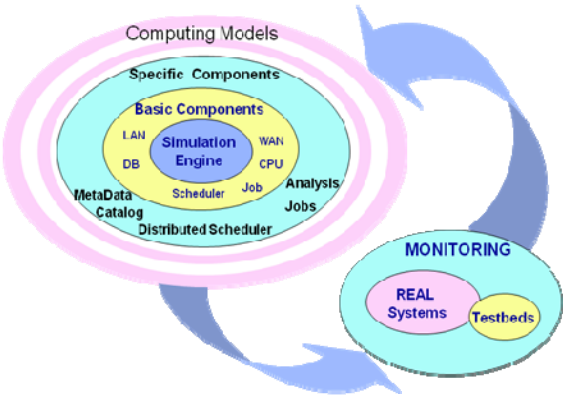
2. RELATED WORK

Currently there are various simulators designed to evaluate solutions for distributed systems (SimGrid, Grids, OptorSim, etc.). In their vast majority, they were all implemented for the modeling of particular problems (such as scheduling, data management, etc.). Because of their narrow areas of usage, many of them do not support the modeling of security functionalities.
The only existing simulator that offers this capability is *G3S* (*Grid Security Services Simulator*), a simulator that can be integrated with GridSim to combine technical investigations to find more efficient allocation of resources with the testing of the security services (Naqvi and Riguidel, 2005). The simulator is currently no longer available and/or supported.
G3S was developed to support various authentication mechanisms, including X.509 certificates and Kerberos tickets. For authorization G3S uses the Roles Based Access Control (RBAC). It also supports the Bell-LaPadula model for ensuring privacy, and the Watermarking technique for ensuring the integrity of data transmitted between the grid's

resources (Naqvi and Riguidel, 2005). It also simulates the privacy feature.
G3S offers various types of patterns of attack, enabling designers to verify if their design may prevent security threats and survive them. In addition, G3S includes a mechanism for spreading security threats notifications. For example, if a node tryies to exceed / violate the privileges defined then an alert about the existence of a malicious node is transmitted to all major nodes (Naqvi and Riguidel, 2005).
In this we also present a simulation model that includes such capabilities for modeling security features, from patterns of attack to intrusion detection, authentication or privacy enforcement solutions. The proposed model also considers a wide-range of security solutions in the general case of large scale distributed systems. The simulation model provided by MONARC is more generic that others, as demonstrated in (Dobre and Cristea, 2007). It is able to describe various actual distributed system technologies, and provides the mechanisms to describe concurrent network traffic, to evaluate different strategies in data replication, and to analyze job scheduling procedures. MONARC offers ample customization possibilities, thus enabling us to integrate our model while preserving the interface. Also, because of this feature, our model can incorporate custom security solutions designed by the user for particular scenarios.

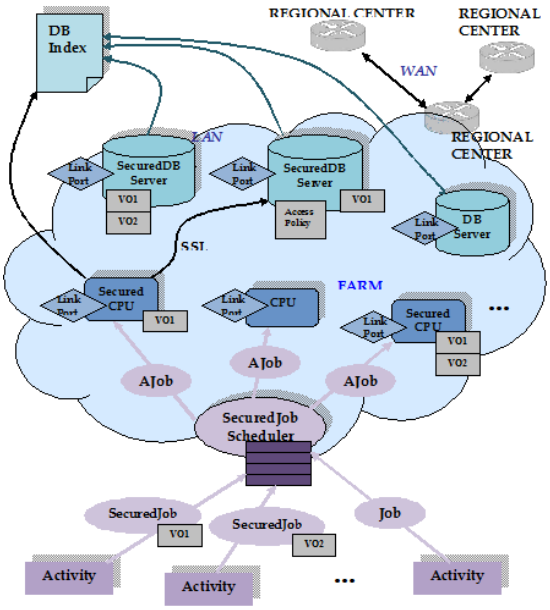
3. A SECURITY MODEL FOR LARGE SCALE DISTRIBUTED SYSTEMS

The proposed model considers the general case of security, as a mean to ensure that systems remain safe and reliable to the errors, threats or malicious changes. The model considers solutions for *data privacy, data integrity* and *system availability*. To ensure such objectives, we consider components designed to protect the services, data and offered information from threats such as *interruption, interception, change* or *forgery*.
The starting point in designing the security model consisted in the specification of security requirements, namely security policy. A security policy describes which actions are allowed and which are prohibited. Entities to which these actions apply include users, services, information, machinery, etc.
Once the security policy is established, the necessary security arrangements for its implementation may be considered. The most important security mechanisms considered are (Johnston, 2004) *confidentiality (*the model includes mechanisms designed to ensure that an authenticated entity can access only the information that has been authorized to*)*, *authentication (*the model includes mechanisms to identify entities involved in a communication or collaboration*)*, a*uthorization (*the model guarantees that once the entity has been authenticated, its options will be restricted / limited to those operations that it is authorized to perform*)*, and a*udit (*the models includes the mechanisms to guarantee the non-repudiation of origin and content of a message*)*.
For the particular case of Grid systems, an additional important concept, also considered by the security model, is the one of Virtual Organization (VO). In a VO different organizations (commercial companies, universities, governmental institutions or laboratories) collaborate to share resources and work together to solve common problems. Each company within a VO is managed independently and has its own security solutions such as Kerberos or PKI infrastructure (Public Key Infrastructure).



Figures 1: The architecture of the MONARC simulator (Dobre, *et al*, 2008)

The model extends the regional center model provided by the MONARC simulator (Dobre and Stratan, 2004). The simulator was chosen based on its capability to allow the modeling of a wide range of distributed systems architectures (Dobre and Cristea, 2007). MONARC provides a realistic simulation environment for modeling large distributed computing systems. The simulator contains the necessary mechanisms for the modeling of competing traffic, for the evaluation of various data replication strategies or the scheduling of task execution.



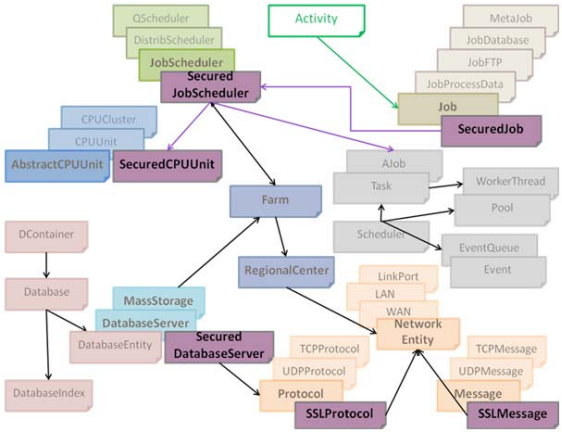
Figures 2: The security model

One of the advantages of MONARC is the ease of expansion and this approach is facilitated by a layered structure (Fig. 1). The first two layers contain the core of simulator (or

"simulation engine") and the basic components of any distributed system (processing units, tasks, databases, networks, the planning, etc.). On top of these are the components designed specifically for certain models of distributed systems. These individual components include different tasks, schedulers with specific algorithms or databases that support data replication. Such simulation components can be easily extended by the user according to requirements of various simulation experiments. Users can also extend the architecture by introducing new components.
The MONARC's model includes several components (Fig. 2). A first set of components is used for describing the physical resources of the distributed system under simulation. The largest one is the regional center, which contains a farm of processing nodes (CPU units), database servers and mass storage units, as well as one or more local and wide area networks. Another set of components model the behaviour of the applications and their interaction with users. Such components are the "Users" or "Activity" objects which are used to generate data processing jobs based on different scenarios. The job is another basic component, simulated with the aid of an active object, and scheduled for execution on a CPU unit by a "Job Scheduler" object.
The security model extends the regional center model. It includes extensions to all existing components of a distributed system (processing units, database server, jobs, job schedulers). We added a new secured job that carry authentication tokens or certificates, and is able to request data based on specific rights. The user can specify the use of X.509 certificate, together with a PKI infrastructure for example, or can easily add new means of authentication.
The model also includes the possibility to define VOs, based on specific security policies shared between regional centers. The model includes the mechanisms to evaluate various authentication solutions. Such authentication mechanisms are applied to the scheduler, processing unit, and even when jobs request data from the database servers.
For example, the job scheduler includes restrictions to where to execute specific jobs, based on the VO to which they belong. The processing units are capable to verify if a particular job is allowed to be executed. The access control verification can be implemented based on various schemas (RBAC, MAC, DAC, etc).
Also, we added delegation, a concept important especially in case of Grid experiments. Thus, a job is capable to delegate his rights (in the form of a certificate for example) to all the jobs that he further instantiates. The database server supports various security policies and includes mechanisms to verify authentication of incoming connections, and to support secured transport protocols. It also includes various mechanisms to certify the validity of data access requests. Similar to the processing unit, the database server maintains a list of VOs to which it belongs and the access policies for the data shared by the server with the other resources within the VO. For example, when a job accesses data on the server the model specifies a series of mechanisms to verify the identity of the requester (by default the model uses validation based on a PKI infrastructure and X.509 certificates) and if the job presents sufficient access rights. For compatibility with other experiments, a regional center can include both secured and non-secured database servers. The same applies for the processing units.
Within the networking simulation model, the model adds the possibility to include secured data transport protocols. For example, we included the SSL protocol to offer the possibility to encrypt the messages being exchanged between entities in a simulation experiment. We added the possibility to implement various handshake mechanisms (for protocols supporting authentication capabilities). The user can easily add and evaluate new protocols and mechanisms. The model includes mechanisms for data encryption, keys and certificate management, etc. In addition, it includes mechanisms for traffic filtering by specifying exclusion rules based on various metrics (ports, addresses, protocols, etc) and corresponding actions (reject for example).

4. IMPLEMENTATION DETAILS

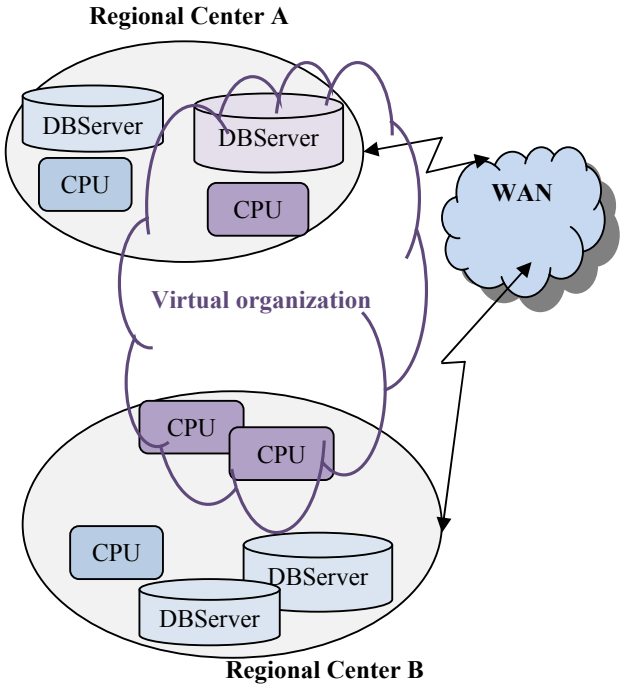
Figure 3 presents the main classes added to the MONARC simulator. The implementation allows the possibility of *defining virtual organizations* in which users can share resources from various regional centers, such as computing power or databases, in a manner as safe as possible. Also, it allows *defining security polices* for each of the components belonging to a virtual organization, policies through which the proprietary organization is able to impose restrictions on the access to its shared components. These restrictions may include restrictions on the execution of jobs on workstations or access restrictions to the data on certain database servers.
Based on the security module of Globus Toolkit (The Globus security Team, 2005) and the GSI (The Globus security Team, 2010), we added within the security model the possibility to specify *authentication* mechanisms using *X.509 certificates*. The X.509 certificate is used for authentication to various Grid entities. Users belonging to a VO present such a certificate that 'vows' for their identity when submitting a job for execution or when communicating with other system entities.



Figures 3: The classes added in MONARC

The security policies in VO shape the access *authorization* mechanism to the shared resources within them. For instance, if a shared workstation belonging to a VO has an associated security policy that imposes a limit on the amount of memory that a job submitted can use then any job that

violates this restriction is ignored. Authentication and authorization ensure the system from unauthorized access to its resources.
The security implementation also enables the *protection of message content* sent throughout the network against attacks such as interception (eavesdropping), and thus keeping its confidentiality, by encrypting its content. It also ensures *secure data transfers* by proposing the use of various protocols that allow the authentication of the parties involved in the communication (such as the SSL protocol). This ensures both the integrity of messages transmitted, and their protection against attacks such as man in the middle.
The implementation also includes an exclusion rule based *traffic filtering* of all components of a virtual organization. This mechanism can be used to prevent attacks such as DoS. In case of many connections coming from the same address, for example, the filtering policy can specify that that particular address is banned for a certain period of time (or permanently).



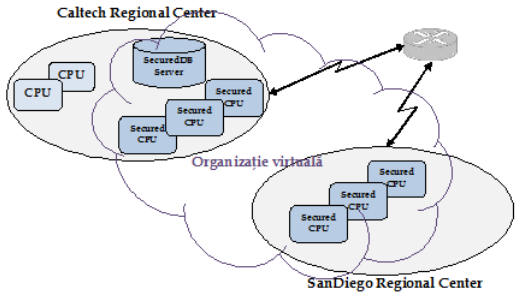
Figures 4: A Virtual Organization example

In the implementation we extended all basic components: the processing unit, the job, the database and the job scheduler by introducing an authentication mechanism, access control mechanisms, or the possibility to schedule the jobs according to the restrictions imposed by the virtual organization in which they are executed. For the jobs being executed the processing units include various mechanisms for specifying their execution rights. Also, the data can specify different protection rights and mechanisms.
A component that models the specific behavior of a SSL protocol was added in an effort to simulate the authentication of the components involved in the message transfer and to ensure the confidentiality of the message transmitted through the network by using cryptography.
In addition we added a component to simulate traffic filtering based not only on static rules defined by users in the configuration file, but also dynamic ones, added during the simulation.
The simulation model had the important goal to preserve the original extensibility capacity offered by the simulator. Thus, the user can use the proposed security technologies simulation model for evaluating the characteristics of a new cryptographic protocol for example, or for testing the performance of a new tool for non-repudiation, or to ensure the integrity of users accessing applications running in a distributed environment. The user has the possibility to not only extend the job scheduling classes, but also the components that implement the security features (such as the processing unit, protocols and entities involved in the transfer of data or databases). He can even add additional functionalities.
As previously stated, one objective was to enable the definition of VOs that would allow resource sharing and collaboration between various different organizations (in this case between different regional centers). Thus a virtual organization can gather many resources in the system (e.g., processing units, servers, databases, etc.). In turn shared resources within a virtual organization may belong to different regional centers (Fig. 4).
The security policy implementation is based on several assumptions. For example, a VO defines at least one associated certificate. A job must also present the certificate that validates its identity and the name of the virtual organization in which it has to be processed in order to be executed. A user can delegate its certificate attesting its identity to all of the tasks he submits in the system. Every component of a VO can set restrictions on access to it. Any operation of the system must comply with imposed restrictions on access or use. The operations between two entities in the system requires the authentication of at least one of them, and by default any data transfer between entities requires the authentication of the components involved in the transfer. And finally, any component of the system reserves the right to filter out messages from unwanted sources.
The certificates associated with a virtual organization are designed to model the certification authorities (CA). These certificates must be X.509 certificates. Each system user or submitted jobs are associated with certificates signed by a trusted authority of the VO. In order to trust the certificate of an entity a mechanism must be defined that validates the trust in the CA that signed the certificate. In the implementation of the security model a factory object was defined to hold the multitude of certification authorities of the virtual organizations. Thus to verify and validate a certificate of another entity it is sufficient to interrogate the existing certificates from the factory. This will also contain all the trusted authorities of the defined VOs.

5. EXPERIMENTAL RESULTS

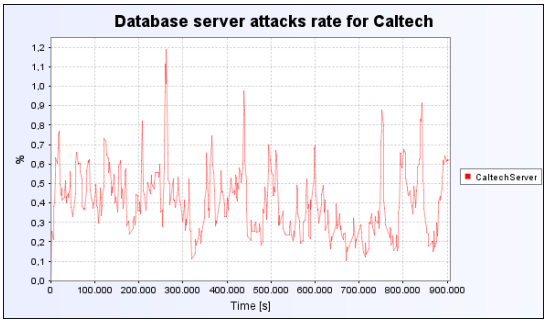
The evaluation of the proposed simulation model consisted of a series of simulation experiments. They were designed to test the capability of the model to correctly model security threats, as well as security solutions designed as

countermeasures (such as security policies, or various components for detecting security problems and react). The experiments also evaluated if the model is easily extendable and if it supports various scenarios. We were interested in possible performance degradation caused by its use in various scenarios.
One such simulation experiment (Fig. 5) evaluates the possibility of simulating an access policy enforcement mechanisms acting for all requests made to a database server. We were particularly interested if the simulation model allows the interpretation of security breaches in such a scenario. In particular, the experiment consists of two regional centers sharing several workstations and one database server.



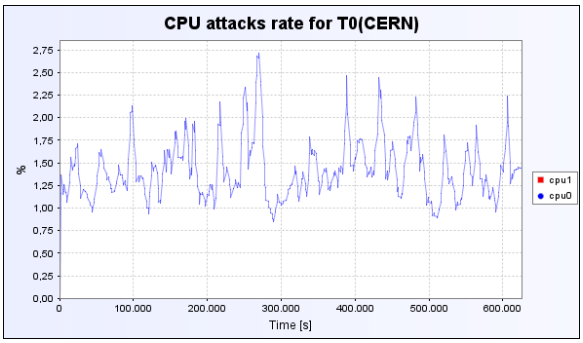
Figures 5: Simulation scenario

For this experiment we defined two custom jobs working with the shared database. One job creates and writes data to the database. The other connects and requests the data matching a specific pattern. We added a security policy on the database server, similar to a UNIX file system. For reading data the job (or the VO generating the job) must have read rights, for writing data it must have a corresponding right, and for removing data the job must have both read and write rights.



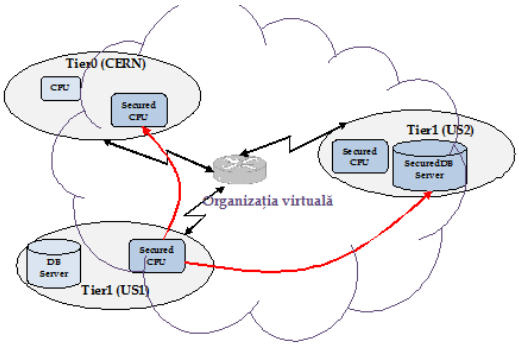
Figures 6: The percent of attacks recognized on the database side from the total number of generated requests

By extending the security model, we were able to concurrently simulate both ordinary jobs, as well as ones that tried different operations on the database without having sufficient rights. We logged and compared how many attacks were randomly generated (reads without the read right, etc.) versus how many attacks did the database server successfully recognized (Fig. 6).
Another scenario consisted of three regional centers sharing together, in the same VO, several workstations (Fig. 7). The experiments evaluated the authentication mechanism used for when jobs are submitted. We defined two types of jobs, with and without a valid certificate used for authentication.
Again, the addition of the authentication mechanism to the simulation model was done easily because of its extensibility. We evaluated and compared the number of generated non-valid authentication attempts versus the number of successfully recognized attempts to authenticate with such certificates (Fig. 7).



Figures 7: The attack rate

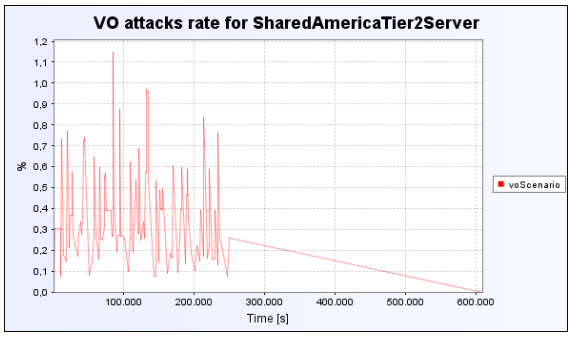
Next we evaluated the possibility to add a data filtering mechanism to the experiment. We defined a filtering rule for both a workstation within the virtual organization, as well as for the database server. In the scenario the virtual organization shares the resources belonging to three regional centers. The filtering mechanism is responsible with verifying all data received from one of these regional centers (Fig. 8).



Figures 8: The configuration used for the filtering experiment

For this experiment we added an extra database server within the regional center T1-US2. In case of this database server and the workstation shared by T0-CERN we defined filtering rules for all messages received from the network belonging to T1-US1. The experiment considered both data traffic between workstations belonging to different regional centers, as well as traffic generated towards the database server (reading and writing data into the database).
In the experiment the jobs in T0-CERN are waiting data or send requests to the database server. The jobs running in T1-US2 send data to T0-CERN, and the ones running in T1-US1 send messages to T0-CERN and requests to the database server in T1-US2. The data messages and database requests sent from T1-US1 are filtered because they come from addresses outside the filtering rule and are reported as

security problems. Figure 9 presents the attack rate on the database server in T1-US2.



Figures 9: The attack rate over the database server at VO level

In all these cases not only the security solutions designed and included in the proposed security model correctly handled possible attacks, but also the performance of the distributed simulated environment (throughput in the network or processing capability of the simulated processing units) was not affected beyond rendering the environment to be used anymore.

6. CONCLUSIONS

Large scale distributed systems are currently progressing from operational infrastructures to environments providing many "modern" capabilities. Security in large scale distributed systems represents an important research subject in this area. There are many solutions for enforcing security policies, or establish well-defined administrative domains between distributed organizations, secure communication taking place between distributed resources, etc. Validation of such security solutions is generally accomplished using real-world implementations.

Simulation is an attractive alternative to evaluating such solutions. Unfortunately, even though there are several simulators designed for distributed systems, with few exceptions they do not present solutions that can be used for the evaluation of security methods and techniques.

In this paper we proposed a simulation model suitable for evaluating methods and techniques designed to increase security in large distributed systems. As presented, this model has the characteristics needed to develop a wide range of security scenarios, being able to assess from solutions to secure data transfers to various mechanisms to assess the access management in a distributed system.

We presented implementation details of an extension of the MONARC simulator for distributed systems. The proposed components and mechanisms allow the evaluation of a wide range of security protocols and solutions, in the context of various distributed architectures. The implementations allow the evaluation of secure communication protocols, of mechanisms for authentication, of VOs, etc.

We also presented experimental results demonstrating the capability to correctly model security solutions for large scale distributed systems, and the capability of the model to pinpoint likely security problems in the simulated environments.

In the future we plan to extend the simulation model with the support for other authentication mechanisms (such as Kerberos tickets for example), include additional patterns of attack, and experiment with more security scenarios to further evaluate the generality of the model.

.

ACKNOWLEDGMENTS

The research presented in this paper is supported by national project "DEPSYS – Models and Techniques for ensuring reliability, safety, availability and security of Large Scale Distributes Systems", Project "CNCSIS-IDEI" ID: 1710, and by national project "TRANSYS – Models and Techniques for Traffic Optimizing in Urban Environments", Project "CNCSIS-PD" ID: 238.

REFERENCES

Dobre, C, and C. Stratan. 2004. "MONARC Simulation Framework", in *Proc. of the 3rd Edition of RoEduNet International Conference*, Timisoara, Romania.

Dobre, C, and V. Cristea. 2007. "A Simulation Model for Large Scale Distributed Systems", in *Proc. of the 4th International Conference on Innovations in Information Technology*, Dubai, United Arab Emirates.

Dobre, C., F. Pop, and V. Cristea. 2008. "A Simulation Framework for Dependable Distributed Systems", *First International Workshop on Simulation and Modelling in Emergent Computational Systems* (SMECS-2008), Portland, USA.

Johnston, S. 2004. "*Modeling security concerns in service-oriented architectures*", Accessed on 28.06.2010, from: http://www.ibm.com/developerworks/rational/library/4860.html , published 2004.

Naqvi, S., and M. Riguidel. 2005. "*Grid Security Services Simulator (G3S) – A Simulation Tool for the Design and Analysis of Grid Security Solutions*", Ecole Nationale Supérieur des Télécommunications, France.

The Globus security Team, 2010. "*Overview of the Grid Security Infrastructure*", Accessed on 18.06.2010, from http://www.globus.org/security/overview.html.

The Globus security Team. 2005. "*Globus Toolkit Version 4 Grid Security Infrastructure: A Standards Perspective*", Accessed on 17.06.2010, from http://www.globus.org/toolkit/docs/4.0/security/GT4-GSI-Overview.pdf, published 2005.

BIOGRAPHY

CIPRIAN DOBRE PhD, is lecturer with the Computer Science and Engineering Department of the University Politehnica of Bucharest. The main fields of expertise are Grid Computing, Monitoring and Control of Distributed Systems, Modeling and Simulation, Advanced Networking Architectures, Parallel and Distributed Algorithms. Ciprian Dobre is a member of the RoGRID (Romanian GRID) consortium and is involved in a number of national projects (CNCSIS, GridMOSI, MedioGRID, PEGAF) and international projects (MonALISA, MONARC, VINCI, VNSim, EGEE, SEE-GRID, EU-NCIT). His research activities were awarded with the Innovations in Networking Award for Experimental Applications in 2008 by the Corporation for Education Network Initiatives (CENIC).