Computing and Informatics, Vol. 26, 2007, 89–101

OVERHEAD VERIFICATION FOR CRYPTOGRAPHICALLY SECURED TRANSMISSION ON THE GRID*

Wojciech Rząsa

Rzeszów University of Technology
Wincentego Pola 2, 35-959 Rzeszów, Poland
e-mail: wrzasa@prz-rzeszow.pl

Marian Bubak

Institute of Computer Science, AGH
al. Mickiewicza 30, 30-059 Kraków, Poland, &
Academic Computer Centre CYFRONET AGH
Nawojki 11, 30-950 Kraków, Poland
e-mail: bubak@agh.edu.pl

Bartosz Baliś

Institute of Computer Science, AGH
al. Mickiewicza 30, 30-059 Kraków, Poland
e-mail: balis@agh.edu.pl

Tomasz Szepieniec

Academic Computer Centre CYFRONET AGH
Nawojki 11, 30-950 Kraków, Poland
e-mail: t.szepieniec@cyf-kr.edu.pl

Manuscript received 22 February 2005; revised 29 June 2006
Communicated by Jesús Marco

* This work was partly funded by the EC, project IST-2001-32243, CrossGrid [7]

Abstract. It is well known that the network protocols frequently used in Internet and Local Area Networks do not ensure the security level required for current distributed applications. This is even more crucial for the Grid environment. Therefore asymmetric cryptography algorithms have been applied in order to secure information transmitted over the network. The security level enforced by means of the algorithms is found sufficient, however it introduces additional transmission overhead. In this paper we describe experiments performed in order to evaluate transmission efficiency depending on the security level applied.

Keywords: Grid, security, GSI, application monitoring, OCM-G

1 INTRODUCTION

Nowadays, the Grid can be considered a new technology for sharing diverse resources [10] – computing capability, disk capacity, data etc. The distributed nature of the Grid assumes neither the existence of any centralized control point nor existing relationships or trust between entities sharing their resources. However, the sharing should necessarily be carefully controlled. Resource providers and consumers should be able to clearly define what resources are shared, under what conditions, who is allowed to share resources, and who can use them. In order to make coordination possible *Virtual Organizations* (VOs) have been defined as *a set of individuals and/or institutions defined by sharing rules* [10].

Communication on the Grid is performed by means of broadly available Internet infrastructure, therefore security of transmitted information is an inherent issue. In order to ensure the required security level protocols based on cryptography have been applied. Thus, transmitted data are secured but communication efficiency decreases. The aim of the research presented in this paper is to estimate the overhead resulting from the use of cryptographic enforcement of security of transmitted information depending on different security levels.

2 THREATS TO THE TRANSMISSION AND SECURITY SOLUTION FOR THE GRID

The distributed nature of the Grid which uses public links to communicate is the reason for which security of the Grid infrastructure is a complex issue. In order to make resource sharing coordinated and secure, trust relationships between resource providers and consumers are required. The relationships are defined by *Virtual Organizations*; however, entities require means to verify the identity and VO membership of each other. Despite this security requirement users should be able to authenticate just once when "entering" the Grid and thereafter should be able to access all the resources they are authorized to use. Data transmitted via the Grid infrastructure should be secured in order to avoid alteration or eavesdropping. Authenticity, in-

tegrity and confidentiality of information transmitted between the entities should also be guaranteed.

The most important security requirements of the Grid infrastructure can be divided into the following aspects:

- Authentication: the peers of the connection should be identified upon connection establishment.
- Authenticity and integrity of transmitted information should be guaranteed in order to avoid accidental or deliberated alteration.
- Confidentiality of transmitted data should be guaranteed to prevent eavesdropping.
- Single sign-on should be enabled in order to facilitate resource exploitation.

Vulnerabilities of protocols widely used on the Internet or on Local Area Networks are widely known. The primary threats related to network transmission are briefly described below.

Sniffing (or eavesdropping) is possible in some low-level communication protocols. It is a significant threat to confidentiality of the transmission.

Spoofing is possible for each protocol commonly used on the Internet. Depending on the protocol, it allows impersonation of a host or gives an attacker the capacity to deceive authentication methods based on the source address of the packets or even allows a third host to become an agent between two other hosts, and to fully control connections. Different varieties of spoofing threaten all aspects of secure data transmission.

Session take over (or session hijacking) allows an attacker to steal an already established TCP/IP session. Since authentication is usually performed only on initialization of a connection, this is a significant threat to authenticity of the transmitted data [4, 6].

One may conclude that standard network protocols cannot provide the security level mentioned above. Therefore, cryptography algorithms [13] are applied in order to ensure the desired level of security. Secure connections are enabled by another protocol layer created over the standard network protocols. For secure connection establishment asymmetric cryptography (Figure 1) algorithms are used by entities to verify the identity of the peer. Digital certificates are the basis of the authentication process. Thereafter asymmetric cryptography is used to establish symmetric *session key(s)* for the connection. Data transmitted over the secure connection are encrypted with the symmetric *session key(s)*. Asymmetric keys are used to ensure authenticity and integrity of transmitted information by the use of *digital signatures*. The latter are computed as the hash of data being sign encrypted with the private key of the sender. Both data and signature are transmitted to the destination. The receiver decrypts the signature with the public key of the sender and verifies if the

hash matches the received information (see Figure 2). Nowadays, asymmetric cryptography that is used e.g. by TLS [8] secures a great deal of information transmitted over Internet links.

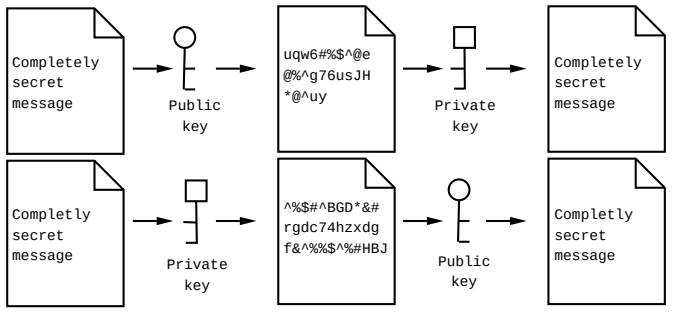

Fig. 1. Encryption and decryption with asymmetric cryptography

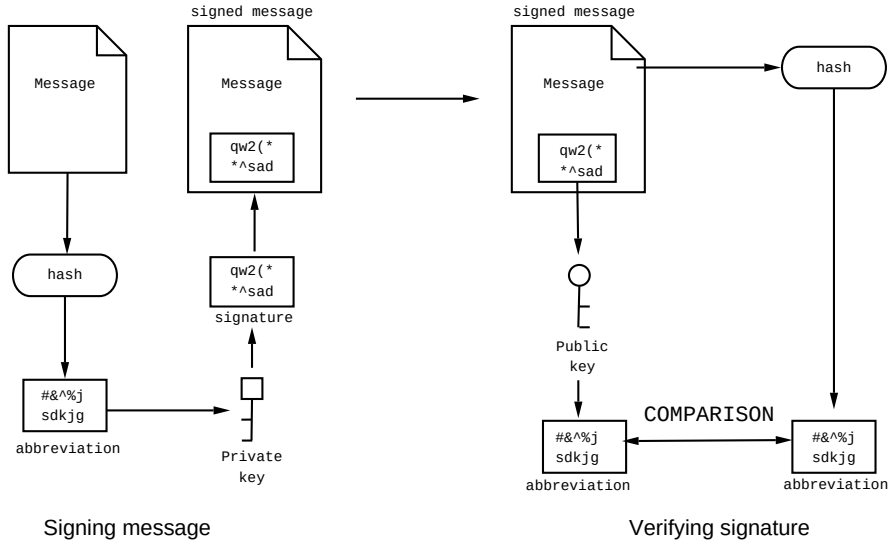

Fig. 2. Creation and verification of a digital signature

Security for Grid applications is also achieved by the use of cryptography. However, the issues that we encounter in Grid environments are significantly more complex; therefore they are not addressed by any standard solution. Most of the existing Grid security solutions have roots in the Grid Security Infrastructure (GSI) [9], which is based on public key cryptography. In order to better define Grid security requirements and find appropriate solutions, the *Global Grid Forum Grid Security Working Group* [12] was formed. Since there are various existing concepts and well tested security solutions addressing different security issues, the GSI WG decided

to develop a security solution for Grid systems on the basis of existing ones. Thus, cryptography is used to ensure security in the Grid environment.

The GSI solution based on cryptography responds to Grid security requirements. However, communication efficiency is decreased by the secured connections layer. Cryptography algorithms require additional computations, integrity and authenticity protection assumes transmission of additional data over the network. Therefore securing transmitted data strongly affects efficiency of the transmission.

Asymmetric cryptography algorithms are significantly more CPU-intensive then symmetric ones due to the more complex mathematical operations and longer keys required. However, they solve the problem of encryption key distribution that is crucial for secure communication protocols exploited by GSI. Therefore asymmetric keys are used for secure connection establishment, but whenever possible, they are substituted by a symmetric *session key(s)*. Digital signatures are computed using the hash of transmitted information in order to minimize overhead introduced by the use of asymmetric cryptography algorithms. Thus, more efficient algorithms are used to transmit data over the secure connection.

Data transmission secured by means of cryptography algorithms used by GSI introduces additional overhead. However, we should notice that connection establishment is significantly more resource-consuming than transmission itself. These overheads have a direct impact on the efficiency of the tools used on the Grid. As an example let us consider the OCM-G – CrossGrid monitoring system for interactive applications, which is intended to facilitate the development of complex Grid applications [5].

The OCM-G is designed as an on-line monitoring system capable of monitoring and manipulating application processes in real time. It is an infrastructure that is shared between numerous users and applications.

It seems obvious that security is an important issue for the Monitoring System. Security weakness of its infrastructure would strongly affect the security of shared resources and computations. However, efficiency of the monitoring infrastructure created by the OCM-G is no less important. The latencies between the user and his application processes should be appropriate for on-line monitoring. GSI was chosen to secure the monitoring system since it fits the Grid environment and provides a means to match the security requirements of OCM-G.

We are aware of the dependency between security and efficiency of the transmission, and therefore we decided to perform experiments in order to evaluate the overhead resulting from using GSI. The aim is to determine the communication efficiency attainable at a required security level.

3 EXPERIMENTS

In this section we describe three experiments performed in order to evaluate resource consumption and other undesirable effects of cryptographic algorithms at different stages of a connection.

The security requirements of Grid applications may vary. Therefore we performed tests with the following security levels:

Level	Description
CLEAR	no security mechanisms were enabled,
AUTH	authentication and authorization were performed upon connection establishment,
PROTECT	authenticity and integrity of transmitted data via digital signatures were guaranteed,
ENCRYPT	transmitted data were encrypted.

Note that stronger security levels includes all aspects of weaker levels.

We found it convenient to use two different execution environments for two circumstances. Experiments concerning transmission overhead were carried out on slower machines, where it was easier to observe the CPU time usage. DoS vulnerability tests required a large cluster of machines, while the high computing capability of particular nodes did not make the interpretation of the results difficult.

All test programs were implemented using `Globus_IO` – a communication library that is part of the *Globus Toolkit 2* [11]. `Globus_IO` implements the GSI security solutions and is designed specifically for Grid systems.

3.1 Transmission Overhead

The first experiment we performed in order to estimate the transmission overhead was related to secure data transfer. The aim of this test was to evaluate the overhead resulting from the security mechanisms applied, depending on the security level used during data transmission.

In order to perform the experiment, two short programs were written: `sender` and `responder`. `Responder` was designed to receive data and to send it back. `Sender` was designed to send data to the `responder` and to receive it back.

All measurements were performed on the sender side. Each measurement was started before connection was established and stopped when all data were received; thus all results concern two transmissions: from `sender` to `responder` and from `responder` to `sender`. For the measurements that required transmission of multiple data packets, the connection was established once. Thereafter multiple transmissions were performed through this connection.

The experiment was performed on a PC cluster. The `sender` process was executed on an Intel Celeron 300 MHz machine, the `responder` on an Intel PIII 600 MHz machine. The hosts were connected with 100 Mbps switched LAN.

The measurement was conducted by transmitting data through the network between `sender` and `responder`. `Sender` measured the CPU time for the two-way transmission. The measurement was carried out for different quantities of 100-byte packets with different security levels applied. The results are presented in Figure 3.

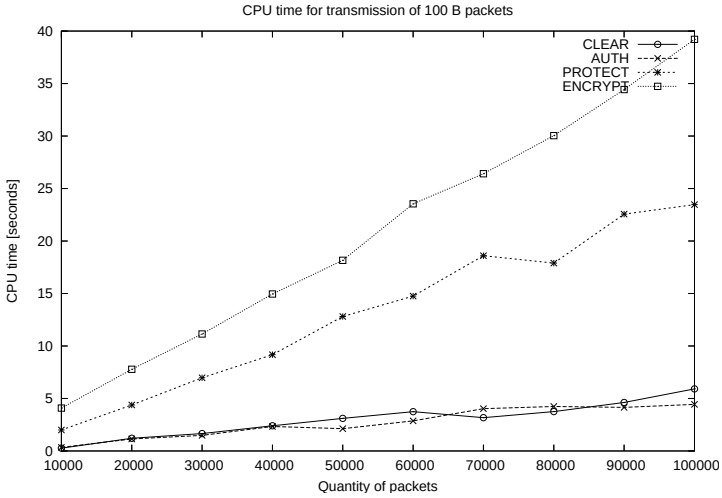

Fig. 3. Results of the security overhead test

We may notice a linear relationship between CPU time and quantity of packets for each security level; however, for higher security levels the CPU time increases faster. In order to estimate the overhead resulting from the proposed solution, we present an average CPU time for the discussed security levels (see Table 1).

Security level	Avg. CPU time [ms]
CLEAR	0.0530
AUTH	0.0448
PROTECT	0.2357
ENCRYPT	0.3826

Table 1. Average CPU time for 100-byte packet

3.2 Transmission Time and Packet Size

The previous experiment was performed for fixed-size packets. Subsequently, we studied the dependency between transmission time and size of the transmitted packets. We should expect it is more efficient to transmit data in large packets.

For this experiment we used the `sender` and the `responder` processes configured as before. We performed two-way transmission of 10,000 packets of different sizes, and measured CPU time consumption. In order to compare results, we computed the average CPU time required to transmit 100 bytes for each size of the packet. The results of the experiment are presented in Figure 4.

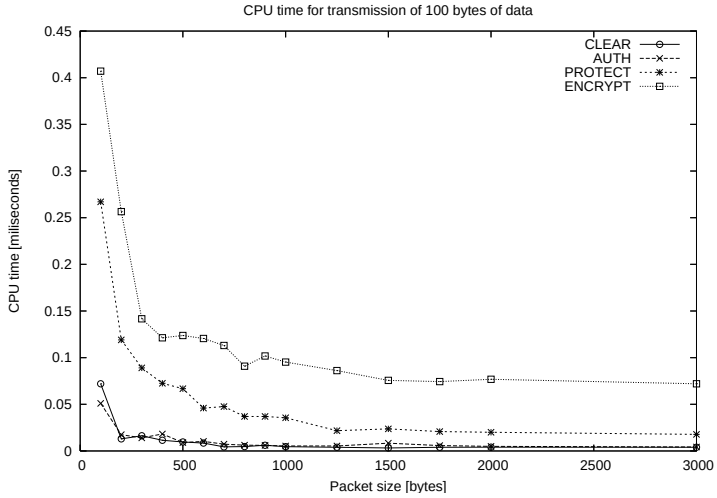

Fig. 4. Size of the packets and CPU time consumption for different security levels

3.3 Susceptibility to Denial of Service Attacks

As mentioned before, establishment of a secure connection consumes more resources due to the complex asymmetric-cryptography operations performed. Thus we can conclude that servers may not be able to simultaneously establish numerous secure connections. This will also make the port listening for secure connections more vulnerable to *Denial of Service* attacks.

The aim of the experiment described below is to estimate the vulnerability of the secure connections server to DoS attacks. We try to find the practical limit of secure connections that can be handled by the server, to verify the results of server overload and to estimate the number of connections that can be safely handled.

In order to perform the experiment we implemented two short programs: `client` and `server`. `Server` listened for connections, and after a connection request arrived, it logged the time of the arrival. Then it tried to establish a connection and logged the time of establishment or failure. `Client` first waited until a specified clock time and tried to establish connection with the specified host on a specified port. The start time and connection establishment or failure time were logged. In this scenario, the `client` was the intended attacker while the `server` was the "victim" of this attack.

The experiment was performed on a PC cluster with machines equipped with two 2.4 GHz Intel Xeon processors and 1 GB RAM each. Eighteen machines connected with 100 Mbps switched LAN were involved in the test.

The experiment consisted of a simulated *Denial of Service* attack that was performed by `clients` to `server`. Seventeen hosts requested connections to the server, which ran on the remaining host. The clocks of the hosts were synchronized with

the NTP protocol [14], thus we were able to achieve the requisite accuracy while triggering `clients`.

During the experiment we increased the number of `client` processes on each node until we reached the maximum number of incoming connections per second that could be handled by `server`. The experiment was performed for the connections with different security levels applied. We present the maximum number of connections that were requested in one second and properly established (Table 2). We have observed that in the case of connection failure `client` returned a *Connection timed out* error.

Security level	Connections				
	Requested		Established		Failed
	Overall	In 1 second	Overall	In 1 second	
AUTH	901	**899**	**881**	30	20
PROTECT	901	**894**	**871**	30	30
ENCRYPT	901	**896**	**897**	30	4

Table 2. Results of the simulated DoS attack

Security level	Connections				
	Requested		Established		Failed
	Overall	In 1 second	Overall	In 1 second	
CLEAR	1700	**1692**	**1700**	1691	0

Table 3. TCP connections established in one second

We did not attempt to establish results for a DoS attack for *clear text* transmissions since a successful DoS attack using the raw TCP protocol may strongly affect the server availability. We do present results that show that *clear text* connection establishment is significantly more efficient than the one required for secure transmission (Table 3).

4 CONCLUSION

The first experiment showed that the average CPU time required to perform data transmission with integrity checking is over four times greater than the time required for a *clear text* transmission. Transferring data via encrypted connections requires even more resources: the average CPU time for this connection is over seven times greater than for *clear text* ones. However, we should realize that the CPU time required to transmit 100 bytes through the most resource consuming connection is on the order of 1/10 millisecond even though we did not perform the experiment on particularly fast hardware. Thus, even the connection that requires the most computing resources should not cause significant overhead, neither in CPU time consumption nor in the delay of message delivery.

While designing the security policy of a system we should consider which security level is really required for data transfer. From the results of the first test we can see that the differences in CPU time consumption between particular levels of security are significant. Therefore, it is desirable to restrict the security level only to that which is really required by the system.

Considering the results of the second experiment we may conclude that the size of the transmitted packet is significant for transmission efficiency. The average CPU time required to transmit 100 bytes significantly decreases with packet size increasing to the size of 1.5 kilobytes. Thus, as expected, it is more efficient to transmit a small number of large packets than a huge number of small packets. This is also true for raw TCP connections, however, this factor seems to be more significant for secure ones.

The establishment of a secure connection is an exceptionally resource-consuming process. Therefore, servers can handle fewer simultaneously incoming secured connections than raw TCP ones. Also, it is important to notice that a client that cannot establish a connection with an overloaded server receives the *Connection timed out* network error. Thus it is possible to deduce the reason for which the connection could not be established.

5 SUMMARY AND FUTURE WORK

In this paper, we have shown the results of three experiments which tested the efficiency of different aspects of connection secured with cryptographic algorithms. We have evaluated the overhead resulting from secure data transfer as well as connection establishment.

It can be seen from the presented tests that applying security mechanisms to secure network connections results in significant overhead. The increase of CPU time required to transmit protected information in comparison to clear data transfer, as well as increased vulnerability of secured connections to a DoS attack cannot be overlooked. Security offered by asymmetric cryptography algorithms results in significant resource consumption.

For the reasons described above, the security mechanisms applied to data transmission in a system should always match the real security requirements. It seems advantageous to introduce an optional lower security level for less vulnerable network connections whenever permitted by the system security policy, especially when we desire highly efficient data transmission.

The experiments we have performed prove that the security overhead is acceptable for the OCM-G monitoring system. However, we realize that they do not provide complete information concerning the efficiency of GSI-secured network communication. Depending on requirements of particular Grid systems their developers may require results of specific experiments conducted in specific circumstances. Possible tests may concern e.g. details about wall time overhead during authentication, encryption and transmission or network overhead resulting from the necessity of

sending additional data required by security protocols. Therefore we found it advantageous to work out an analytic method enabling estimation of security overhead on the basis of a model of the considered distributed system. Thus, developers would be able to verify the efficiency of their applications without performing complex experiments.

Acknowledgments

Authors are grateful to the anonymous reviewers for their valuable suggestions and to Mr. Piotr Nowakowski for his remarks.

REFERENCES

[1] Baliś, B.—Bubak, M.—Rząsa, W.—Szepieniec, T.: Efficiency of the GSI Secured Network Transmission. Proc. of Internationnal Conference on Computational Science, Kraków, Poland, June 2004, LNCS 3036, pp. 107–115, 2004.

[2] Baliś, B.—Bubak, M.—Funika, W.—Szepieniec, T.—Wismuüller, R.: An Infrastructure for Grid Application Monitoring. In: D. Kranzlmueller, P. Kacsuk, J. Dongarra, J. Volker (Eds.): Recent Advances in Parallel Virtual Machine and Message Passing Interface, Proc. 9th European PVM/MPI Users' Group Meeting, Linz, Austria, 2002, LNCS 2474, pp. 41–49.

[3] Baliś, B.—Bubak, M.—Rząsa, W.—Szepieniec, T.—Wismüller, R.: Security in the OCM-G Grid Application Monitoring System. Proc. of 5th International Conference on Parallel Processing and Applied Mathematics, September 2003, LNCS 3019 pp. 779–787, 2004.

[4] Bellovin, S.: Security Problems in the TCP/IP Protocol Suite. Computer Communication Review 19, Vol. 2, 1989, pp. 32–48, `http://www.research.att.com/~smb/papers/ipext.ps`.

[5] Bubak, M.—Funika, W.—Wismüller, R.: The CrossGrid Performance Analysis Tool for Interactive Grid Applications. In: D. Kranzlmueller, P. Kacsuk, J. Dongarra, J. Volker (Eds.): Recent Advances in Parallel Virtual Machine and Message Passing Interface, Proc. 9th European PVM/MPI Users' Group Meeting, Linz, Austria, 2002, LNCS 2474, pp. 50–60.

[6] Bellovin S.: Defending Against Sequence Number Attacks. RFC 1948.

[7] The CrossGrid Project. `http://www.eu-crossgrid.org`.

[8] Dierks, T.—Allen, C.: The TLS Protocol Version 1.0. RFC 2246.

[9] Foster, I.—Kesselman, C.—Tsudlik, G.—Tuecke S.: A Security Architecture for Computational Grids. To appear in the 5th ACM Conference on Computer and Communication Security.

[10] Foster, I.—Kesselman, C.—Tuecke, S.: The Anatomy of the Grid. International Journal of Supercomputer Applications, Vol. 15, 2001, No. 3.

[11] Globus Alliance homepage: `http://www.globus.org`.

[12] GGF GSI Working Group: `http://www.ggf.org/security/gsi/index.htm`.

[13] Menezes, A.—van Oorschot, P.—Vanstone, S.: Handbook of Applied Cryptography. CRC Press, 1996, http://www.cacr.math.uwaterloo.ca/hac/.

[14] Network Time Protocol project homepage http://www.ntp.org/.

Wojciech Rząsa obtained his M. Sc. degree in 2003 from the AGH University of Science and Technology, Krakow, Poland. He works on his Ph. D. as a research and teaching assistant at the Division of Informatics and Control of the Rzeszow University of Technology, Poland. His research concentrates on security and efficiency of distributed applications, in particular the applications designed for the grid environment. Co-author of two scientific papers in Springer Lecture Notes in Computer Science and two in conference proceedings.

Marian Bubak obtained M. Sc. degree in technical physics and Ph. D. in computer science from the AGH University of Science and Technology, Krakow, Poland. He is an adjunct at the Institute of Computer Science AGH, a staff member at the ACC CYFRONET AGH, and Professor of Distributed System Enginnering at the Informatics Institute of the Universiteit van Amsterdam. In 2001–2004 he was the dean of the Faculty of Computer Science at the School of Banking and Management in Krakow. His research interests include distributed and grid systems; he is author and co-author of about 230 papers. He was leader and main investigatior of several Polish and international projects addressing parallel algorithms, irregular and out-of-core computing, monitoring of parallel applications, performance analysis, and grid systems. In the EU IST CrossGrid Project, he was the leader of the Architecture Team, he is the Scientific Coordinator of the K-WfGrid Project, the member of the Integration Monitoring Committee of the CoreGRID, and the workpackage leader in the ViraLab and GREDIA projects. He served as a program committee member, chairman and organizer of several international conferences (HPCN, Physics Computing, EuroPVM/MPI, SupEur, HiPer, ICCS, HPCC, e-Science'2006); he is co-editor of 16 proceedings of international conferences.

Bartosz Baliś obtained his Master degree in computer science from the AGH University of Science and Technology, Krakow, Poland. He is employed as research and teaching assistant at Institute of Computer Science, AGH. Co-author of 35 international publications including journal and conference papers and a book chapter. His research interests include monitoring and performance analysis of applications, Grid computing, scientific workflows, migration of legacy code to modern environments, and provenance tracking. Participant of several international research projects including EU-IST CrossGrid, CoreGrid, K-Wf Grid, VIROLAB, Gredia. Member of Program Committee for ICCS 2007 Conference and a reviewer at several other conferences including previous ICCS, EuroPVM/MPI, and e-Science 2006.

Tomasz Szepieniec obtained his M. Sc. in 2003 from the AGH University of Science and Technology, Cracow, Poland. He works on his Ph. D. as a researcher and developer at the ACC CYFRONET AGH. He was involved in several grid-related projects. He is an author of above ten scientific papers. Among other interests, he concentrates his scientific activity on applications running on grid environments, namely on their performance analysis and on application-level scheduling algorithms suitable for heterogeneous and dynamic environments.