International Journal of Computer Science and Applications, ©2009 Technomathematics Research Foundation
Vol. 6, No. 1, pp. 98 – 107

PRIVACY AND ANONYMITY PROTECTION IN COMPUTATIONAL GRID SERVICES

DEBASISH JANA

IT Department, Simplex Infrastructures Ltd, Kolkata 700 087, India
djana@alumni.uwaterloo.ca
http://debasishj.googlepages.com

AMRITAVA CHAUDHURI

Tata Consultancy Services, Kolkata, India
amritava.chaudhuri@gmail.com

BIJAN BIHARI BHAUMIK

Dept of Computer Science & Engineering, Jadavpur University, Kolkata, India
bbbhaumik@yahoo.com

In computational grid computing, grid nodes spanning over several diverse computing resources belonging to heterogeneous administrative domains form the backbone of Virtual Enterprise [VE]. In order to offer *service-on-demand*, various service providers, requesters, brokers and administrators collaborate in request-response manner among each other in Service Oriented Virtual Enterprise through service registry, service discovery and service binding mechanisms. Security issues for integrated and collaborative sharing of computing resources across heterogeneous administrative domains are principal concern. At the same time, the privacy and anonymity are also of prime importance while communicating over publicly spanned network like web. The individual service providers or requesters may not reveal their true identity to one another for privacy needs. Also, computational grid services may be required to be availed anonymously within the grid framework to keep the personal sensitive information about the service requester protected. This paper focuses on the protection of privacy and anonymity of grid stakeholders in the service oriented computational grid framework. An extension of onion routing has been used with dynamic token exchange along with protection of privacy and anonymity of individual identity.

Keywords: Grid computing, Computational Grids, Security, Privacy, Anonymity, SOA, XML.

1. Introduction

With the evolving computational need, the world of computing has faced a sea change with shift of entire paradigm from traditional centralized mainframe based computing to networked and distributed computing and now towards grid computing. The mounting popularity of the Internet and accessibility to high-speed networks along with abundance of affordable powerful personal computers, workstations and servers amplified the computing usage by leaps and bounds. With the advent of distributed computing with sophisticated load balancing, distributed data and concurrent computing power using clustered servers, the need of collaborative computing has been felt. The grid computing has emerged to cater the need of *computing-on-demand*. In grid computing, geographically dispersed heterogeneous computing stations belonging to diverse

administrative domains can connect to the grid and offer services or request services in a loosely coupled environment with services-on-demand style. Grid computing exploits idle or unused processing cycles of all capable computers in a network in order to solve problems, which may otherwise be excessively intensive for any single computing station. Similar to electric power grid, the computational grid provides a collaborative infrastructure with easy, consistent and inexpensive access to diverse computational resources belonging to heterogeneous administrative domains through a unified view of single virtual resource. Grid computing infrastructure thus involves integrated and collaborative sharing of computing resources forming Virtual Enterprises [VE].

While messages are exchanged among grid nodes belonging to public realm, the service stakeholders, particularly, the service providers and service consumers may need to conceal the identity. Sensitive information like user profile including demographic and geographic location identification in addition to navigation and personal likings and preferences need to be protected [Jana et al (2008)]. While security is of utmost importance for communication among geographically dispersed distributed grid nodes, concealing individual profiles and identities are worth concerning. Privacy and anonymity in the user profile management are primary concerns of this paper.

In this paper, we have focused on the protection of privacy and anonymity of grid stakeholders in the service oriented computational grid framework. An extension of onion routing with dynamic token exchange has been used for protecting from intruders within service oriented grid backbone. The organization of this paper follows. Section 2 discusses service oriented computational grid architecture basics in the light of grid computing. Section 3 talks about privacy and anonymity issues and concerns related to grid services. Section 4 presents some related works and some of our earlier works. Section 5 describes our implementation model using anatomy of SOAP and web services as applicable with onion routing to our solution architecture. Simulation experiments and results are presented in Section 6. Section 7 presents conclusion and future works.

2. Service Oriented Computational Grid Architecture

In Service-oriented architecture (SOA) [Jana (2006)], services are the crux of the architectural backbone.

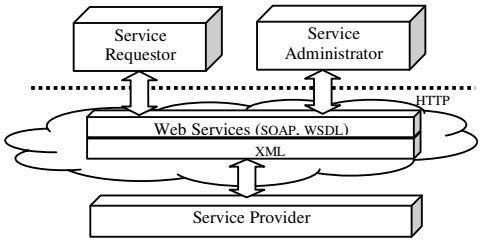


Fig. 1. Simple grid architecture

In SOA, we can view a relationship of the stakeholders – service providers, requesters or consumers and service registry or service administrators connected in a service-on-

demand scenario. Service providers register themselves with Service Registry, Service consumers discover or find provider for required services from service registry. Then, service consumer binds the service provider to execute the service request by the provider of services. Grid can be used as a layer (Fig. 1) through which service providers, service consumers or grid clients, grid manager or administrators communicate. XML format is used for platform agnostic communications with open standards for published web services in request / response mode using Simple Object Access Protocol (SOAP) or Web Services Definition Language (WSDL).

3. Privacy and Anonymity Issues

In a Virtual Enterprise (VE), a service provider as well as service requester or consumer must be allowed to define and enforce privacy policies [Foster (2004)] to protect sensitive information like personal information, credit history of a customer or some confidential data etc. While collating responses from multiple service providers, the master service provider or broker has to open individual sections of the form. But, the individual service provider is supposed to open the portion of the form designated for its own filling up and not to intervene with anyone else's area. This requires that the XML document be multi-parted and have some means to protect portions to undesired service provider. One level of concern is that the XML content, which may contain information of multiple heterogeneous service providers, may get exposed to one service provider, a node on the grid. This may have a breach of privacy between multiple service providers.

Many a times, privacy is closely resembled with anonymity that demands the need of being unidentified or unobserved while transacting over public domain such as web or other public realm. Adequate level of privacy needs to be achieved through controlled disclosure of identity and associated information. Anonymity can ensure achievement of privacy needs. In general, anonymous message transmission requires that the transacting message would not carry any information about the original sender and intended receiver.

4. Earlier Works

A service oriented grid framework based on the peer-to-peer paradigm has been talked about by Amoretti et al (2005). Phatanapherom et al (2003) showed a simulation model for grid scheduling. Foster et al (2002a, 2002b) have talked about grid services for distributed system and associated security architecture [Foster et al(1998)]. Fujita (2004) showed the use of web services for dynamic collaborations. Security of grid resource is very important. Many platforms like Avaki (2003), Legion [Grimshaw (1994)], Globus [Globus (http://www.globus.org/toolkit), Foster et al (1999a, 1999b), Kanaskar (2005)] are the possible grid middleware choices. They all support authentication and coarse-grained security [Bertino (2004), Haider (2006)]. Our earlier work [Jana and Bhaumik (2004)] was on security protection through hierarchical administrative servers and single signon across several administrative domains. We worked with fine-grained hierarchical role based grid access control [Jana et al (2007)]. The applicability of security issues and

its protection in ubiquitous environment was investigated in one paper [Jana et al (2006b)]. Security Model of Service Oriented computational grids was explored by Jana et al (2006a, 2006c). Our earlier work [Jana et al (2005a, 2005b)] used dynamic user credential management by using dynamic token generation during session establishment and ongoing communication. In this approach, it is possible to provide more security and less hack-proneness in grid environment in public realm. The scheme for dynamic token generation in a grid environment ensures more security and less hack-proneness because of the dynamic changing of the token, used in all transactions. The dynamic token thus generated forms part of the private key and the user id of the client provides the public key in terms of PKI. Public Key Infrastructure (PKI) is the most widely adopted security infrastructure used in Grid environments [Foster(2001)]. In addition to PKI based security tools, traditional security issues have been managed through well-known identity management [Lim et al(2004, 2005)] and access control technologies, e.g. X.509 certificates[Tuecke et al(2004)], Secure Sockets Layer (SSL) communication protocol etc. Another of our work [Jana et al (2005c)] used the idea of encrypting or signing a portion of a XML document, so that same XML document can float through multiple service providers with different sections locked for view by a particular server. The particular server, upon receipt of the XML content can unlock (view) only his designated portion and response is padded in. The XML content then floats around to the next server in the chain. Finally, it comes back with all the reply, keeping confidentiality and privacy of the individual service providers.

5. Our Implementation Model

Pseudonymous or anonymous communication can be achieved through *onion routing* [Onion Routing (Wiki)] technique, originally developed by D Goldschlag, M Reed, and P Syverson. In onion routing, a set of encrypted layers are responsible for encoding routing information, thereby provides the cloud of routing onions comprised of several routers or nodes. Onion routing layer provides protection of privacy by concealing the identity of the sender and recipient of a message. It also protects the underlying message through encryption during inter-router traversals in a network. It also provides a strong degree of unlinkability, so that an eavesdropper cannot easily determine both the sender and receiver of a given message at a given time, thereby ensuring anonymity to the greatest extent [Jana et al (2008)]. However, an eavesdropper or attacker with the ability to monitor an under-loaded onion router in a network might be able to trace the path of a message through the network and intrude. Onion routing networks become vulnerable to such intrusion or intersection attacks and predecessor attacks. Intersection attacks are caused due to failing or leaving nodes in a network. Predecessor attacks are facilitated through session tracking of the infected node.

Onion routing does not provide any absolute guarantee of privacy; rather, it provides a continuum in which the degree of privacy is generally a function of the number of participating routers versus the number of compromised or malicious routers. The degree of privacy can be given as below:

Say, Rn = Total # of routers in a network

Rp= number of compromised or malicious routers

Degree of privacy = P = f(Rn, Rp)

Our testbed environment uses Java [Jana (2005)] for implementation. The server or service providers or the service collators have the web page as well as servlet [Danny (2003)] and JSP container as the application server (Tomcat). We implemented using Java Servlet on Tomcat 5.x using SOAP-RPC and Messaging combination. Java servlets are loaded when the client browser intends to load the designated servlet, thereby communication is initiated and continued through SOAP-XML messages. We have adopted architecture similar to distributed-core architecture of Canali et al (2006). In our architecture (Fig. 2), the server nodes are divided into edge nodes (En nodes), lying closer to the network edge, and core nodes (Cn node), placed in strategic positions within the cloud. High user-perceived performance is the goal. The edge nodes will finally collate the services from several service providers, while the core nodes are responsible to form the part of onion routing cloud for finding out and finally availing the desired service. Here follows the steps as shown in the Activity Diagram presented in Fig. 2.

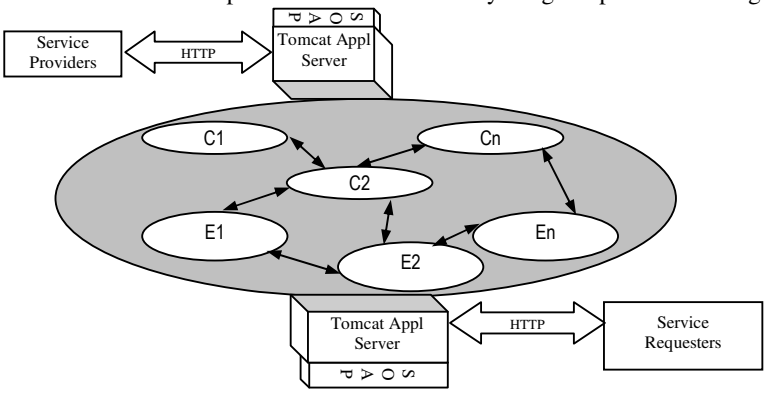


Fig. 2. Architecture of the model

At first, a service requester requests for a service. This request is received by a designated Edge node. The edge node extracts the identification of the service requester and generates a dynamic token (ticket) based on internally generated encryption technique similar to our dynamic token generation algorithm. It then finds out a core node in the onion routing layer to get the desired service. The edge nodes store the sender's details and then find the suitable core node in an onion routing framework. The core node then seeks for the subsequent core nodes in onion routing methodology and establishes a path to flow the message to the respective Service Provider or a set of Service Providers. The nodes may have multiple roles of a broker, collator, sender on behalf of some other service requester etc. The message flow is done in an encrypted mode. After the service provider(s) complete the processing, the response is sent back to the service requester back in the same path as that established during request sending. The broker, on its own or others in the cloud collates responses from several service providers, if applicable. The collation may require authenticating the ticket, which is all

along flowing with the messages. The final results are sent back to the edge node. Edge node in turn sends responses back to the requester. However in a practical scenario, intermediate nodes may not be available and therefore the encrypted message may get lost. This situation can be better handled by the introduction of an Authentication Server in the Onion Routing Grid framework, where every time a node gets a message registers the sender and receiver information along with the dynamically generated ticket.

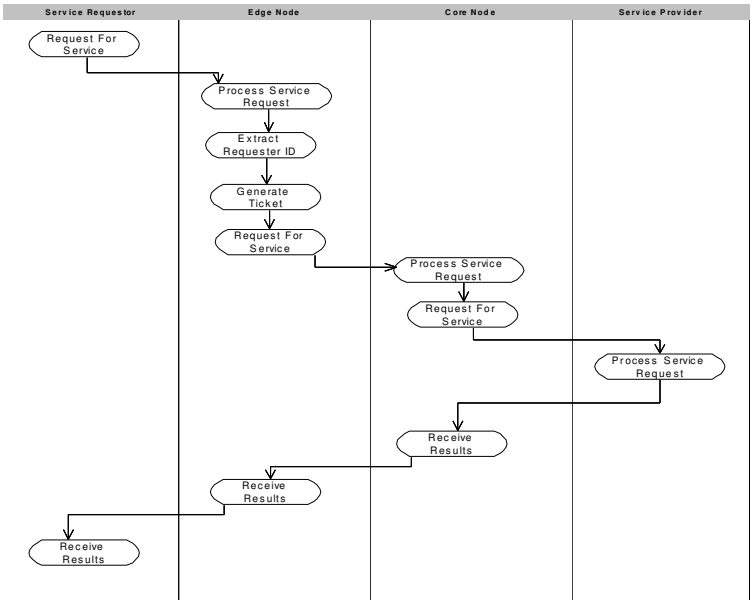


Fig. 3. Activity Diagram showing interactions among Service consumers, Edge Nodes, Core Nodes and Service providers

Every time the ticket gets changed or the intermediate nodes fail or become unavailable, the node that currently holds the message can query the Authentication server to fetch the path of message traversal. By this technique the request receiving path and the response sending path may not be the same. We have used the dynamic ticket generation concept for overcoming the drawback of onion routing to ensure the availability of return path in case of malicious attacks. Also, signing XML document in portion and encrypting that portion help to achieve privacy of the content in concern.

6. Simulation Experiments and Results

A number of simulation experiments have been conducted to evaluate the applicability of the implementation model architecture for protection of privacy and anonymity. We have used GridSim [Buyya (2002)]. The service requesters, service providers, brokers and their collaborations are in the backbone. Our implementation model uses Java [Jana (2005)] as the implementation language. The results of our simulation experiments reveal that the model has high scalability and robustness and suitable to achieve the privacy and anonymity with insignificant load as expected.

The simulation results in Fig. 4(a) shows that increasing number of service requesters has an impact on turnaround time with positive slope, i.e. turnaround time increases with increase of requesters keeping other parameters i.e. providers, edge and core nodes remaining same. The simulation results in Fig. 4(b) shows that average turnaround time increases with increase of processing time of service requests. This means more computationally intensive a job, demands more processing time as well as turnaround time. The simulation results in Fig. 4(c) shows that average turnaround time decreases with increase of edge nodes, eventually stabilizes. With increased number of edge nodes, service requesters get more edge nodes to handle the request thereby reducing the turnaround time. The results of Fig. 4(d) shows average turnaround time increases with more core nodes because of the inherent delay incurred due to more number of layers to reach the provider and vice versa.

Fig. 4 (a). Number of Service Requesters vs Turnaround Time (Avg)

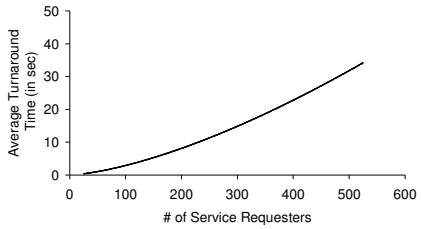


Fig. 4 (b). Processing Time of Service Requests vs Turnaround Time (Avg)

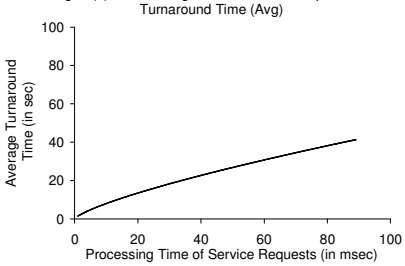


Fig. 4 (c). Number of Edge Nodes vs Turnaround Time (Avg)

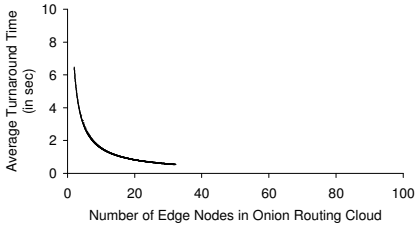
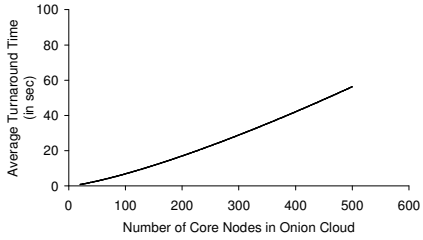


Fig. 4 (d). Number of Core Nodes vs Turnaround Time (Avg)



With increased number of Edge nodes as well as Core Nodes in the Onion cloud, expectedly the overall turnaround time increases with more protection for privacy and anonymity. Number of layers increases the degree of privacy and anonymity. On the other hand, keeping the number of Edge as well as Core nodes in Onion cloud same, and varying Service Providers show us that more service providers will ease the service availability with more fault tolerance and availability reducing the time to wait for services. Increasing number of service requesters has an impact on the performance with more wait time for availing the service.

7. Conclusion and Future Work

By using XML encryption techniques to sign a portion of a multi-part document, we have protected the privacy of the stakeholder in concern. The concept of Onion Routing has

been used with enhanced features such as anonymity, unlinkability and inter-nodal encryption and merging the same with the existing features of privacy protection, trust, integrity, confidentiality and authorization in Service Oriented Computational Grid. To eradicate the drawback of Onion Routing we have introduced dynamic token generation and exchange in the system so that every node in the network gets tagged to a centralized Authentication server where it can register message sender and receiver details and also can get the updated refreshed token periodically. Thus we have ensured that the no encrypted message/request gets lost in an Onion Routing network during traversal even if the intermediate nodes fail or get malicious. This is to ensure the fail-safe mechanism without compromising privacy and establishing anonymity.

Our current research is focused towards establishing a hierarchical authentication based services to reduce the load of the authenticating the dynamic ticket while ensuring security in grid transactions.

Acknowledgement

The authors wish to thank the Department of Computer Science & Engineering of Jadavpur University for the facilities extended while conducting the implementation for this work in a simulated environment for grid.

References

Core and hierarchical role based access control (RBAC) profile of XACML. (2005). v2.0. Standard, OASIS, February 2005.

Onion Routing. (2008). http://en.wikipedia.org/wiki/Onion_routing, accessed August, 2008.

Amoretti, M., Zanichelli, F., and Conte, G. (2005). "SP2A: a service-oriented framework for P2P-based Grids", In Proceedings of the 3rd International Workshop on Middleware For Grid Computing, Grenoble, France, Nov 28-Dec 02, 2005. vol. 117. ACM Press, New York, pp. 1-6.

Avaki: Avaki Compute Grid. (2003). http://www.avaki.com/products/acg.html, Apr. 2003.

Baker M, Buyya R, Laforenza D(2002):Grids and Grid technologies for wide-area distributed computing.Software: Practice and Experience, Vol 32(15), 2002, John Wiley & Sons, pp. 1437 – 1466.

Berstis V. (2002): Fundamentals of Grid Computing. IBM Red Book, 2002, pp. 1-28.

Bertino E, Mazzoleni P, Crispo B, Sivasubramanian S. (2004): Towards supporting fine-grained access control for Grid Resources. Proceedings of 10th IEEE International Workshop on Future Trends of Distributed Computing Systems 2004, FTDCS'04, May 2004, pp. 59-65.

Buyya R and Murshed M. (2002): GridSim: A Toolkit for the Modeling and Simulation of Distributed Resource Management and Scheduling for Grid Computing. Concurrency and Computation: Practice and Experience, Volume 14, Issue 13-15, Pages: 1175-1220, Wiley Press, New York, USA, November - December 2002.

Canali, C.; Colajanni, M.; Lancellotti, R. (2006): Distributed Architectures for High Performance and Privacy-Aware Content Generation and Delivery. Automated Production of Cross Media Content for Multi-Channel Distribution, 2006, Proceedings of AXMEDIS '06. Second International Conference on Dec. 2006 pp. 11 – 18.

Cannataro M, Talia D. (2003): Towards the Next-Generation Grid: A Pervasive Environment for Knowledge-Based Computing. Proceedings of the International Conference on Information Technology: Computers and Communications (ITCC'03), 28-30 April 2003, pp. 437–441.

Coward Danny, Yoshida Yutaka. (2003): Java™ Servlet Specification Version 2.4, Sun Microsystems, November, 2003.

eXtensible Access Control Markup Language (XACML) (2005). V2. Standard, OASIS, Feb 2005.

Foster I., Kesselman C. Eds. (2004): The Grid: Blueprint for a New Computing Infrastructure. Second Edition, The Elsevier Series in Grid Computing, 2004.

Foster I, Kesselman C, Tsudik G, Tuecke S(1998):A security architecture for computational grids. Proc. of the 5th ACM conference on Computer and communications security, 1998, pp.83–92.

Foster I(2001):The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid, 15-18 May 2001, pp. 6–7.

Foster I. and Kesselman C. (1999): Globus: A Toolkit-Based Grid Architecture. Foster, I. and Kesselman, C. (eds.), The Grid: Blueprint for a New Computing Infrastructure. Morgan Kaufmann, 1999. pp. 259—278.

Foster I., and Kesselman C. (1999): Globus: A Toolkit-Based Grid Architecture. The Grid: Blueprint for a New Computing Infrastructure, I. Foster and C. Kesselman, eds., Morgan Kaufmann, San Francisco, 1999, pp. 259-278.

Foster I., Kesselman C., and Tuecke S. (2001): The Anatomy of the Grid: Enabling Scalable Virtual Organizations. Int'l J. High-Performance Computing Applications, vol. 15, no.3, 2001, pp.200-222; http://www.globus.org/research/papers/anatomy.pdf.

Foster I., Kesselman C., Nick J., Tuecke S. (2002): The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.

Foster Ian, Kesselman Carl, Nick Jeffry M., Tuecke Steven. (2002): Grid Services for Distributed System Integration. IEEE Computer, June 2002, pp 37-46.

Fujita S. (2004): Dynamic Collaboration of Businesses using Web Services. NEC Journal of Advanced Technology, Vol. 1, No. 1, Jan, 2004, P. 36-42.

GLOBUS – The Globus Project: The Globus Toolkit (2008). http://www.globus.org/toolkit/, accessed August, 2008

Grid Security Infrastructure. (2007). http://www.globus.org/security/, accessed December, 2007.

Grimshaw A., Wulf W., French J.,. Weaver A and Reynolds P. (1994): Legion: The Next Logical Step Toward a Nationwide Computer. Tech. Report, University of Virginia, UMI Order Number: CS-94-21, Jun. 1994.

Haidar D A, Cuppens-Boulahia N, Cuppens F, and Debar H. (2006): An extended RBAC profile of XACML. Proceedings of the 3rd ACM Workshop on Secure Web Services (Alexandria, Virginia, USA, November 03 - 03, 2006), SWS '06. ACM Press, New York, NY, pp. 13-22.

Jana Debasish (2005): Java and Object-Oriented Programming Paradigm, First Edition, ISBN 8120327756, PHI Learning Pvt Ltd, 2005.

Jana Debasish. (2006): Service Oriented Architectures – A New Paradigm, CSI Communications, Computer Society of India, March, 2006, pp. 12-14.

Jana Debasish, Chaudhuri Amritava and Bhaumik Bijan Bihari. (2008): Privacy Protection In Anonymous Computational Grid services. IEEE International Region 10 Conference, IEEE TENCON 2008, Hyderabad, India, Nov.18-21, 2008 [Accepted for oral presentation]

Jana Debasish, Chaudhuri Amritava, Datta Abhijit and Bhaumik Bijan Bihari. (2007): A Fine-Grained Hierarchical Role Based Grid Access Control. IEEE India International Conference, Proceedings of the IEEE INDICON 2007, September 06-08, 2007, Bangalore, pp. 1-5.

Jana Debasish, Chaudhuri Amritava and Bhaumik Bijan Bihari. (2006): Security Model of Service Oriented Computational Grids. IEEE India International Conference, Proceedings of the IEEE INDICON 2006, September 15-17, 2006, New Delhi, pp. 1-5.

Jana Debasish, Chaudhuri Amritava, Datta Abhijit and Bhaumik Bijan Bihari. (2005): Dynamic User Credential Management in Grid Environment. IEEE International Region 10 Conference, Proceedings of the IEEE TENCON 2005, Nov.21-24, 2005, Melbourne, Australia pp. 836–840.

Jana Debasish, Chaudhuri Amritava, Datta Abhijit and Bhaumik Bijan Bihari. (2005): Framework for Handling Security Issues in Interoperable Grid Services.Proceedings of the IEEE India International Conference(INDICON 2005), Dec.11-13, 2005,IIT Madras, Chennai, pp.280-285.

Jana Debasish, Chaudhuri Amritava, Datta Abhijit and Bhaumik Bijan Bihari. (2006): Interoperability and Security Issues of Grid Services for Ubiquitous Computing. Proceedings of the 4th ACS/IEEE International Conference on Computer Systems and Applications, March 8-10, 2006, Dubai UAE, pp. 1114-1117.

Jana Debasish, Chaudhuri Amritava, Datta Abhijit and Bhaumik Bijan Bihari. (2005): Privacy Protection of Grid Services in a Collaborative SOA Environment. IEEE International Region 10 Conference, Proceedings of the IEEE TENCON 2005, Melbourne, Australia, Nov.21-24, 2005, pp. 2252– 2257.

Jana Debasish, Chaudhuri Amritava, Datta Abhijit and Bhaumik Bijan Bihari. (2006): Security Challenges Of Computational Grids. Proceedings of the First International Conference on Emerging Applications of IT, CSI EAIT 2006, Feb 10-11, 2006, Kolkata, India, pp. 265-268.

Jana Debasish, Bhaumik Bijan B. (2004): Single SignOn for Grid Services. Proceedings of the First IEEE India Annual Conference(INDICON 2004), IIT Kharagpur,Dec 20-22, 2004 pp. 513-516.

Kanaskar N V, Topaloglu U, Bayrak C. (2005): Globus Security Model for Grid environment. ACM SIGSOFT Software Engineering Notes, Volume 30 Number 6, November 2005 , pp. 1-9.

Lim H.W. and Robshaw M.J.B.(2005):. A Dynamic Key Infrastructure for GRID. In, P.M.A. Sloot, A.G. Hoekstra, T. Priol, A. Reinefeld, and M. Bubak, Eds., Proceedings of the European Grid Conference(EGC 2005), Amsterdam, Netherlands, pp. 255-264. Springer-Verlag LNCS 3470.

Lim H.W. and Robshaw M.J.B. (2004): On Identity-Based Cryptography and GRID Computing. In M. Bubak, G.D.v. Albada, P.M.A. Sloot, and J.J. Dongarra, editors, Proceedings of the 4th International Conference on Computational Science (ICCS 2004), Krakow, Poland, pages 474-477. Springer-Verlag LNCS 3036, 2004

Lorch, M., Proctor, S., Lepro, R., Kafura, D., and Shah, S. (2003): First experiences using XACML for access control in distributed systems. In Proceedings of the 2003 ACM Workshop on XML Security (Fairfax, Virginia, October, 2003). XMLSEC '03. ACM Press, New York,pp 25-37.

Nagaratnam N., Janson P., Dayka J., Nadalin A., Siebenlist F., Welch V., Foster I, Tuecke S. (2002): The Security Architecture for Open Grid Services. White paper, Open Grid Service Architecture Security Working Group(OGSA-SEC-WG), July 2002.

Pearlman L., Welch V., Foster I., Kesselman C., and Tuecke S. (2002): A community authorization service for group collaboration. In Proceedings of the 3rd IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY'02), pages 50-59. IEEE Computer Society Press, June 2002.

Pereira A L, Muppavarapu V, and Chung S M. (2006): Role-Based Access Control for Grid Database Services Using the Community Authorization Service. IEEE Transactions On Dependable And Secure Computing, Vol. 3, No. 2, April-June 2006, pp. 156-166.

Phatanapherom S, Uthayopas P, Kachitvichyanukul V. (2003): Simulation-based scheduling: Dynamic scheduling II: fast simulation model for grid scheduling using HyperSim. Proc. of the 35th conference on Winter simulation: driving innovation, December 2003, pp. 1494-1500.

Siddiqui B. (2002): Exploring XML Encryption. IBM DeveloperWorks Article Series, Mar 1, 2002

Steiner J., Neuman B.C. and Schiller J. (1988): Kerberos: An Authentication System for Open Network Systems. In Proc. Usenix Conference, 1988, pp. 191-202.

Talia D. (2002): The Open Grid Services Architecture: Where the Grid Meets the Web. IEEE Internet Computing, Vol. 6, No. 6, 2002, pp. 67-71.

Tuecke S., Welch V., Engert D., Pearman L., and Thompson M. (2004): Internet X.509 public key infrastructure proxy certificate profile. The Internet Engineering Task Force (IETF), RFC 3820, June 2004.