Research Journal of Applied Sciences, Engineering and Technology 7(11): 2253-2260, 2014
DOI:10.19026/rjaset.7.523
ISSN: 2040-7459; e-ISSN: 2040-7467

Submitted: June 17, 2013 Accepted: August 12, 2013 Published: March 20, 2014

Research Article

Four Factor Secure Authentication and Authorization for Economic Grid

[1]M. Victor Jose and [2]V. Seenivasagam
[1]Department of CSE, Noorul Islam Centre for Higher Education,
[2]Department of CSE, National Engineering College, Kovilpatti, Tamil Nadu, India

Abstract: In this study, a low cost virtually single storage private grid has been implemented as an economic one. In this economic grid that has been overcome, without scheduling resource down problem and needs of dedicated resource storage. The resource storage is done within a virtual single private grid nodes using random object addressing to prevent stolen attack. If any grid resource goes down, its resource identity will be removed from the resource object table that is maintained in middleware and resource recovery is efficiently managed by replicas. To overcome the limitation of biometric authentication and maintain the secret privacy this authentication uses four factors such as username and password, Universal Unique Identifier (UUID) of Operating System, Mother Board serial number and MAC address. IP address spoofing is overcome by using Secret Token. This Secret token is generated using UUID of operating System, mother board serial number and MAC address. This proposed system is used in grid, sharing of messages between source and destination which prevents attacks and maintains privacy, integrity of data. This private grid is simulated in GridSim Toolkit 5.2 with various criteria and the results are verified and displayed.

Keywords: Object grid architecture, object storage, random object addressing, resource identity, resource object table, secure resource access

INTRODUCTION

In Object based Grid Architecture (OGA) Jose and Seenivasagam (2011), each system consists of a local and grid platform service. When the system boots, the grid manager service will sends a message of registration to the middleware or Monitoring and Information System (MIS). Using Local Service Active Protocol (LSAP) (Jose *et al*., 2012), without scheduling downtime problem has been overcome. OGA acts as a virtually single system; the resources connected with this are treated as the single components or peripherals, no need of separate costly memories, local memory shared from various nodes will be treated as single space i.e., (Gi = {gN1, gN2, gN3,….., gNn}). The grid environment is specified as Gi and the elements belong to the Gi are the nodes which get connected to it; the nodes can be specified as the gN1, gN2, …. gNn. Similarly the resources connected to the environment also treated as the single space, which is shown in Fig. 1.

To avoid the resource attacks and maintain the privacy, four factor authentications are proposed which are username and password, UUID of Operating System (OS) Mother Board serial number (MBsn) and MAC address of the grid user. To handle the resources, a new identity management system (Boneh and Franklin, 2001) and a secure file access control protocol through the Middleware of its own environment (Keahey *et al*., 2002) are introduced.

In this study we have proposed economic grid Jose *et al*. (2012), for the grid platform to enhance the security on the file access. Since, Security is a much more important factor in planning and maintaining a grid than in conventional distributed computing, where secure resource sharing (Pahlevi and Kojima, 2008) comprises the bulk of the secure activity. Furthermore, it is important to understand the issues involved in authenticating users (Jospth, 1993) and providing proper authorization (Ionut *et al*., 2005) for specific operations as well as resources in the grid. It maintains the secret privacy using four factors such as username and password, UUID of Operating System, Mother Board serial number and MAC address. Secret token is generated using UUID, mother board serial number and MAC address, which is used, sharing of messages between source and destination to prevents attacks and handling secure resources (Foster *et al*., 2008; Allock *et al*., 2002) privacy and integrity of data. In this architecture all of these where analysed and implemented as secure and powerful grid environment (Buyya *et al*., 2000; Buyya *et al*., 2002), Schwiegelshohn and Yahyapour (2003), which was implemented in Gridsim Toolkit 5.2 (Buyya *et al*., 2002) and various criteria were tested.

Corresponding Author: M. Victor Jose, Department of CSE, Noorul Islam Centre for Higher Education, Tamil Nadu, India

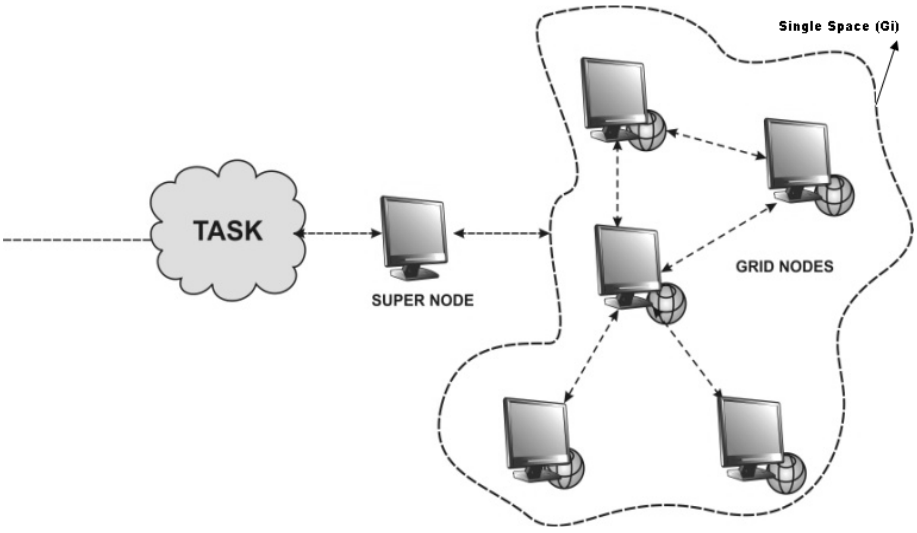

Fig. 1: OGA architecture

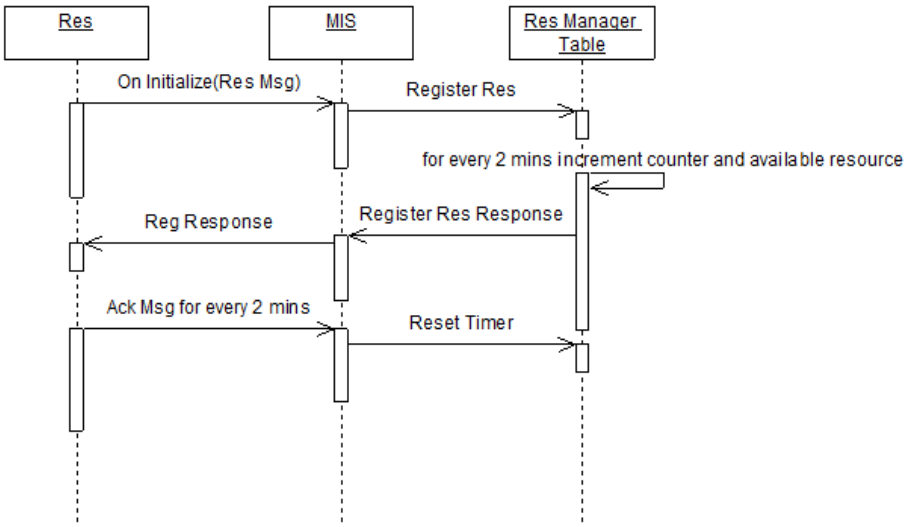

Fig. 2: Sequence diagram of LSAP

METHODOLOGY

Local Service Active Protocol (LSAP): In the design, local system running with grid manager service Jose and Seenivasagam (2011) can act as grid resource. when the system boots, grid manager Service gets loaded into memory and sends a registration message to the MIS. This message contains resource ID and the resource object, which contains information about grid resources and response counter. The Resource Manager Table (RMT), is a directory structure where object for each grid resource will be stored. RMT runs at MIS, for every 2 min increments the response counter of every resource object and it is registered to MIS. MIS sends the registered response to resource. After receiving registration acknowledgement, for every 2 min the resource sends an acknowledgement message to MIS, then resets the response count of resource object to 0 and process is repeated. The whole task is done by LSAP which will act as the bridge between the local platform and MIS. This process is denoted in sequence diagram Fig. 2.

Random object addressing for storage: In general, grid user specifies the place where the file should be stored in the dedicated storage where as in this design (Jose and Seenivasagam, 2011), there is no need to specify storage. MIS takes care of it; grid user has a feeling to work a single resource. The procedure to get random resource is:

Procedure Get_Random_RES
Begin
Do

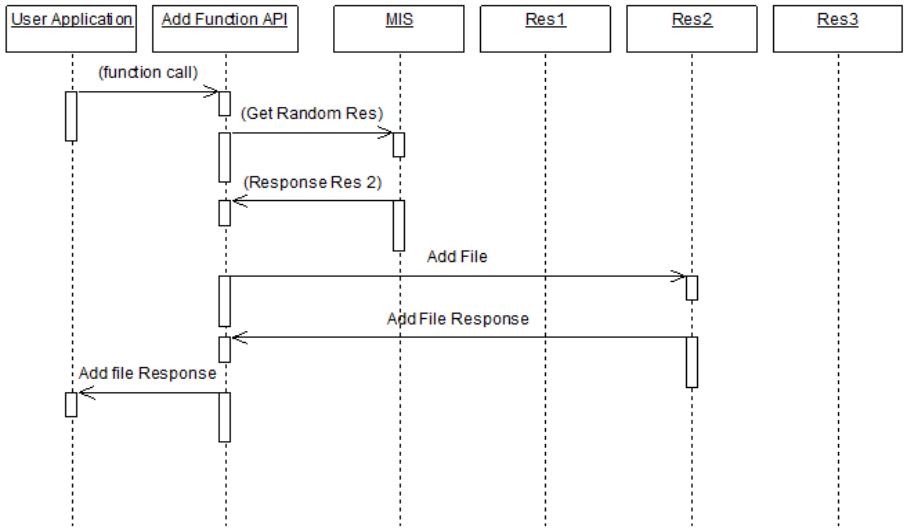

Fig. 3: Sequence diagram for add file response process

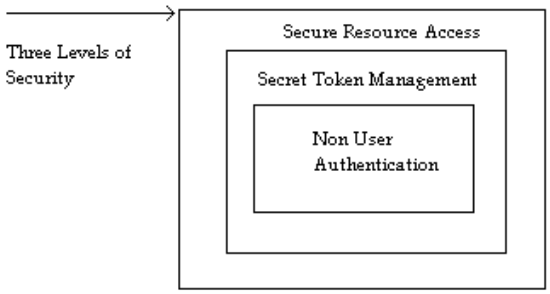

Fig. 4: Four factor authentication

begin
Index = Get a random no. between 1
and number of resource;
If (selected resource's response count is >5)
Then break do;
End do;
Return selected resource's id;
End Procedure

For example MIS takes care of assigning resource 2 to user application as add file response process is represented in sequence diagram Fig. 3.

Secure authentication and authorization : Many grid security research initiatives which have been proposed in Grid environment (Foster *et al*., 2002) are driven by the need to support scalable, dynamic, distributed Virtual Organizations (VOs) (Zhang *et al*., 2007; Sinnott *et al*., 2008). Important levels of authentication process are represented in Fig. 4.

In the initial level 'Secure Resource Access' is managed by Secure Resource Access Protocol (SRAP) to handle the secure file on the grid platform. It eliminates the attacks on the secure resources. Second level is 'Secret Token Management' authenticates (Jospth, 1993) the user by the user name, IP address and password. Next level is the non-user authentication and authorization method (Allock *et al*., 2002; Ionut *et al*., 2005), it is a pure system oriented verification, without the knowledge of user, system verifies the UUID of OS, MBsn and MAC address of the grid user. If all factors are quite ok, then authentication is successful and authorizes the user to access secure file. In the following session, these modules have been discussed in detailed manner.

SECURE RESOURCE ACCESS PROTOCOL

SRAP is proposed to handle the secure resource access in the grid environment (Buyya *et al*., 2002; Schwiegelshohn and Yahyapour, 2003). Access the secure resources is a challenging issue in the grid environment. If the users of the grid know that the resource is a protected one, the continuous attacks will be highly possible even by a trusted system. For example if a client on the grid may wish to protect a resource for some limited users, the grid has to face a big problem to solve this problem like ontology's (Keahey *et al*., 2002; Blanquer *et al*., 2009), Kerberos (Downnard, 2003; Ganesane, 1995) concepts etc., in the above security issue the cyclic attack is a massive problem, because this will make the file to Deny of Service (DoS). So to handle all this type of attack over a secure resource SRAP is introduced with the grid platform. It's a real-time grid platform protocol to enhance the security of the grid.

SRAP registration mechanism:

- **Resource administrator:** If a resource wants to secure, the resource administrator should register the resource with the middleware.

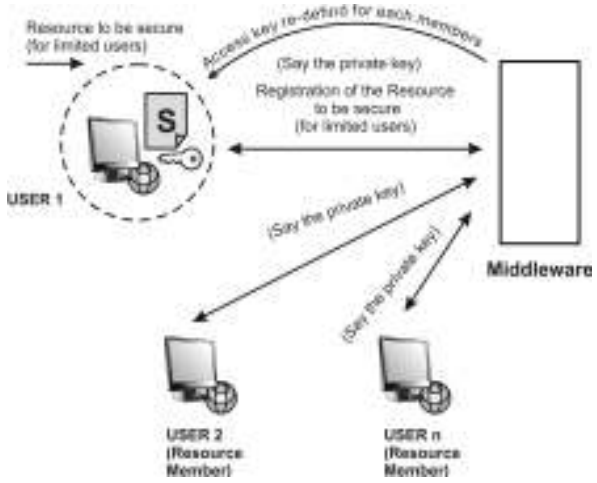

Fig. 5: SRAP resource registering process

- During registration the resource administrator should submit his private key with the middleware.
- The middleware will encrypt the resource by its private key and give an Access Key File (AKF) to access the resource.
- Similarly the members of the resource should also register with the middleware and added to the AKF (Section n).

The resource will be encrypted by the Asymmetric key encryption method by the middleware called public key cryptography. The RSA cryptography (Bellovin and Merritt, 1990) system is a prime example of public key cryptography. In public key cryptography, an asymmetric key pair (so-called public key and a private key) are used. Normally, the public key is present in the digital certificate that is issued by the Certificate Authority. Whichever (public/private) key encrypts your data, the other key is required to decrypt the data. A message encoded with the public key, can only be decoded with the private key. One of the keys is designated as the public key because it is made available, publicly, via a trusted Certificate Authority, which guarantees the ownership of each of the public keys. The corresponding private keys are secured by the owner and never revealed to the public. AKF is dynamically finalized by Resource administrator and middleware. The SRAP initial mechanism is illustrated in Fig. 5.

SRAP real-time resource access mechanism: SRAP is a real-time protocol with the grid platform, once the secure registration process completed by the members with the middleware through the resource administrator then the next step is the resource access. If any of the grid users needs the resource to be used and it is secured, the SRAP will handle the resource accessing system (Chakrabarti *et al*., 2008; Menasce and Casalicchio, 2004; Nagaratnam *et al*., 2003).

Step 1: The user has to request the resource (i.e., double click the resource or select the resource).
Step 2: If the resource is protected, SRAP will give an Access Request (AcReq) to the requester.
Step 3: Then the AcReq should be submitted to the middle ware.
Step 4: The middleware will process the AcReq and authenticate the user with the Access Key File.
Step 5: In the authentication process the middleware checks the UUID of operating System, Mother board serial number and MAC address, if it is ok, the middleware will give a permission key to the resource and give an access key to the resource requester to access the resource.
Step 6: Then the requester will submit the access key to the SRAP, so that the SRAP will permit the resource to access.

SRAP resource access procedures are diagrammatically represented in Fig. 6. A simplified pseudo code SRAP resource access algorithm as:

```
ALGORITHM SRAP (User Request, Resource
Access File)
{
Usr_Req ← User Request;
ResAcKey ← Resource Access File;
If (Usr_Req ≠ EMPTY) then
{
Middleware (Usr_Req, ResAcKey);
}
Middleware (Request, Access Key File)
{
User_i ←decrypt (Access Key File);
Verify. User_i (Usr_Req);
If (Usr_Req. User_Name == User_i. User_Name)
then
{If (Usr_Req.Password == User_i.Password) then
{If (Usr_Req. Uniq_Usr_ID ==
User_i.Uniq_Usr_ID) then
{Res_ Req.Status←ACCESS_RIGHT_OK;
SENT THE PERMISSION KEY TO RESOURCE;
}
Else
{Res_ Req.Status←ACCESS_RIGHT_DENY;
SENT THE DENY STATUS;
}}
}} END
```

Secret token management: The Grid computing services are featured by two important functions: group oriented communication (Yu and Buyya, 2005) and

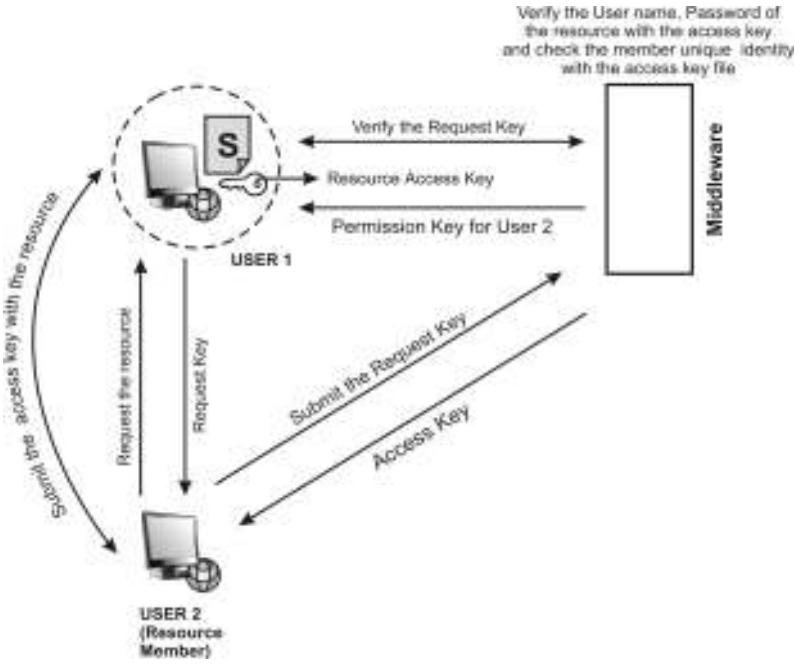

Fig. 6: SRAP resource access procedure

information sharing/exchange (Ernemann and Yahyapour, 2003). As long as communication and information exchange are conducted over the Internet, communication messages should be encrypted with a common key for confidentiality. However, due to the high dynamic nature of grid computing, how to update group key(s) efficiently and effectively becomes a challenging problem. In grid, every participating node offers its resources to be used by other nodes in a controllable manner. It is assumed that every valid user/machine in the system is assigned a permanent username. For example, when a user or an organization registers to the grid via the Middleware, several certificates need to be issued, including the host certificate that authenticates the machine involved in the grid, the service certificate that authenticates the services offered to the grid and the user certificate that authenticates the use of grid services. In this registration process, the permanent secret token can be embedded into the certificates issued to the member. The secret token is generated by grid node using system firmware. To prevent the attacks, this token is processed using stranded SHA1 algorithm. Token hash value is manipulated using UUID of OS, Mother Board serial number and MAC address. A simplified pseudo code generation of secret token algorithm as:

```
ALGORITHM Secret Token (Username,
Password)
{
UUID ← wmic, csproduct;
MBsn ← WMI Service.get Object ();
MAC address ← Network Interface address
Usr Name ←Accept (Username);
Secret Token = SHA1 (UUID || MBsn || MAC
address
|| Usr Name)
}
```

Non user authentication and authorization: In this authentication process username, password is collected from each node and without the knowledge of user UUID of operating System, Mother board serial number and MAC address of the grid user are implicitly verified by middleware with secret token. If all factors are true then authorization process begins and rights are granted by MIS. That is MIS sends the original key to the resource requester and permission key to the resource administrator.

SIMULATION RESULTS AND DISCUSSION

The simulation was made on the GridSim tool kit 5.2, which are real machines and connected by the Intranet. We can conduct our experiments in a realistic real world environment. Our experimental environment consists of 10 distributed nodes that serve as grid nodes. Our framework is implemented with GridSim tool kit 5.2, the schedule node and database server are also deployed on it. It is simulated over time and secure-shared resources with different capabilities and configurations.

The Results of experiments, Fig. 7 and 8 provided a basis for different types of resources with various configuration levels in the grid environment. In detail,

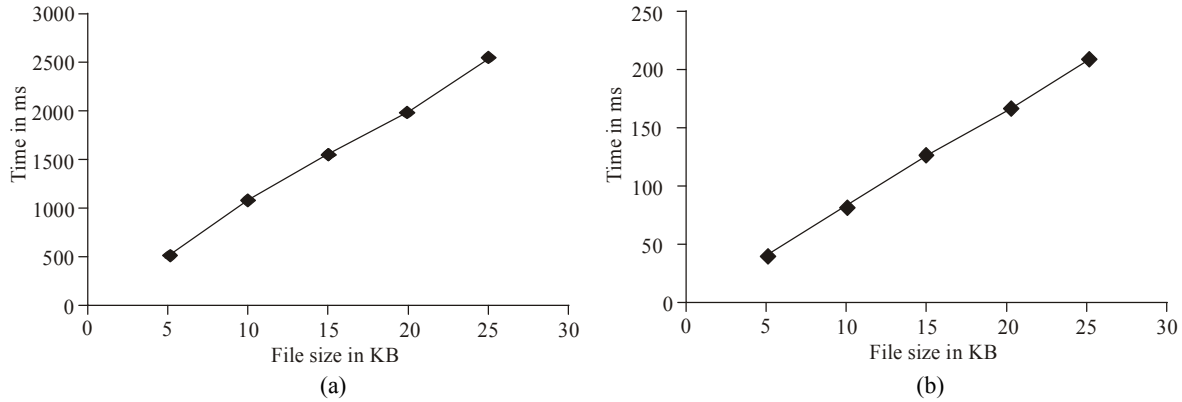

Fig. 7: (a) Authentication vs. time and (b) storage vs. time

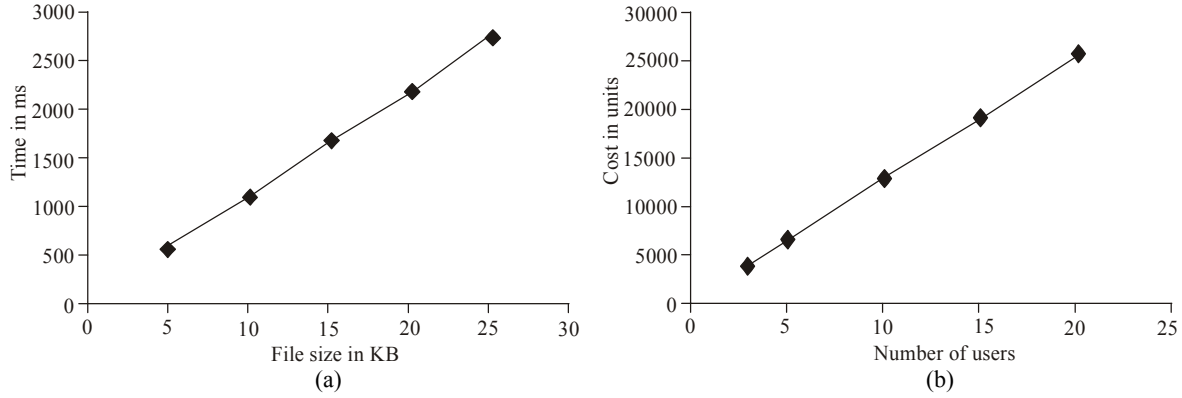

Fig. 8: (a) Secure resource sharing vs. time and (b) secure resource sharing vs. resources users

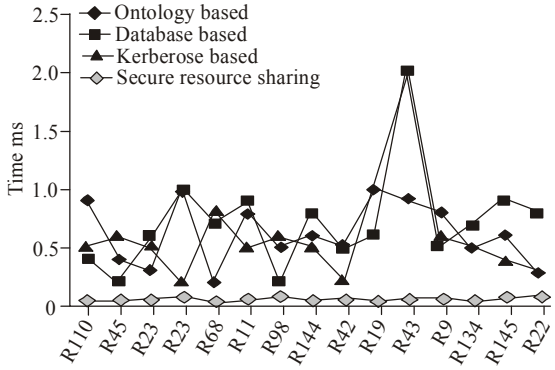

Fig. 9: Secure resource sharing vs. resources performance

performance of various grid resources were measured with the Secure Resource Sharing Protocol and timeliness attributes for each resource was measured. Then, we discovered from this dataset by applying the different type of key authentication algorithms such as Ontology based (Amarnath *et al*., 2009; Pernas and Dantas, 2005), Database based and Kerberos based (Bellovin and Merritt, 1990). Resources performance was shown in Fig. 9. This approach allows the identification of resources effectively in a simple way. The sample simulation code is shown in Fig. 10.

CONCLUSION

The algorithms were simulated over time and secure-shared resources with different capabilities and configurations. The architecture and components of the secure resource sharing authentication have been discussed. In addition, the average execution times on different resources and middleware have been tested. The results shown in these experiments are fine. It is believed that this work can help to give a real-time security on Grids for secure resource sharing. The future

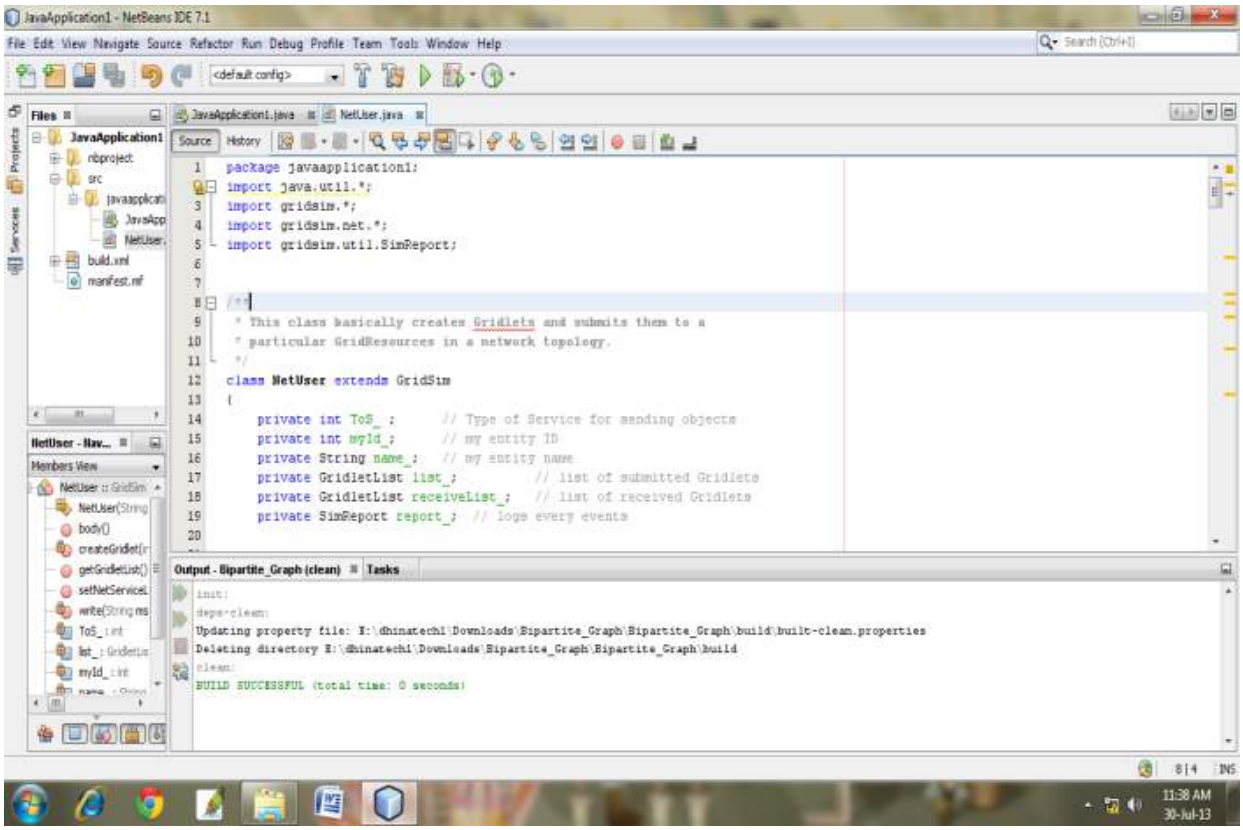

Fig. 10: IDE-simulation code

one is the cloud computing, which is the advanced version of the grid computing. A cloud has enormous power than a grid because cloud consists of a group of grid together. So the economic grid supports the cloud architecture also, in the same way the OGA economic grid and effective authentication are implemented in the cloud also.

REFERENCES

Allock, B., J. Bester, J. Bresnahan, A.L. Chervenak, I. Foster, C. Kesselman and S. Meder, 2002. Data management and transfer in high performance computational grid environments. Parallel Comput., 28(5): 749-771.

Amarnath, B.R., T.S. Somasundaram, M. Ellappan and R. Buyya, 2009. Ontology-based grid resource management. Software Pract. Exper., 39(17): 1419-1438.

Bellovin, S.M. and M. Merritt, 1990. Limitations of the Kerberos authentication system. ACM SIGCOMM Comput. Commun. Rev. 20(5): 119-132.

Blanquer, I., V. Hern´andez, D. Segrelles and E. Torres, 2009. Enhancing privacy and authorization control scalability in the grid through ontologies. IEEE T. inform. Technol. Biomed., 13(1): 16-24.

Boneh, D. and M.K. Franklin, 2001. Identity Based Encryption from the Weil Pairing. Proceedings of the 21st Annual International Cryptology Conference on Advances in Cryptology, pp: 213-229.

Buyya, R., D. Abramson and J. Giddy, 2000. An economy driven resource management architecture for global computational power grids. Proceeding of the 7th International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA 2000). Las Vegas, USA.

Buyya, R., D. Abramson, J. Giddy and H. Stockinger, 2002. Economic models for resource management and scheduling in grid computing. Concurr. Comp-Pract. E., 14: 1507-1542.

Chakrabarti, A., A. Damodaran and S. Sengupta, 2008. Grid computing security: A taxonomy. IEEE Secur. Priv., 6(1): 44-51.

Downnard, I., 2003. Public-key cryptography extensions into kerberos potentials. IEEE Potentials, 21(5): 30-34.

Ernemann, C. and R. Yahyapour, 2003. Applying Economic Scheduling Methods to Grid Environments. In: Nabrzyski, J., J.M. Schopf and J. Weglarz (Eds.), Grid Resource Management - State of the Art and Future Trends. Kluwer Academic Publishers, pp: 491-506.

Foster, I., C. Kesselman, J. Nick and S. Tuecke, 2002. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. The Globus Project. Retreieved form: www.Globus.org.

Foster, I., Y. Zhao, I. Raicu and S. Lu, 2008. Cloud computing and grid computing 360-degree compared. Proceeding of 14th IEEE Grid Computing Environments (GCE08). Fox Valley.

Ganesane, R., 1995. Yaksha: Augmenting Kerberos with the public key cryptography. Proceedings of the 1995 Symposium on Network and Distributed System Security, pp: 132-143.

Ionut, C., O. Daniel and N. Wolfgang, 2005. Policy Based Dynamic Negotiation for Grid Services Authorization. REWERSE-RP-2005.

Jose, M.V. and V. Seenivasagam, 2011. Object based grid architecture for enhancing security in grid computing. Proceeding of the International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), pp: 414-417.

Jose, M.V., V. Seenivasagam and P. Venkadesh, 2012. Minimum cost virtually single storage private grid using OGA. Eur. J. Sci. Res., 80(1): 77-86.

Jospth, P., 1993. Using Pre-authentication to Avoid Password Attack. OSFDCE Request for Comments 260.

Keahey, K., T. Fredian, Q. Peng, D.P. Schissel, M. Thompson, I. Foster, M. Greenwald and D. McCune, 2002. Computational grids in action: The national fusion collaboratory. Future Gener. Comp. Sy., 18(8): 1005-1015.

Menasce, D.A. and E. Casalicchio, 2004. QoS in grid computing. IEEE Internet Comput., 8(4): 85-87.

Nagaratnam, N., P. Janson, J. Dayka, A. Nadalin, F. Siebenlist, V. Welch, S. Tuecke, I. Foster, 2003. Security architecture for open grid services. Proceeding of GWD-I Document, GGF OGSA Security Workgroup.

Pahlevi, S.M. and I. Kojima, 2008. Semantic grid resource monitoring and discovery with rule processing based on the time-series statistical data. Proceeding of 9th International Conference on Grid Computing, pp: 358-360.

Pernas, A.M. and M.A.R. Dantas, 2005. Using ontology for description of grid resources. Proceedings of 19th International Symposium on High Performance Computing Systems and Applications (HPCS'2005), pp: 223-229.

Schwiegelshohn, U. and R. Yahyapour, 2003. Attributes for Communication Between Grid Scheduling Instances. In: Nabrzyski, J., J.M. Schopf and J. Weglarz (Eds.), Grid Resource Management. Kluwer Academic Publishing.

Sinnott, R.O., D.W. Chadwick, T. Doherty, D. Martin, A. Stell, G. Stewart, L. Su and J. Watt, 2008. Advanced security for virtual organizations: The prosand cons of centralized vs decentralized security models. Proceeding of the 8th IEEE International Symposium on Cluster Computing and the Grid, pp: 106-113.

Yu, J. and R. Buyya, 2005. A taxonomy of workflow management systems for grid computing. J. Grid Comput., 3: 171-200.

Zhang, N., L. Yao, A. Nenadic, J. Chin, C. Goble, A. Rector, D. Chadwick, S. Otenko and Q. Shi, 2007. Achieving fine-grained access control in virtual organizations. Concurr. Comp-Pract. E., 19: 1333-1352.