DOI:10.5392/IJoC.2010.6.3.015

Privacy Enhanced Security Mechanism for Grid Applications

Sang-Bae Park
Supercomputing Center,
KISTI, 335 Gwahangno, Yuseong, Daejeon, 305-806, Korea

ABSTRACT

Grid system is based on the Grid Security Infrastructure (GSI). GSI uses user's proxy to guarantee availability among multi-trust domains. Since grid system has been developed focusing on availability, GSI provides authentication and authorization performed by systems, but there are lacks of privacy consideration. For this reason, some researchers decide to use their own cluster system and do not want to use public grid systems. In this paper, we introduce a new privacy enhanced security mechanism for grid systems. With this mechanism, user can participate in resource allocation and authorization to user's contents more actively. This mechanism does not need to change previous middleware and minimize the computational overheads.

***Keywords**: Grid, Security, Privacy, Pilot Job System, GSI.*

1. INTRODUCTION

Now large-scale distributed computing systems, computational grid systems, are widely adopted for mass computations in many areas such as High Energy Physics, Biomedical researches, Astronomy, etc. High Energy Physics is the primary driver for the development of grid systems to unify theory, experiment, and simulation by using data exploration tools. The Worldwide Large Hadron Collider Computing Grid (WLCG) is the first and largest implementation of Ian Foster's grid definition [8].

Grid systems have distinctive characteristics from typical computation model. The main characteristic is that user population and resource pools are very huge and widely located even in different countries. It means grid system consists of multi-trust domains. And each domain's security level might be various. Moreover, there are various security protocols for identification, authentication, and authorization.

Grid security is mainly based on the Grid Security Infrastructure (GSI). [2],[3],[5],[9],[10] GSI defines useful security policy and related protocols based on the public key infrastructure (PKI). [1] The main idea of GSI is using a proxy that is delegated by user. Then system (e.g. gatekeeper or user interface) manages user's job using user's proxy without user's direct manipulation. And, operations that are confined to a single trust domain are subject to local security only. These are very useful to increase availability of grid system, but lead to a fact that user should trust a grid system fully. The weakest point of the whole systems determines the security level of the whole systems. So grid system might have a weak point against most people's expectations. This can be an obstacle for people to grid systems.

In this paper, we propose a new privacy enhanced security mechanisms for grid systems adopting a mutual-authorization concept. This makes users participate in various process of grid system more actively. So users can authorize grid elements (e.g. Computing Element, CE) to their contents in grid systems. In chapter 2, we discuss the privacy issues in grid systems. These are our motivations for this paper. Then we introduce a design for privacy enhanced mechanisms and brief discussion of this. With this mechanism we can expect to increase a security level for user's privacy. Since this mechanism complies with GSI, we can implement this mechanism without changing established grid middle-wares.

2. BACKGROUNDS

In this chapter, we briefly discuss some security issues in grid systems. Since grid system has developed focused in availability, there are some lacks of privacy considerations. At first, Grid security is based on GSI adapting a proxy concept. There are authentication and authorization performed by systems and most authorization processes depend on access control list (ACL). GSI had the following limitations [5].

- GSI does not consider a malicious active entity (e.g. hacker).
- GSI does not deal an encryption explicitly.
- People should trust a system fully.

Because grid system consists of many cluster systems world widely, there are multi-trust domains from well controlled big laboratories to small private systems. Each domain has its own

This is an excellent paper selected from the papers presented at ICCC 2009.
[*] *Corresponding author. E-mail : plucky@kisti.re.kr*
Manuscript received Mar. 29, 2010 ; accepted Sep. 20, 2010

www.kci.go.kr

security policy and different protection mechanism. If some active malicious entity tries to intrude a weak domain, he/she can get access right to whole system.

Another issue is related to storage component of grid. In grid system, most users' contents are stored in a public storage, storage element (SE). Since SE is a shared resource of grid system, SE has some characteristics for convenience[8],[9],[10].

- Contents in SE can be duplicated automatically.
- There is rare update in SE. There are only write and delete operations.

Since SE is relatively slow, SE might be a bottleneck in computations. So people make an automatic duplication functions and recommend once-write many-read usage. But, these might cause some privacy problems. Though there is an access control mechanism, we consider that our contents in grid system are open to everyone and transfer without our interference. If malicious active entity succeeds to intrude a SE then he/she can get all contents in that SE.

For these reasons, there are many dedicated user environment for their experiment or user groups. [3],[4] But, there are still lacks of privacy protection mechanisms. This can be an obstacle to lead to grid system for some research areas. For example, medical compound test for new medicine and clinical trial data should be protected very carefully. Researchers in those areas usually don't use a grid system but their own private cluster system.

3. A DESIGN FOR PRIVACY ENHANCED SECURITY MECHANISM

Now, we introduce a privacy enhanced security mechanism. This is implemented in application layer on the grid middleware. Usually, grid application consists of four steps: user's data preparation, job submission or computation, result storing, and result retrieval. In computation stage, user can patch an agent to computing nodes. This agent pulls user's real job and computes with user's data. We enhance the functionalities of this agent to achieve our aims.

The basic concept of our mechanism is a mutual authorization. Mutual authorization means user and system authorize each other. System elements of grid system are authorized by not only system but also user. This increases user's participation in grid operations. We design this mechanism with the following requirements.

- Mutual authorization: User takes part in resource allocation more actively.
- Privacy Enhancement: Only granted entity can read a user's contents.
- GSI compliance: There's no conflict with GSI policy and no need to alter established systems.
- Efficiency: There's a minimum overhead for cryptographic operations.

The following figure is the overview of this protocol. In fact, user's agent is running on computational nodes. But we just depict as a WMS, because WMS is a gate of grid system.

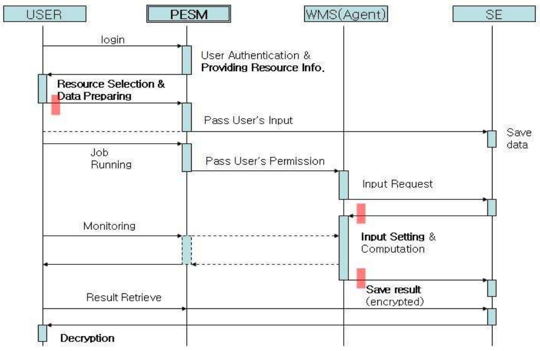

Fig. 1. Privacy Enhanced Security Mechanism

This mechanism consists of 5 steps, but additional step 0 can be regarded a part of step 1. Each step can be implemented by specified requirements of each system.

Step 0. (Group Key Management) If user wants to share the contents with others, user can designate a group and share the session key.

Step 1. (Data Preparing Stage) User login a grid system and prepare user's data for computation.
1.1 User login by a local authentication mechanism.
1.2 Server provides a list that user can use.
1.3 User encrypts his or her data and sends to SE.
1.4 User selects resource group and sends a cryptographic key to them.

Step 2. (Computation Stage)
2.1 Designated working nodes download user's data and decrypt it.
2.2 Compute user's job.

Step 3. (Storing Stage)
3.1 Encrypt a result by user's encryption key (user's key or group key).
3.2 Send encrypted result to SE.
3.3 Wipe out the local area.

Step 4. (Result Retrieval)
4.1 User contacts SE and download the result.
4.2 Decrypt it.

The main advantage of this mechanism is that user can choose computational resources and only granted nodes can read user's contents. Since this mechanism is not conflicts with GSI policy and located on the grid middle ware, there is no need to alter the existing systems. Most cryptographic operations that need heavy computations are done before job submission, so there's little overhead for grid computations.

The followings are a simple implementation with symmetric key encryption algorithm AES(Advanced Encryption Standard) and public key algorithm RSA [6],[7].

www.kci.go.kr

Step 1. (Data Preparing Stage) User login a grid system and prepare user's data for computation.

1.1 User login by a local authentication mechanism.
1.2 Server provides a list that user can use.
1.3 User chooses a random session K and encrypts his or her data (AES_Enc(Data, K)) and sends to SE.
1.4 User embeds AES decryption code (AES_Dec(Data, K)) into his agent.
1.5 User selects resource group and sends agents to selected resource.

Step 2. (Computation Stage)

2.1 Designated working nodes download user's data and decrypt it.
2.2 Compute user's job.

Step 3. (Storing Stage)

3.1 Encrypt a result by user's public encryption key (user's key or group key).
3.2 Send encrypted result to SE.
3.3 Wipe out the local area.

Step 4. (Result Retrieval)

4.1 User contacts SE and download the result.
4.2 Decrypt it.

This implementation provides basic functions that we mentioned above. Since agents are located in the intended nodes, user's data is decrypted just in intended one. Moreover, since user's contents in SE are encrypted, no one can read user's contents except designated agents.

Additional computation in computation stage is just a symmetric key decryption. Since symmetric key decryption is very fast, we don't mind the computational overhead.

Since resources are dynamically allocated and there are various resources in grid, we cannot check the overhead exactly. In this paper, we just compute theoretical overhead. We consider the following factors.

- j: number of jobs
- t: running time of each job
- i: input data size for a job
- o: output data size for a job
- e_a: AES encryption or decryption speed
- e_p: RSA encryption speed
- n: number of computational node

Without cryptographic operation, total running time T is simply $T = (j \times t) / n$. With our simple implementation, total running time $T' = (j \times i) / e_a + (j \times (t+ i/ e_a)) /n + (j \times o)/ e_p$. The first term is just for data preparation stage. The final term is the time for output encryption. Output encryption can be encrypted by AES and the session key is encrypted by RSA.

In [11], 1024 bit RSA encryption is done in 0.08 msec. 128 bit key AES encryption speed is 109 MB/sec. When we did a data challenge for finding a new material for malaria, we test 300,000 candidates and each test was done in about 20 minute. So we assume the similar situations. If number of jobs is one million, running time of a job is 10 minute, input and output size is 2KB, and we use 5000 node, then the running time T is 4000 minute and T' is (18.3 second + 4000 minute + 18.3 second + 98.3 second. Thus the overhead is less than 3 minute and this value is negligible for grid applications.

This implementation provides basic functions that we mentioned above. Since agents are located in the intended nodes, user's data is decrypted just in intended one. Moreover, since user's contents in SE are encrypted, no one can read user's contents except designated agents.

The limitations of this simple version are as followings.

- If computational nodes compromise after agent is installed, the whole contents can be exposed.
- The process to install agents should be protected.

For more security, we can consider some additional processes.

- When user encrypts user's data, user fragments his/her contents. Then, encrypts each fragments with different keys.
- User can give a grant access right to agents dynamically.

If user dispatches agent with periodic key, the agent is valid only for designated period. This can reduce the threat mentioned above. For one's own environment, more cryptographic protocols can be added to our mechanisms for more secure functions.

4. CONCLUSION

In this paper, we propose a privacy enhanced security mechanism for grid systems. This mechanism adopts a mutual authorization concept that lessens the obligation that people should trust a system fully. This mechanism can be implemented on existing grid middleware easily with very small overhead.

Although we just give a simple implementation example with basic requirements, we can easily expand this mechanism for more security. With this mechanism, we expect that more people use a grid system without privacy concern.

REFERENCE

[1] A. Menezes, P. C. van Oorschot, S. A. Vanstone, *Handbook of Applied Cryptography*, CRC Press, 1997.

[2] R. Alfieri, R. Cecchini, V. Ciaschini, L. dell'Agnello, A. Frohner, K. Lorentey, F. Spataro, "From gridmap-file to VOMS: managing authorization in a Grid environment", Future Generation Computer Systems 21, Elsevier, 2005, pp. 549-558.

[3] V. Korkhov, J. Moscicki and V. Krzhizhanovskaya, "Dynamic workload balancing of parallel applications with user-level scheduling on the Grid", Future Generation Computer Systems Vol. 25, Issue 1, Elsevier, 2009, pp. 28-34.

www.kci.go.kr

[4] Saiz, P. et al., “AliEn-ALICE environment on the Grid”, Nucl. Instrum. Meth., A502, 2003, pp. 437-440.

[5] I. Foster, C. Kesselman, G. Tsudik, S. Tuecke, “A Security Architecture for Computational Grids”, *Proc. of the 5th ACM Conference on Computer and Communications Security*, 1988, pp. 83-92.

[6] IEEE Std 1363-2000, IEEE Standard Specifications for Public-Key Cryptography, 2000.

[7] FIPS PUB 197, Advanced Encryption Standard, 2001.

[8] EGEE, http://public.eu-egee.org

[9] Globus Project, GSI, http://www.globus.org/security/

[10] gLite Middleware, http://www.glite.org

[11] http://www.cryptopp.com/benchmarks.html

Sang-Bae Park
He received the B.S. in Mathematics from Seogang university and, M.S from POSTECH in 1993 and 1995. Since then, he has been with IDIS, SoftForum and KISTI. His main research interests include cryptography and information security, and application protection.

www.kci.go.kr