International Journal of Computer and Communication Technology

Volume 3 | Issue 3 Article 4

July 2012

Studying Security Issues in HPC (Super Computer) Environment

Anirban Mitra
1MITS Innovation and Research Centre (MIRC) Department of CSE and IT, MITS Kolnara, Rayagada, Orissa 765 017, anir.mitra@gmail.com

Ramanuja Nayak
1MITS Innovation and Research Centre (MIRC) 1,2Department of CSE and IT, MITS Kolnara, Rayagada, Orissa 765 017, ramanuja.nayak@gmail.com

Studying Security Issues in HPC (Super Computer) Environment

[1]Anirban Mitra & [2]Ramanuja Nayak
[1]MITS Innovation and Research Centre (MIRC)
[1,2]Department of CSE and IT, MITS Kolnara, Rayagada, Orissa 765 017
[1]anir.mitra@gmail.com, [2]ramanuja.nayak@gmail.com

Abstract--**HPC has evolved from being a buzzword to becoming one of the most exciting areas in the field of Information Technology & Computer Science. Organizations are increasingly looking to HPC to improve operational efficiency, reduce expenditure over time and improve the computational power. Using Super Computers hosted on a particular location and connected with the Internet can reduce the installation of computational power and making it centralise. However, centralise system has some advantages and disadvantages over the distributed system, but we avoid discussing those issues and focusing more on the HPC systems. HPC can also be used to build web and file server and for applications of cloud computing. Due to cluster type architecture and high processing speed, we have experienced that it works far better and handles the loads in much more efficient manner then series of desktop with normal configuration connected together for application of cloud computing and network applications.**

In this paper we have discussed on issues re

lated to security of data and information on the context of HPC. Data and information are vanurable to security and safety. It is the purpose of this paper to present some practical security issues related to High Performance Computing Environment. Based on our observation on security requirements of HPC we have discuss some existing security technologies used in HPC. When observed to various literatures, we found that the existing techniques are not enough. We have discussed, some of the key issues relating to this context. Lastly, we have made an approach to find an appropriate solution using Blowfish encryption and decryption algorithm. We hope that, with our proposed concepts, HPC applications to perform better and in safer way. At the end, we have proposed a modified blow fish algorithmic technique by attaching random number generator algorithm to make the encryption decryption technique more appropriate for our own HPC environment.

Keywords: *Cluster Computing, Grid Computing, Random Number generator and Blow Fish Algorithm*.

1. INTRODUCTION

HPC has seen as the next generation architecture and service provider for IT Enterprise. In comparison to the traditional solutions, where the services are physically, logically and personally controlled, HPC facilitates to moves the application software, database to the large computing and data centers [1,3,9].

This technology has lead computing transformed to a model, consisting of various services which can be compared with the delivered in a manner similar to the utilities such as water, electricity and telephony. In such a model, users are allowed to access the services based on their requirements without thinking and knowing about the detail of installation and maintenance of the software at the remote end [11,12,14].

One of the issues for HPC is on security aspects. There are certain concerned about making this technique a fully secured network for not only data transfer but for also running various applications on those data. This new scenario has created many new security challenges which are yet to be explored. Various technique has been proposed on the direction of making the cloud and its computing – more secure and stable. We are proposing blow fish technique for achieving the security of data inside the computing cloud. Analytical results over proposed security methods shows that this method, once implemented may yield accepted results. Works are in advance to code it and put in the Campus Cloud – the HPC prototype project for providing service inside the campus [10, 13, 15, 19].

2. HPC Specification, Definitions, Characteristics, and Trends

2.1 Specification of Supercomputer (HPC)

Super Computer at MIRC: The HPC is installed with Cent OS and Scientific Linux ver 4.4, x86_64 and is with open source products having Intel Platform compatibility. Using Gluster-HPC ver 1.3 x86-64 bit edition the complete cluster along with Intel and gcc and gcc4 compiler makes it one of the most stable HPC.

H/W Platform is consisting of Intel Xeon Dual Quad core, 64 Bit. Having 01 head Node and 15 Compute Nodes connected internally with Gigabit Ethernet / Infiniband switch (Silverstrom 9024). Each node is made up of two processor of Intel Xeon Quad Core 2.66 GHZ, memory of 4 GB, two 250 GB SATA II disk

The working and architecture of HPC can be related with Cluster Computing and Grid Computing. In the following sub section, we briefly discuss about these two terms and characteristics.

2.2 Definition

Cluster Computing: A cluster is a type of parallel and distributed system, which consists of a collection of inter-connected stand-alone computers working together as a single integrated computing resource.

Grid Computing: A Grid is a type of parallel and distributed system that enables the sharing, selection, and aggregation of geographically distributed 'autonomous' resources dynamically at runtime depending on their availability, capability, performance, cost, and users' quality-of-service requirements.

After looking into several literatures and having an exposure to HPC (Supernova – Supercomputer), we experienced that HPC can be defined as a certain type of parallel and distributed system consisting of inter-connected and virtualized systems, making a unified computing resource. There are certain similarity and dissimilarity between the concepts of cluster computing and grid computing with that of working nature of HPC [4,18].

2.3 Characteristics

There are many characteristics those helps to distinguish cluster, grid and HPC systems. The clusters for the resource are located in a single administrative domain. This domain is managed by a single entity. In case of Grid systems, resources are geographically distributed across multiple administrative domains. Here, each domain have there own protocols, management policies and goals. Another major difference between cluster and Grid systems is of scheduling of applications. Cluster system scheduler, looks after the complete system and give priority for enhancing the overall system performance. Grid systems schedulers (resource brokers) focus on enhancing the performance of a specific application.

HPC techniques have evolved by accepting characteristics from both, clusters and grid computing. HPC give emphasis on computing, storing, and providing application services without any infrastructural limitation in client's environment [7].

3. Security Aspect in HPC

The main security threats that HPC faces can be solve by – (a) Protect applications and data' from system where computation execute, (b) Ensure that resources and data are not provided by a attacker, (c) Protect local execution from remote systems and (d) Different admin domains and Security policies [19].

Analysing of our problem, we found two of the basic cases had drawn our attention: (a) Access Control Service, which is used to protect all resources from unauthorized use and (b) Secure Communication Service, which provide authentication, message integrity and confidentiality and undeniable service of the parties that involved in the communication. Presently, Firewall, SSL, SSH and Kerberos are used for securing the network related to HPC environment [19].

Based on various literature surveys, security aspect in HPC can be discussed as: (a) Access security of nodes in cluster system. (b) Run security of the node. (c) Secure data transmission between user and node. (d) Secure move and data transmission of tasks among the nodes that participated in the tasks. (e) Prevent server from impersonation. (f) Prevent user from impersonation and (g) User's rights authentication [19].

When observed, one way to provide security to our clusters at HPC is to use encryption and decryption method. Since, internal data communication between clusters plays a vital role; we focused ourselves for such a method which already exists in data communication technique. We found Blow Fish Technique as one of the solution. This technique is already exists for data communication in network. It is also know as block cipher technique which makes 64 iterations in general. Blow Fish encrypting technique is efficient, fast to execute and hard to detect/decrypt without the knowledge of key. Looking at our requirement, we have made some minor modification in the Blow Fish Algorithm. In next section, we have mentioned the algorithm and in followed section, about its implementation in our Institute's Supernova Super Computer [2, 3, 5].

3.1 Blowfish Technique

Blowfish is a variable-length key, 64-bit block cipher. The algorithm consists of two parts: a key-expansion part and a data- encryption part. Key expansion converts a key of at most 448 bits into several sub key arrays totaling 4168 bytes. Data encryption occurs via a 16-round Feistel network. Each round consists of a key dependent permutation, and a key- and data-dependent substitution. All operations are XORs and additions on 32-bit words. The only additional operations are four indexed array data lookups per round [13, 15, 16, 17].

3.1.1 Sub keys

Blowfish uses a large number of sub keys. These keys must be pre-computed before any data encryption or decryption.

(i) The P-array consists of 18 32-bit sub keys: P_1, P_2,..., P_{18}.

(ii) There are four 32-bit S-boxes with 256 entries each:

$S_1,0$, $S_1,1$, ..., $S_1,255$;
$S_2,0$, $S_2,1$, ..,, $S_2,255$;
$S_3,0$, $S_3,1$, ..., $S_3,255$;
$S_4,0$, $S_4,1$, ..,, $S_4,255$.

3.1.2 Encryption

Blowfish is a Feistel network consisting of 16 rounds. The input is a 64-bit data element, x.

Divide x into two 32-bit halves: x_L, x_R:
For i = 1 to 16:
$X_L = x_L$ XOR P_i
$X_R = F(x_L)$ XOR x_R
Swap x_L and x_R
Next i
Swap x_L and x_R (Undo the last swap.)
$xR = x_R$ XOR P_{17}
$xL = x_L$ XOR P_{18}
Recombine x_L and x_R
Function F:
Divide x_L into four eight-bit quarters: a, b, c, and d
$F(x_L) = ((S_1,a + S_2,b \bmod 232)$ XOR $S_3,c) + S_4,d \bmod 232$

3.1.3 Decryption

Decryption is exactly the same as encryption, except that P_1, P_2, ..., P_{18} are used in the reverse order. Implementations of Blowfish that require the fastest speeds should unroll the loop and ensure that all sub keys are stored in cache.

3.1.4 Sub-Keys Generation

The sub keys are calculated using the Blowfish algorithm. The exact method is as follows:

(i) Initialize first the P-array and then the four S-boxes, in order, with a fixed string. This string consists of the hexadecimal digits of pi (less the initial 3). For example:

(ii) P_1 = 0x243f6a88, P_2 = 0x85a308d3, P_3 = 0x13198a2e, P_4 = 0x03707344

(iii) XOR P_1 with the first 32 bits of the key, XOR P_2 with the second 32-bits of the key, and so on for all bits of the key (possibly up to P_{14}). Repeatedly cycle through the key bits until the entire P-array has been XORed with key bits. (For every short key, there is at least one equivalent longer key; for example, if A is a 64-bit key, then AA, AAA, etc., are equivalent keys.)

(iv) Encrypt the all-zero string with the Blowfish algorithm, using the sub keys described in steps (1) and (2).

(v) Replace P_1 and P_2 with the output of step (3)

(vi) Encrypt the output of step (3) using the Blowfish algorithm with the modified sub keys.

(vii) Replace P_3 and P_4 with the output of step (5).

(viii) Continue the process, replacing all entries of the P- array, and then all four S-boxes in order, with the output of the continuously - changing Blowfish algorithm.

In total, 521 iterations are required to generate all required sub keys. Applications can store the sub keys rather than execute this derivation process multiple times [13], [14], [15].

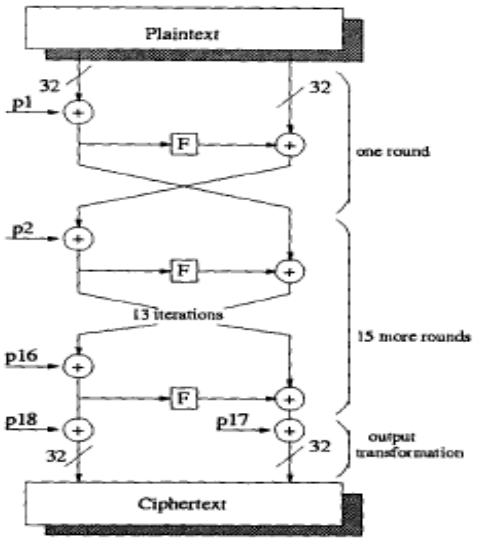

Fig 1. : Structure of Blow Fish

3.2 Implementation of Blow Fish in Supernova HPC at Campus

We have added a random number generator. This program will generate random number as key. Whenever, a specific user will log into the HPC for accessing the service, random number generator and blow fish algorithm will run. The generated number from the random number generator will be the key for the Blow Fish Technique. This key will remain for a single user until he logs off. Every time, the encryption and decryption process takes place for a single session, this key will be used.

Now, simultaneously, if any other user logs in, the random number generator will provide a different key to the blow fish algorithm. This key will remain active until the session expires. When the same is implemented in HPC, number of users can log in to use same or different clusters for data storing as well as for processing, the active session key will reduce the security bleach. After building the prototype using this concept, it was observer that, the technique of running the random number generator, followed by Blow fish

method works fine. Future works lies in making these methods integrated and cluster (centos) based.

4. CONCLUSION

HPC makes information and application for accessibility at faster speed and with more stability. The use of such computing resource can be made available any time, any place and through remote access. Data safety is one of the important issues in this method, specially, in case of sharing the same services and applications by different users. The issue becomes complicated in parallelism. In this paper we have discussed on issues related security of data and information on the context of HPC, specially, during accessing of services and resource. We have discussed here on Blow Fish Technique and its implementation. We hope that implementation of this technique will make HPC applications to perform better and in safer way. We have made minor change in blow fish technique. We have integrated random number generator to generate the key in form of a random number. This will act as key for encrypt and decrypt purpose for a single session. This method is in preliminary stage of execution at the HPC in the campus.

5. REFERENCE

[1] Amazon.com, 2008, Amazon Web Services (AWS), Online at http://aws.amazon.com, 2008.

[2] D. L. G. Filho and P. S. L. M. Barreto, 2006, Demonstrating Data Possession and Uncheatable Data Transfer, *Cryptology ePrint Archive*, Report 2006/150, 2006, http://eprint.iacr.org/.

[3] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, 2008, Scalable and Efficient Provable Data Possession, *Proc. of SecureComm* '08, pp. 1–10.

[4] George Reese, 2009, Cloud Application Architectures, April 2009, *O'Reilly Media*.

[5] H. Feistel, 1973, Cryptography and Computer Privacy, *Scientific American*, v. 228, n. 5, May 73, pp. 15-23.

[6] J. Deamen, R. Govaerts, and J. Vandewalle, 1993, Block Ciphers Based on Modular Arithmetic, *Proceedings of the 3rd Symposium on State and Progress of Research in Cryptography*, Rome, Italy, 15-16 Feb 1993, pp. 80-89.

[7] J. Hendricks, G. Ganger, and M. Reiter, 2007, Verifying Distributed Erasurecoded Data, *Proc. 26th ACM Symposium on Principles of Distributed Computing*, pp. 139–146.

[8] J.-H. Evertse, 1988, Linear Structures in Blockciphers, Advances in Cryptology--EUROCRPYT '87, *Springer-Verlag*, 1988, pp. 249- 266.

[9] John Rittinghouse, 2009, HPC: Implementation, Management, and Security, *Amazon Book Stores*, 2009.

[10] K. D. Bowers, A. Juels, and A. Oprea, 2008, HAIL: A High-Availability and Integrity Layer for Cloud Storage, *Cryptology ePrint Archive, Report* 2008/489, 2008, http://eprint.iacr.org/

[11] M. A. Shah, M. Baker, J. C. Mogul, and R. Swaminathan, 2007, Auditing to Keep Online Storage Services Honest, *Proc. 11th USENIX Workshop on Hot Topics in Operating Systems* (HOTOS '07), pp. 1–6, 2007.

[12] Michael Miller, 2008, HPC: Web-Based Applications That Change the Way You Work and Collaborate Online, *Online Journal*, August 2008 issue.

[13] Q. Wang, K. Ren, W. Lou, and Y. Zhang, 2009, Dependable and Secure Sensor Data Storage with Dynamic Integrity Assurance, *Proc. of IEEE INFOCOM*, 2009

[14] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, 2008, MR-PDP: Multiple- Replica Provable Data Possession, *Proc. of ICDCS '08*, pp. 411–420.

[15] T. S. J. Schwarz and E. L. Miller, 2006, Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage, *Proc. of ICDCS* '06, pp. 12–12.

[16] T.W. Cusick and M.C. Wood, 1991, The REDOC-II Cryptosystem, Advances in Cryptology--CRYPTO '90 *Proceedings*, *Springer- Verlag*, 1991, pp. 545-563.

[17] X. Lai, J. Massey, and S. Murphy, 1991, Markov Ciphers and Differential Cryptanalysis Advances in Cryptology--*EUROCRYPT 91 Proceedings*, Springer-Verlag, 1991, pp. 17-38.

[18] Yunhong Gu, Robert L. Grossman, 2008, Sector and Sphere: *The Design and Implementation of a High Performance Data Cloud*, *UK*, 2008.

[19] Y. Shuyuan, H. Dake and W. Jianbo, Security Issues in National High performance Computing Environment, *IEEE 2003*, pp 227 – 230.