www.ijecs.in
International Journal Of Engineering And Computer Science ISSN:2319-7242
Volume 3 Issue 1 Jan, 2014 Page No. 3758-3762

On some Security aspect of HPC Environment

Subrata Paul[1], Anirban Mitra[2], Ramanuja Nayak[3]

[1]M.I.P.S, M.I.T.S, Rayagada, Orissa 765 017
subratapaulcse@gmail.com

[2]Departmemt of CSE, M.I.T.S., Rayagada, Orissa 765 017
mitra.anirban@gmail.com

[3]M.I.P.S, M.I.T.S, Rayagada, Orissa 765 017
ramanuja.nayak@gmail.com

Abstract:***HPC (High Performance Computing) or simply, supercomputer has evolved to become one of the most exciting and challenging areas in the field of information technology & computer science. In this paper, we have discussed on issues related to security of data and information in respect with HPC environment. Data and information are valuable and deserve a proper security and safety. The purpose of this paper is to present some practical security issues related to HPC or super-computing Environment. Based on our literature survey on issues related to security requirements of HPC, we have observed that some existing security technologies are already existing, which are being used in HPC. But, our observation also leads to a conclusion that improvements are required and the existing techniques are not enough. Further, we have discussed some of the key issues relating to this context and made an approach to find an appropriate solution using RSA encryption and decryption algorithm. We have studied the concept of parallelism in RSA environment and found that it produces a better and efficient computation result in respect of space and time complexity. In the later part of the paper, we have proposed a modified RSA algorithmic technique by attaching a large prime number generator algorithm to make the encryption decryption technique more appropriate for our HPC environment [20].***

Keywords:HPC, Supercomputer, Cluster Computing, Grid Computing, Random Number generator and RSA Algorithm.

1. Introduction

Organizations are increasingly looking forward towards HPC in order to improve operational efficiency, reduce expenditure over time and improve the computational power. Using Super Computers hosted at a particular location and connected with the networks can reduce the installation complexities and increase the power of computation. Thus making it centralize, centralized system has some advantages and disadvantages over the distributed system. 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 efficiently than a series of desktop with normal configuration connected together for application of cloud computing and network applications.

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 move 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 concerns about making this technique a fully secured network for not only data transfer but for also running various applications on those data. This scenario has created many new security challenges which are yet to be explored. Various technique has been proposed on the direction of making such environment and it's computing part – more secure and stable. Among various techniques available, we are proposing traditional RSA technique with some added characteristic for achieving the security of data inside the computing environment. Analytical results over proposed security methods shows that this method, once implemented may yield the accepted and satisfying results [6,10,13,15, 19].

2. HPC: Specification, Definitions and characteristics

Specification of Supercomputer (HPC)

This section of our paper focuses on the hardware and computing facility that we are having. Our HPC is installed with Cent OS and Scientific Linux version 4.4, x86_64 and is with open source products having Intel Platform compatibility. Using cluster-HPC version 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.

Hardware Platform of HPC consists of Intel Xeon Dual Quad core, 64 Bit. Having 01 head Node and 15 Compute Nodes connected internally with Gigabit Ethernet infinite band 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 its characteristics [6]. However, our HPC and its environment belong to Cluster Computing category.

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 [6].

*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 [6].

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 it an 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 [6,4,18].

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 their 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].

Analyzing 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 [6,19].

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 RSA Technique as one of the solution. This technique is already exists for data communication in network. It is also known as block cipher technique which makes 64 iterations in general. RSA 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 RSA Algorithm. In next section, we have mentioned the algorithm and in followed section, about its implementation in our Institute’s Supernova Super Computer [2,3,5,6].

3.1 RSA Encryption Technique

RSA is an Internet encryption and authentication system that uses an algorithm developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. The RSA algorithm is the most commonly used encryption and authentication algorithm and is included as part of the Web browsers from Microsoft and Netscape. The encryption system is owned by RSA Security. The company licenses the algorithm technologies and also sells development kits. The technologies are part of existing or proposed Web, Internet, and computing standards. The algorithm involves multiplying two large prime numbers (a prime number is a number divisible only by that number and 1) and through additional operations deriving a set of two numbers that constitutes the public key and another set that is the private key. Once the keys have been developed, the original prime numbers are no longer important and can be discarded. Both the public and the private keys are needed for encryption /decryption but only the owner of a private key ever needs to know it. Using the RSA system, the private key never needs to be sent across the Internet.

Keys in RSA are usually between 512 and 2048 bits, the larger the better for security. The RSA algorithm is as follows:

- Find P and Q, two large prime numbers (e.g.,1024-bit).
- E is chosen in a manner such that:
    - E is greater than 1.
    - E is less than PQ, and
    - E and (P-1) (Q-1) have no factors in common other than 1. (making it relatively prime).

  E does not have to be prime, but it must be odd. (P-1)(Q-1) cannot be prime because it is an even number.

- D is chosen such that (DE-1) is evenly divisible by (P-1) (Q-1), that is (DE-1)/ (P-1) (Q-1) is an integer. Simply if we can find an integer X that causes D=(X (P-1) (Q-1) +1)/E to be an integer, then that specific value of D is further used in the process.

Encryption and Decryption Functions.

The encryption function is:

$C= (T^E) \bmod PQ$

Where

C is the encrypted message (or ciphertext), a positive integer.

T is the message being encrypted (plaintext), a positive integer. T must be less than the modulus, PQ.

The decryption function is:

$T= (C^D) \bmod PQ$

Where

C is the encrypted message (ciphertext), a positive integer

T is the message being encrypted (plaintext), a positive integer.

The public key is the pair (PQ, E). The private key is the number D. PQ is called the modulus. E is the public exponent. D is the secret exponent. There is currently no known easy method of calculating D, P, or Q given only (PQ, E)(the public key) if P and Q are very large(1024 bit or more). PQ has to be factored to get P and Q. Once P and Q are in hand (with E), private key D could be obtained. Though it is widely suspected to be true, it is not yet proven that no easy method of factoring exists. It is not yet proven that the only way to crack RSA is to factorize PQ [20].

3.2 Parallelism in RSA Algorithm

In the basic operation of RSA: the public key (e, m) consist of an exponent and a modulus, and the encryption operation transforms a message t<m into t^e(mod m). The private key (d, m) uses the same modulus but a different exponent: the operation is the same modular exponentiation. The modulus m is chosen to be the product of two large prime numbers p and q, and e is usually chosen to be 3 or some other smaller number, so that public key operations are reasonably fast (about $O(n^2)$ operations , where n is the size of the modulus). The private exponent, d is of the same order as m, so private key operations require $O(n^3)$ time. Because of the great speed disparity, we will focus on speeding up the private key operations.

Given a message t, a modulus m, and an exponent $e = \sum_{i=1}^{n} e_i 2^i$, the basic modular exponentiation algorithm used by RSA loops over each bit e_i of the exponent:

- x :=t
- v :=1
- for i := 1 to n
- if e_i=1 then v :=vx(mod m)
- x := x^2(mod m)

The final value of v is the result, t^e (mod m).

There does not seem to be any way to perform modular exponentiation faster than the method of repeated sharing, so there appears to be an inherent sequentiality to the main loop of RSA. Nevertheless, there are four clear opportunities for parallelism in this basic algorithm:

Firstly, if there is a stream of messages to be decrypted, each of these operations may be performed independently on a different processor (or a set of processors). This process may not speed up the elapsed time for a single private key operation, but will surely speed up the overall throughput.

Secondly, the process of squaring in step 5 can be performed in parallel with the multiplication in step 4 from the previous iteration which may save up to 33% of the overall time. We should keep in mind that this is only possible with the loop running from low order exponent bits to high order. It is more common to scan the exponent in high to low order, but doing so removes some inherent parallelism.

Thirdly, for private key operations, we may assume that the factors p and q of m are known. The modular exponentiation can be performed separately and in parallel mod p and mod q, and then the two results can be combined by the Chinese remainder theorem. This is often done in fast software implementations, since two half size modular exponentiations are still four times as fast as one full size exponentiation.

Finally, the high precision multiplications in steps 4 and 5 are slow($O(n^2)$) operations. Using standard long multiplications with 32 bit words and a 512 bit modulus, there are 16 words in each number and 16^2=256 32x32 bit multiplies to perform. These multiplies can all be performed and the results are summed in parallel.

A good parallel implementation of RSA will certainly incorporate the first three of these ideas. The first performing several RSA operations independently on different processors, is the most scalable, and undoubtedly most important. But for speeding up the response time of a single operation the other three techniques are needed. Performing squaring in parallel with the multiplication and calculating the exponential modulo the two prime factors together gain only a factor of 3 [21].

3.3 Implementation of RSA Technique HPC Environment

RSA technique generates two large prime numbers. We then work on the generated prime numbers and using them and the co-prime of the product of the prime numbers we generate the private key. Since the HPC environment has a high computational ability so we do not face a problem in generating the prime numbers and generate the secret private key using them. Whenever, a specific user will log into the HPC for accessing the service, the prime number will be generated and the RSA algorithm will run. The generated

number from the prime numbers will be the private key for the RSA Technique. During encryption the sender will generate the Cipher text from the Plain text using the private key and the prime numbers.

Now, simultaneously, on the other end the receiver can decrypt the Plain text from the Cipher text using the private key and the prime numbers. Any other users except the sender and the receiver cannot have an access to the generated prime numbers and thus cannot harm the data transmitted. 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 observed that, the technique of running the large prime number generator, followed by RSA method works fine. Future works lies in making these methods integrated and cluster based.

4. Conclusion

In this paper, we have discussed on issues related to security of data and information in respect with HPC environment. Data and information are valuable and deserve a proper security and safety. The purpose of this paper is to present some practical security issues related to HPC or super-computing Environment. Among some of the existing security technologies in related area, used in HPC, our observation leads to a conclusion that improvements are required and the existing techniques are not enough. In this direction, this work had discussed some of the key issues relating to this context and made an approach to find an appropriate solution using RSA encryption and decryption algorithm. The paper deals on the concept of parallelism in RSA environment. Observation shows that it produces a better and efficient computation result in respect of space and time complexity. The later part of the paper, we have proposed a modified RSA algorithmic technique by attaching a large prime number generator algorithm to make the encryption decryption technique more appropriate for our HPC environment.

References

[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, Demonstrating Data PossessionandUncheatable Data Transfer, Cryptology ePrint Archive, Report2006/150, http://eprint.iacr.org/ ,2006.

[3] G. Ateniese, R. D. Pietro, L. V. Mancini, and G. Tsudik, "Scalable andEfficient Provable Data Possession", In Proceedings . of Secure Communication, pp. 1–10, 2008.

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

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

[6] A.Mitra and R.Nayak, " Studying Security Issue in HPC(Super Computer) Environment", In proceedings of International Journal of Computer & Communication Technology (IJCCT), Issue:2,3,4, pp 309-312, 2010.

[7] J. Hendricks, G. Ganger, and M. Reiter, "Verifying Distributed Erasure coded Data", In Proceedings of 26th ACM Symposium on Principles of Distributed Computing, pp. 139–146.

[8] J.-H. Evertse, "Linear Structures in Blockciphers", In Proceedings of Advances in Cryptology--EUROCRPYT '87, Springer-Verlag, pp. 249- 266, 1987.

[9] John Rittinghouse, HPC: Implementation, Management, and Security, 2009.

[10] K. D. Bowers, A. Juels, and A. Oprea, HAIL: A High-Availability and Integrity Layer for Cloud Storage, Cryptology ePrint Archive, Report2008/489, http://eprint.iacr.org/, 2008.

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

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

[1]3 Q. Wang, K. Ren, W. Lou, and Y. Zhang, "Dependable and Secure Sensor Data Storage with Dynamic Integrity" Assurance, In Proceedings. of IEEE INFOCOM, 2009.

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

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

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

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

[18] YunhongGu, Robert L. Grossman, "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.

[20] Barry Wilkinson, "Grid Computing-techniques and applications", CRC Press, pp-124-125, 2011.

[21] David Pearson, "A parallel implementation of RSA", Cornell University report, pp-2-3, July 1996.

Author Profile

Subrata Paul is currently working as Teaching Assistant in the MITS Institute of Professional Studies (MIPS), Rayagada (Odisha). He had completed his B.E(CSE) from VTU – Belgaum, Karnataka in 2010 and M.Tech(CS) from Berhampur University in the year 2013. His research area includes Social Network Analysis, HPC and Cloud Computing. He had several publications at national and international levels, both in journals and conferences including a paper in IEEE - conference. He had an experience of nearly 3 years in teaching undergraduate courses and 1 year in handling post graduate classes. He usually teaches papers like Software Engineering, Computer Network, Artificial Intelligence and Programming in C Language.

AnirbanMitra is currently working as Associate Professor in the Department of CSE at MITS – Rayagada (Odisha). He had submitted his Ph.D. thesis in Computer Science under Berhampur University. His research area includes KDD and Social Network Analysis and HPC. He had several publications at national and international levels, both in journals and conferences. He is associated with several technical societies as a member which includes IEEE, CSI, Institute of Engineers (India). He had an experience of nearly 7 years in teaching and had taught papers like Software Engineering, Computer Graphics, Artificial Intelligence and Internet Technologies for under graduate and post graduate courses

RamanujaNayak is presently working as an Associate Professor in the Department of CSE at MITS – Rayagada (Odisha) and also looking after MIPS as the Principal. He had received M.Tech.(CS) from Berhampur University and M.Sc.(Physics) from Khallikote College (Autonomous) Berhampur and MCA from IGNOU. Currently he is pursuing Ph.D. in Computer Science under Centurion University, Odisha. He is having an experience of more than 12 years in teaching and research. He had experience of teaching papers like Computer Network, Operating System, and Computer Organisation for under graduate and post graduate courses. He has also acted as a question setter and examiner for various autonomous college and few universities. His research area includes Social Network Analysis, Data Mining and High Performance Computing. He had published several papers in both national and international journals and conferences.