ISSN 2250-3153

Grid Computing and Security Issues

Rashmi Bhatia

Dept of Computer Applications, Dev Samaj College for Women
Sec-45B, Chandigarh, India

Abstract— Grid is a utility or infra-structure for complex, enormous computations, where remote resources are accessible through the web (internet), from desktop, laptop, mobile phone. It is similar to power grid, where the user does not have to worry about the source of the computing power. Grid can be thought of as aggregation of millions of discrete computers owned by individuals, institutes from various countries across the world connected to form a single, huge, super-computer! Undoubtedly it is an evolution of internet facility, but such aggregation of networked computer resources in dynamic and multi-institutional environment demand for higher security. This paper deals with the challenging security issues that demand new technical approaches. We describe how these issues can be resolved.

Index Terms- **Control grid, Cryptography, Digital Certificate, Handshaking, Kerberos, Middleware**

I. GRID COMPUTING

At its most basic level, grid computing is a computer network in which each computer's resources are shared with every other computer in the system. Processing power, memory and storage devices are all community resources that authorized users can tap into and leverage for specific tasks.

More precisely a grid:

- integrates and coordinates resources that are not subject to centralized control (that live within different control domains)
- using standard, open, general-purpose protocols and interfaces (that address such fundamental issues such as authentication, authorization, resource discovery, and resource access.)
- to deliver significant qualities of service (with respect to response time, throughput, availability, and security, and/or co-allocation of multiple resource types to meet complex user demands, so that the utility of the combined system is significantly greater than that of the sum of its parts.)

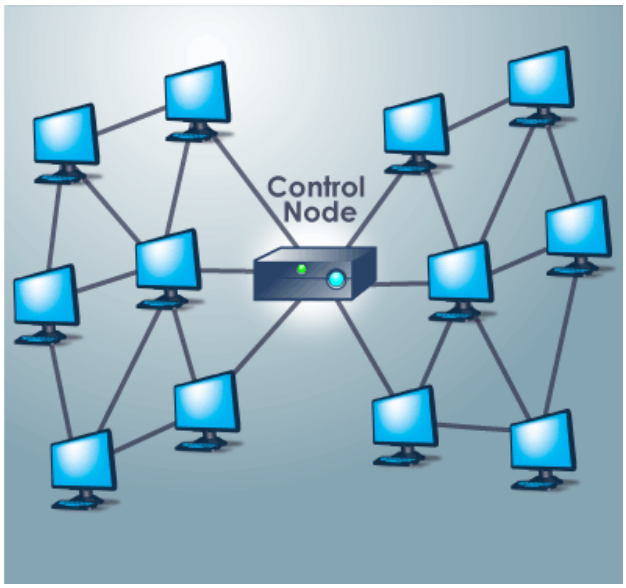

Fig. 1 Example of Grid Computing

A grid computing system can be as simple as a collection of similar computers running on the same operating system or as complex as inter-networked systems comprised of every computer platform you can think of.

This technology, developed since last only one decade, is being used presently, by high energy physicists to analyze data to be produced very soon in LHC (Large Hadron Collider) experiment where Indian scientists are taking part and by earth scientists to monitor Ozone layer activity (deals daily with Data whose volume is equivalent to 150 CDs).

II. REQUIREMENTS OF GRID COMPUTING SYSTEM

In general, a grid computing system requires:

- At least one computer, usually a server, which handles all the administrative duties for the system i.e. a control node. The control node must prioritize and schedule tasks across the network. It's the control node's job to determine what resources each task will be able to access. The control node must also monitor the system to make sure that it doesn't become overloaded. It's also important that each user connected to the network doesn't experience a drop in his or her computer's performance. A grid computing system should tap into unused computer resources without impacting everything else.
- A network of computers running special grid computing network software. These computers act both as a point of interface for the user and as the resources the system will tap into for different applications. Grid computing systems can either include several computers of the same make

running on the same operating system (called a homogeneous system) or a hodgepodge of different computers running on every operating system imaginable (a heterogeneous system). The network can be anything from a hardwired system where every computer connects to the system with physical wires to an open system where computers connect with each other over the Internet.

- A collection of computer software called middleware. Middleware is software that enables communication and management of data in distributed applications. In this more specific sense middleware can be described as "the dash in client-server". The purpose of middleware is to allow different computers to run a processor application across the entire network of machines. Middleware is the workhorse of the grid computing system. Without it, communication across the system would be impossible. Like software in general, there's no single format for middleware.

The middleware and control node of a grid computing system are responsible for keeping the system running smoothly. Together, they divide and farm out pieces of a program to as many as several thousand computers and control how much access each computer has to the network's resources and vice versa. While it's important not to let any one computer dominate the network, it's just as important not to let network applications take up all the resources of any one computer. If the system robs users of computing resources, it's not an efficient system.

III. SECURITY RISKS INVOLVED IN GRID COMPUTING

There are security risks in every application downloaded from the Internet. Whenever you link two or more computers together, you have to prepare yourself for certain questions. How do you keep personal information private? How do you protect the system from malicious hackers? How do you control who can access the system and use its resources? How do you make sure the user doesn't tie up all the system's resources? Thus Security requirements are fundamental to the grid design. The critical problems are resource discovery, authentication, authorization, and access mechanism. Without this functionality, the integrity and confidentiality of the data processed within the grid would be at risk. Let's discuss how authorization and authentication is done in grid system.

IV. SOLUTION TO SECURITY RISKS

A. Authentication

Authentication is the process of verifying identity of a participant to an operation or request. Authentication methods are Password-based, Kerberos authentication, SSL authentication, Certification authorities.

1) Password-based Authentication: Password-based Authentication is a simple function where one party presents a set of credentials (user ID and password combination) to a system. If the credentials match a given set on the system, the system returns a value that represents authorization; otherwise it does not. Some important issues in this are to send unencrypted passwords only when messages can't be read by un-trusted processes while on network, otherwise instead of sending passwords over network one can use password as encryption key. They can encrypt a known but non-repeating value, Send encrypted value to party verifying authentication and both parties must know password or trust a third-party to distribute it.

2) Authentication Systems: Kerberos: Kerberos is a computer network authentication protocol which works on the basis of "tickets" to allow nodes communicating over a non-secure network to prove their identity to one another in a secure manner. Its designers aimed primarily at a client–server model, and it provides mutual authentication—both the user and the server verify each other's identity. Kerberos protocol messages are protected against eavesdropping and replay attacks. Kerberos builds on symmetric key cryptography and requires a trusted third party, and optionally may use public-key cryptography by utilizing asymmetric key cryptography during certain phases of authentication. It is well-suited to frequent authentication, centrally administered, requires trusted, on-line certification authority: Key Distribution Center (KDC)

Authentication process using Kerberos:

- Each client and server registers their keys in advance with Kerberos authentication server.
- Client wants to communicate with service provider: sends client and service provider names to Kerberos authentication server
- Kerberos server randomly generates a session key that will be used for symmetric encryption between client and server
- Kerberos server sends session key to client as well as a ticket that contains client's name and session key, all encrypted with server's key
- Client caches encrypted session key and ticket, which are valid for some period that reduces number of authentication requests to server
- Client forwards ticket to service provider and sends server a timestamp encrypted using the session key
- Server decrypts ticket and extracts session key
- Server uses session key to decrypt timestamp and checks that timestamp is recent
- If client needs to authenticate server, server encrypts the timestamp with the session key and sends it back to client.

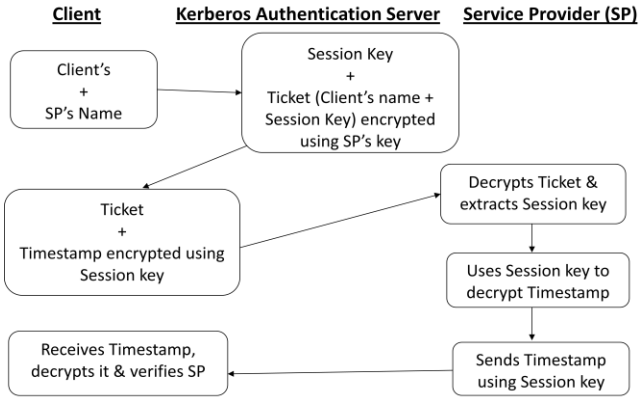

Fig. 2 Authentication process using Kerberos

3) Authentication Systems: Secure Sockets Layer (SSL): Transport Layer Security (TLS) and its predecessor, Secure

Sockets Layer (SSL), are cryptographic protocols that provide communication security over the Internet.TLS and SSL encrypt the segments of network connections above the Transport Layer, using asymmetric cryptography for key exchange, symmetric encryption for privacy, and message authentication codes for message integrity.

Several versions of the protocols are in widespread use in applications such as web browsing, electronic mail, Internet faxing, instant messaging and voice-over-IP (VoIP). Every Client authenticates identity of the server by sending a session key from client to server to set up an encrypted communication. Server has a certificate that contains its public key. If client has a certificate, can authenticate itself to the server. The TLS protocol allows client-server applications to communicate across a network in a way designed to prevent eavesdropping and tampering.

Once the client and server have decided to use TLS they negotiate a stateful connection by using a handshaking procedure. During this handshake, the client and server agree on various parameters used to establish the connection's security.

Handshaking procedure between client and server using SSL is as follows:

- The handshake begins when a client connects to a TLS-enabled server requesting a secure connection and presents a list of supported cipher suites (ciphers and hash functions).
- From this list, the server picks the strongest cipher and hash function that it also supports and notifies the client of the decision.
- The server sends back its identification in the form of a digital certificate. The certificate usually contains the server name, the trusted certificate authority (CA) and the server's public encryption key.
- The client may contact the server that issued the certificate (the trusted CA as above) and confirm the validity of the certificate before proceeding.
- In order to generate the session keys used for the secure connection, the client encrypts a random number with the server's public key and sends the result to the server. Only the server should be able to decrypt it, with its private key. From the random number, both parties generate key material for encryption and decryption.

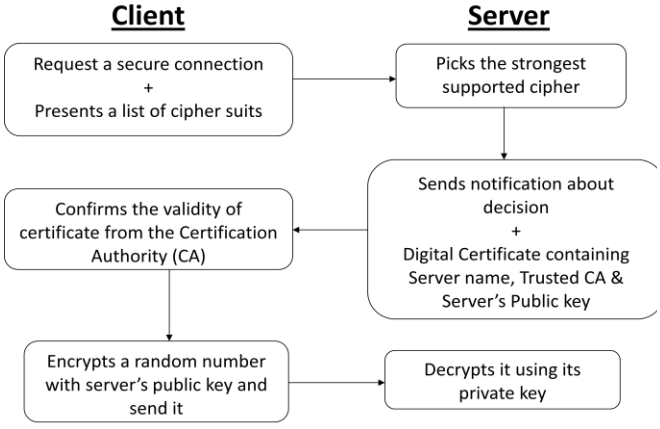

Fig. 3 Handshaking procedure between client and server using SSL

This concludes the handshake and begins the secured connection, which is encrypted and decrypted with the key material until the connection closes. If any one of the above steps fails, the TLS handshake fails and the connection is not created.

4) Digital Certificates and Certification Authorities (CA): If a grid resource needs to securely communicate with another grid resource, it needs a certificate signed by a CA. Digital certificates are digital documents that associate a grid resource with its specific public key. A certificate is a data structure containing public key and pertinent details about the key owner. A certificate is considered to be a tamper-proof electronic ID when it is signed by the Certification Authority for the grid environment. Certificates do not normally contain any confidential information, and their free distribution does not create a security risk. The technical implementation is such that it is considered extremely difficult to alter any part of a certificate without easy detection. The signature of the CA provides an integrity check for the digital certificate.

Obtaining a client or a server certificate from a CA involves the following steps:

- The grid user requiring certification generates a key pair (private key and certificate request containing the public key). When a grid client wants to start a session with a grid recipient, he or she does not attach the public key to the message, but the certificate instead
- The user signs its own public key and any other information required by the CA. Signing the public key demonstrates that the user does, in fact, hold the private key corresponding to the public key.
- The signed information is communicated to the CA. The private key remains with the client and should be stored securely. For instance, the private key could be stored in an encrypted form on a Smartcard, or on the user's computer.
- The CA verifies that the user owns the private key of the public key presented.
- The CA (or optionally an RA) needs to verify the user's identity. This can be done using out-of-band methods, for example, through the use of e-mail, telephone, or face-to-face communication. A CA (or RA) can use its own record system or another organization's record system to verify the user's identity.
- Upon a positive identity check, the CA creates a certificate by signing the public key of the user, thereby associating a user to a public key. The certificate will be forwarded to the RA for distribution to the user.
- The recipient receives the communication with the certificate and then checks the signature of the Certificate Authority within the certificate. If the signature was signed by a certifier that he or she trusts, the recipient can safely accept that the public key contained in the certificate is really from the sender. This prevents someone from using a fraudulent public key to impersonate the public key owner.

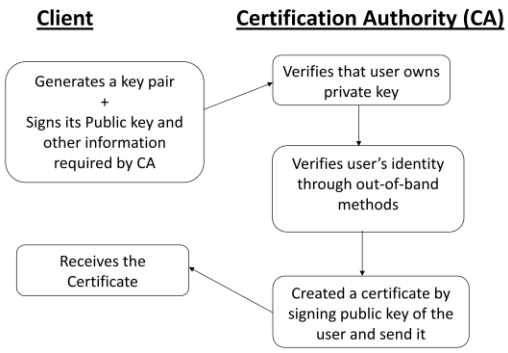

Fig. 5 Process of obtaining a certificate

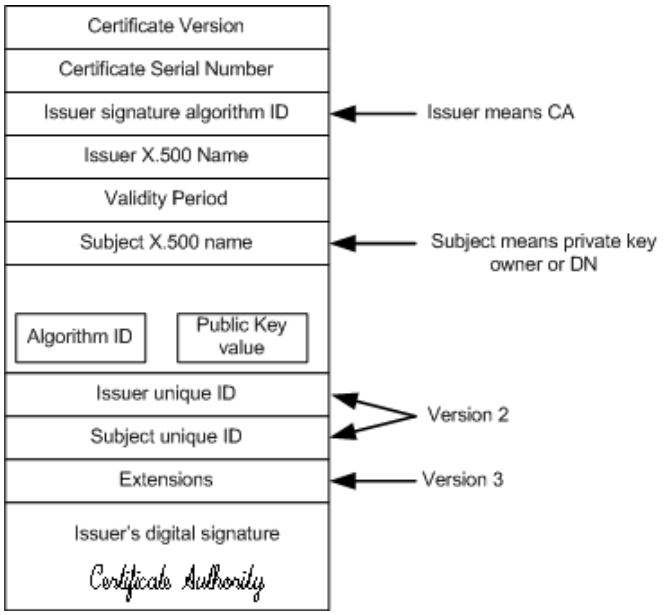

Fig. 4 Graphical depiction of the digital certificate

Certificate can be compared to the passport that serves as an authentication mechanism when this individual travels to foreign countries. Just like passports, digital certificates can subsequently be used for authenticating subjects to other parties that require authentication.

In the grid computing area, the researchers and practitioners have come together to create the Global Grid Forum (GGF) (now called OGF). They have released an open standard called Open Grid Standards Architecture (OGSA). There is a Grid Security Infrastructure (GSI) layer of OGSA which addresses most of the information security challenges mentioned above. A central concept in GSI authentication is the certificate. Every user and service on the grid is identified via a certificate, which contains information vital to identifying and authenticating the user or service.

B. Authorization

Another important security issue is that of authorization. Authorization is the process that determines whether a particular operation is allowed. Authorization is important to limit access for security reasons and also to allow only certain users to access the full capabilities of the network to avoid deadlock by flooding control node with processing requests. Like any resource sharing system, grid systems also require resource specific and system specific authorizations. That is why the authorization systems can be mainly divided into two categories: VO Level Systems and Resource Level Systems.

1) VO Level Systems: A virtual organization (VO) is defined as a dynamic group of individuals, groups, or organizations who define the conditions and rules (business objectives and policies) for sharing resources.

VO level grid authorization systems are centralized authorization for an entire Virtual Organization (VO). These types of systems are necessitated by the presence of a VO which has a set of users, and several Resource Providers (RP) who own the resources to be used by the users of the VO. Whenever a user wants to access certain resources owned by a RP, he/she obtains a credential from the authorization system which allows certain rights to the users. The user presents the credentials to the resource to gain access to the resource. In this type of systems, the resources hold the final right in allowing or denying the access to the users. Examples of VO level grid authorization systems are Community Authorization Service (CAS) Virtual Organization Membership Service (VOMS), and Enterprise Authorization and Licensing System (EALS).

2) Resource Level Systems: Unlike the VO level authorization systems, which provide a consolidated authorization service for the virtual organization, the resource level authorization systems implement the decision to authorize the access to a set of resources. Therefore, VO level and resource level authorization systems look at two different aspects of the grid authorization. Different resource level authorization Systems are Akenti, Privilege and Role Management Infrastructure Standards Validation (PERMIS), and the GridMap system.

C. Integrity and Confidentiality

There is a need to protect data during transmission on network because anyone connected to an open network may observe, insert or possibly remove message. In a grid computing environment where risk is high, one must ensure integrity and confidentiality of the data being transmitted.

Here are some techniques used for creating secure grids as follows:

1) Symmetric key encryption: Symmetric key encryption is based on the use of one shared secret key to perform both the encryption and decryption of data. To ensure that the data is only read by the two parties (sender and receiver), the key has to be distributed securely between the two parties and no others. This form of encryption has performance benefits over asymmetric encryption, but requires additional care and administration in the handling of the shared key. Data Encryption Standard (DES), Advanced Encryption Standard (AES), Triple-DES, RC2 and RC4 are some examples of symmetric key cryptographies.

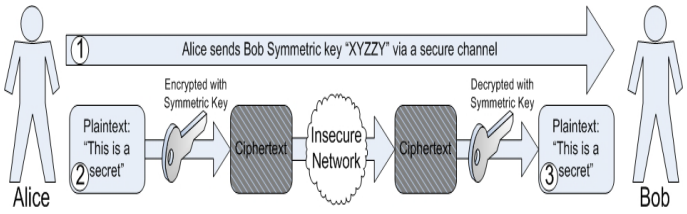

Figure 6 Symmetric key encryption using a shared secret key

The security of the exchange relies on the security of the symmetric key. If an attacker intercepts the symmetric key, the attacker can read the cipher text and he can create new cipher text.

2) Asymmetric key encryption: In public key cryptography, the entities generate public/private key pairs based on some cryptographically secure mathematical function. A message when encrypted by the public key can only be decrypted by the private key corresponding to the public key. The public keys are known to everyone. The asymmetric key pair is generated by a computation that starts by finding two very large prime numbers. Even though the public key is widely distributed, it is practically impossible for computers to calculate the private key from the public key. The security is derived from the fact that it is very difficult to factor numbers exceeding hundreds of digits. This mathematical algorithm improves security, but requires a long encryption time, especially for large amounts of data. For this reason, public key encryption is often used to securely transmit a symmetric encryption key between the two parties, and all further encryption is performed using this symmetric key.

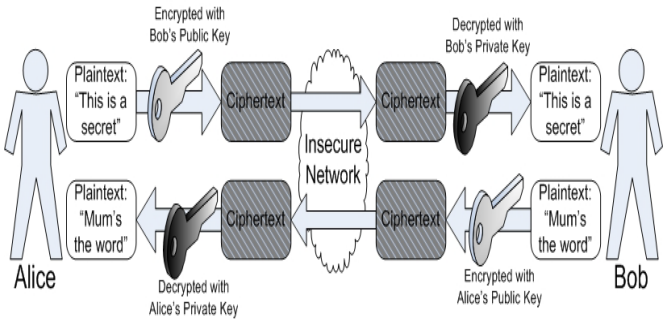

Figure 7 Asymmetric key encryption

Asymmetric key encryption also provides authentication. Only Alice has access to her secret key, so if Bob is able to decrypt a message with Alice's public key, he has assurance that Alice is the author.

V. CONCLUSION

Grid computing appears to be a promising trend for three reasons: (1) its ability to make more cost-effective use of a given amount of computer resources, (2) as a way to solve problems that can't be approached without an enormous amount of computing power, and (3) because it suggests that the resources of many computers can be cooperatively and perhaps synergistically harnessed and managed as a collaboration toward a common objective. When building any new environment or implementing a new software application. But while designing a grid, security checks should be performed. These checks will help determine how these new changes will affect the overall security of the environment and any other areas of change. Only effective security implementation in grid would ensure the reliability on grid computing.

REFERENCES

[1] I. Foster,(2002) What is the Grid? On dlib [Online]. Available http://dlib.cs.odu.edu/WhatIsTheGrid.pdf
[2] J. Strickland, How Grid Computing Works on HowStuffWorks. [Online]. Available: http://computer.howstuffworks.com/grid-computing.htm
[3] Middleware on Wikipedia [Online]. Available http://en.wikipedia.org/wiki/Middleware
[4] (2001) grid computing on Searchdatacentre.techtarget [Online]. Available http://searchdatacenter.techtarget.com/definition/grid-computing
[5] Taxonomy of Grid Security Issues on Infosys [online]. Available http://www.infosys.com/infosys-labs/publications/Documents/grid-computing-security.pdf
[6] B. Jacob, M. Brown, K. Fukui, N. Trivedi, (2005) Introduction to Grid Computing on IBM [Online]. Available http://www.redbooks.ibm.com/redbooks/pdfs/sg246778.pdf
[7] V. Welch1, F. Siebenlist, I. Foster, J. Bresnahan, K. Czajkowski, J. Gawor, C. Kesselman, S. Meder, L. Pearlman, S. Tuecke, Security for Grid Services on Globus [Online]. Available http://www.globus.org/alliance/publications/papers/GT3-Security-HPDC.pdf
[8] A. Chakrabarty, Taxonomy of Grid Security Issues on Springer [Online]. Available http://www.springer.com/computer/communication+networks/book/978-3-540-44492-3
[9] Kerberos (protocol) on Wikipedia [Online]. Available http://en.wikipedia.org/wiki/Kerberos_%28protocol%29
[10] (2010) Transport Layer Security on Wikipedia [Online]. Available http://en.wikipedia.org/wiki/Transport_Layer_Security
[11] E. Conrad, Explanation of the Three Types of Cryptosystems on giac [online]. Available: http://www.giac.org/cissp-papers/52.pdf

Author
Name: Rashmi Bhatia
Qualification: MCA
Asst. Prof., Dev Samaj College For Women, Chandigarh
Email: rashmibhatia@outlook.com

Correspondence Author
Rashmi Bhatia
rashmibhatia@outlook.com
9815863842