A Framework for Distribution of Load on the Database Server from Secure Network Users, Intranet and Internet Users and Ensure Network and Database Security

Mohammad Ghulam Ali, *Member, IACSIT*

***Abstract*—In this paper, we propose a framework to workout how to distribute load of the master (primary) database server where transactions are committed through a Web-based or Client-Server DBMS from a private and secure network, where information are accessed through Web clients either from a secure and private network, from the intranet or from the internet. We also consider an issue to ensure high level network and database security to the master database server. We implement and maintain private and secure network within the premises of the Institution / Organization intranet through a software firewall and we implement and maintain database replications from the secure network to the intranet. Since, actual transactions are committed in the master database server from private users of the secure network, all committed transactions are then replicated to the slave database server which is kept in the Organization intranet. All committed transactions are forwarded from the secure network to the intranet through a software firewall. Since the master database in the private and secure network is replicated to the slave in the intranet and access is provided to the slave, load of the master database server is distributed between private users and the users from intranet and internet. Load on the database server is distributed between master and slave database.**

***Index Terms*—Database, Database Management System, Software Firewall, Database Replication, Database and Network Security, Information Security.**

I. Introduction

Keeping in mind the progress in communication and database technologies (concurrency, consistency and reliability) has increased the data processing potential. Various protocols are proposed and implemented for network reliability, concurrency, atomicity, consistency, recovery and replication. Various methods are proposed for the database and network security. The current demand is how to make actual database very secure from authorized and unauthorized users, how to distribute load of the database server so that access to database is faster and provide high throughput.

We explain few in this paper about some existing approaches related to the load balancing high transaction databases. The work [26] shows cluster of database servers connected to the Broker Hub. Each database was configured to handle a set of users with specific user Ids and thus held data only for those users. Merge replication was used to replicate data from all databases to a central database which was used for all reporting. There is another approach *High Availability and Load Balancing* where database servers can work together to allow a second server to take over quickly if the primary server fails (high availability), or to allow several computers to serve the same data (load balancing). Ideally, database servers could work together seamlessly.

Ensuring network and database security means we ensure our database is secured from unauthorized access and our database is secured from malicious attempts that can steal (view) and modify our sensitive data. We mean to say sensitive data such as bank transactions, credit card transactions, school and university degree grades, defense data etc.

Database security is a growing concern evidenced by an increase in the number of reported incidents of loss of or unauthorized exposure to sensitive data. As the amount of data collected, retained and shared electronically expands, so does the need to understand database security. The Defense Information Systems Agency of the US Department of Defense (2004), in its *Database Security Technical Implementation Guide*, states that database security should provide controlled, protected access to the contents of a database as well as preserve the integrity, consistency, and overall quality of the data. Students in the computing disciplines must develop an understanding of the issues and challenges related to database security and must be able to identify possible solutions [25].

The role of Database Security Manager (DSM) and Database Administrator (DBA) is to maintain and secure sensitive data within an organization.

In this paper we discuss about levels of security, we discuss about three zones of the network and type of corresponding users, we propose a viable framework to distribute load of the master database server and to enforce high level network and database security, we discuss about software firewall that segregate sensitive areas of the organization / institution from other users, we discuss about database replications between private and secure network and the intranet, we discuss about application server, we discuss about organization / institution gateway and finally we illustrate experimental proof.

A. Levels of Security

We have conceived following levels of security:

A.1. Database Level Security

The increased usage of databases to store large amounts of data has created new security problems. Typically a database contains data of various degrees of importance and

Manuscript received May 30, 2011; revised October 10, 2011.

Mohammad Ghulam Ali, System Engineer, Academic Post Graduate Studies & Research, Indian, Institute of Technology Kharagpur, Kharagpur - 721 302, West Bengal, INDIA. (e-mail: ali@hijli.iitkgp.ernet.in, ali_iit@yahoo.com), (Phone: 91-3222-282056, 282188, 220176)

levels of sensitivity. This data is shared among a wide variety of users with different responsibilities and privileges. It is therefore necessary to restrict users of the database to those portions of the total data that are necessary for their activities. Additionally, more control is needed over changes a user can make to data because of the many ways these changes can affect other users of the database [1]. The DBA at Database Server will provide a better database server level and database object level security. We do not explain in details in this paper about database server and object level security implementation. Please see the details [14,15,20] of the database server and object level security implementation.

A.2. Operating System Level Security

A System Administrator at Database Server will provide a better OS level security. We do not explain in details in this paper about operating system level security implementation. Please see the details [13,22] of the operating system level security implementation

A.3. Network Level Security

A Network Security expert can better protect Database Server by implementing a Software Firewall between the Intranet of the Institution / Organization and the Secure and Private Network and will examine each incoming packets coming to the Master Database Server in the private and secure network from authorized users or from unauthorized users of the intranet and the internet, will examine each outgoing packets from the Secure and Private Network to the intranet, to the internet and to the Slave Database Server and will decide whether packets are to be denied, dropped or forwarded. A firewall policy rules are to be written based on our requirements. A Linux-based server which is used for a software firewall will have two network interface cards where 1st network card will be connected to the intranet and will have intranet IP and the second network card will be connected to the private and secure network and will have private IP. We have proposed a viable framework for network level security implementation in section II and experimental proof in section III. Please see also the details [9,10,11,12,16,17,18,19,21,22,23,24] of the network level security implementation. We have proposed a viable framework for master to slave database replication in section II and experimental proof in section III. This is a most viable framework that can be used by any organization / institution.

A.4. Database Replication – considered as database security

Replication enables data from database server (the master) to be replicated to one or more database servers (the slaves). Replication is asynchronous – slaves need not to be connected permanently to receive updates from the master. Replication features support for one-way, asynchronous replication, in which one server acts as the master, while one or more other servers act as slaves. We illustrate database replication in details in section I.A.4, II and III. Please see the details [2,3,4,5,6,7,8] for database replication. We have shown experimental proof of Master-Slave Replications of the database in Section III.

In our proposed framework, if master database is replicated to the slave and if we provide intranet and internet users to access to slave then master database is protected from intranet and internet users and also hidden from authorized and unauthorized users of the intranet and the internet. Hence the fact, we can say replication also ensures database security.

B. Three Zones of the Network and type of Corresponding Users

In our proposed framework, there are three zones of network and corresponding users.

B.1. Private and Secure Zone

All users in the private zone have private IPs. There role is to maintain the database. They commit transactions. They are also in the premises of the Initiation / Organization network.

B.2. Intranet Zone

All users in the intranet zone of the Institution / Organization have either internet IPs or IP as provided by the Institution / Organization. They only access to Slave Database Server through an application server using a web-based interface.

B.3. Internet Zone

All users from internet zone or we can say from across the world can access to the Slave Database Server passing through the Organization / Institution Gateway and through an application server and the web-based interface.

C. Software Firewall

A firewall is a conceptual component in network security. A firewall provides the means for implementing and enforcing a network access policy. In effect, firewall provides access control to users and services. A firewall can greatly improve network security and reduce risks to hosts protected on the subnet by filtering inherently insecure services. If all access to and from between two networks passes through a firewall, the firewall can log accesses and provide valuable statistics about network usage. Firewall addresses the issues of data integrity, confidentiality and authentication of data that is behind the firewall [23].

We explain in details about software firewall rules implementation in section III.B Please see the details [9,10,11,21,23,24] of software firewall implementation

D. Database Replications and the Replication Server

There is a replication of each transaction from the Master Database Server to the Slave Database Server. Actual transactions are committed in the Master Database Server from all private and secure users. Then all committed transactions are replicated to the Slave Database server. The Slave Database Server is also called as Replicated Server.

E. High Level Network and Database Security

A Software Firewall and replication of Master Database Server ensures us high level network and database security to the master database server.

F. Application Server

Application Server hosts API and web-based software, and running a Web Server.

G. Organization / Institution Gateway

Any packet coming to the Institution / Organization will pass through the Institution / Organization Gateway, will reach to the router and finally router will forward to the destination IP or we can also say as gateway to router, router to switch and switch to destination IP.

II. A Proposed Framework

In our proposed framework, there is an intermediate server between intranet and the private and secure network. This is a Linux-based server with two Network Interface Cards. The first card will work as gateway for the private and secure network with private IP and the second network card will work as gateway for the intranet of the Institution / Organization with internet IP or IP as allocated from the Institution / Organization. This gateway is different from the Institution / Organization Gateway. A standard firewall rules are to be written for authentication of all incoming packets coming from the intranet and from the internet to the master database server and all outgoing packets coming from the private and secure network to the intranet of the Institution / Organization, to the internet and to the slave database server. We can also say that a firewall is a secure and trusted machine that sits between the private and secure network and the intranet of the Institution / Organization. The firewall machine is configured with a set of rules that determine which network traffic will be allowed to pass and which will be blocked or refused.

In the proposed framework as shown in Figure 1 the master database server is kept in the private and secure network. All users' machines have private IPs. They are accessing to the master database server through a web-based interface or through a Client-Server DBMS and committing transactions to the master database and maintaining the master database and master database catalog. As soon as transaction is committed, the committed transaction is logged into a binary log of the database server and this is done by the **client thread** that executed the query that modified the database and the same transaction is forwarded through software firewall (a packet filtering firewall) to the slave database server which is kept in the intranet of the institution/organization. The binary log is relayed to the slave database server, where it is logged into a relay log and the same transaction is processed there and slave is updated, slave is maintained and catalog of the slave database is also maintained. Hence, we get identical image of the master database as the slave database. Now slave database is available for access from the intranet or from the internet users through a web-based application server. Intranet and Internet users have no knowledge of the master database as it is kept in the private and secure network. All outgoing packets from the private and secure network to the intranet, to the internet and to the slave database server will be examined at the Software Firewall and software firewall policy will decide whether packets are to be dropped, denied or forwarded. Similarly, all incoming packets to the Master Database Server from intranet and internet will be examined at the Software Firewall and the Firewall policy will decide whether packets are to be dropped, denied or forwarded to the Master Database Server. In our proposed framework only the **I/O thread** related packets of the slave database will be forwarded to the Master Database Server as slave connects to the master requests a copy of the binary log and hence master will relay binary log to the slave and therefore packets related to the relay log will be forwarded to the Slave Database Server. All other incoming and outgoing packets will be straight forward dropped. The master binary log is written to local relay log on the slave before it is processed. In practical, slave pulls the data from the master, rather than the master pushing the data to the slave. Master will have user account as **slave_user** with the IP address of the slave database server and **repl_slave** privileges. User with **repl_slave** priviledge in the master database enables slave to connect to the master to request updates that have been made to database on the master server. The relay log on slave consists of the events read from the binary log of the master and written by the **slave I/O thread**. Events in the relay log are executed on the slave as part of the **SQL thread**.

In our proposed framework, to build a packet filtering firewall under Linux, no special software is required. Linux ipchains / iptables and IP forwarding are used to configure Linux as a Firewall and Router. **ipfwadm** was used in Linux Kernel Version 2.0.x and Red Hat Version 5.x. **ipchains** was used in Linux Kernel Version 2.2.x and Red Hat Version 6.x, 7.0. **iptables** is using in Linux Kernel Version 2.4.x, 2.6.x and in the later version and Red Hat Version 7.1 - 9.0, Fedora 1,2,3 and in the later version. To build a Linux IP firewall, it is necessary to have a kernel built with IP firewall support and the appropriate configuration utility. For more information on how to compile the kernel refer to [10].

In our proposed framework, we have considered iptables packet filtering firewall so called IP-filters. iptables served as firewall software. The firewall rules which inspect incoming IP-packages and decide on different criteria how to deal with that packet. iptables offers different criteria to select packages for further processing. The more important ones are: **protocol**: if it is TCP, UDP, ICMP, etc. package, **socket number** (for TCP/UDP), **incoming interface**: from which network the package is arriving, **outgoing interface**: to which network the package is supposed to be sent, **source IP-address**: where the package originates from, **destination IP-address**: where the package is suppose to be sent. A packet filtering firewalls is normally implemented within the operating system and screens packets based at the IP network layer. It protects the system by making decisions after filtering packets based on information in the IP packet header. A firewall rule has a default Packet Filtering policy. If a packet doesn't match any rule, the default policy is applied. There are two basic approaches: Deny everything by default and explicitly allow selected packets, allow everything by default and explicitly deny selected packets. We have considered the first approach.

The kernel starts with three lists of rules; these lists are called firewall chains. The three chains are called input, output and forward. When a packet comes in the kernel uses the input chain to decide its fate. If it survives that step, then the kernel decides where to send the packet next. If it is destined for another machine, it consults the forward chains. Finally, just before a packet is to go out, the kernel consults the output chain.

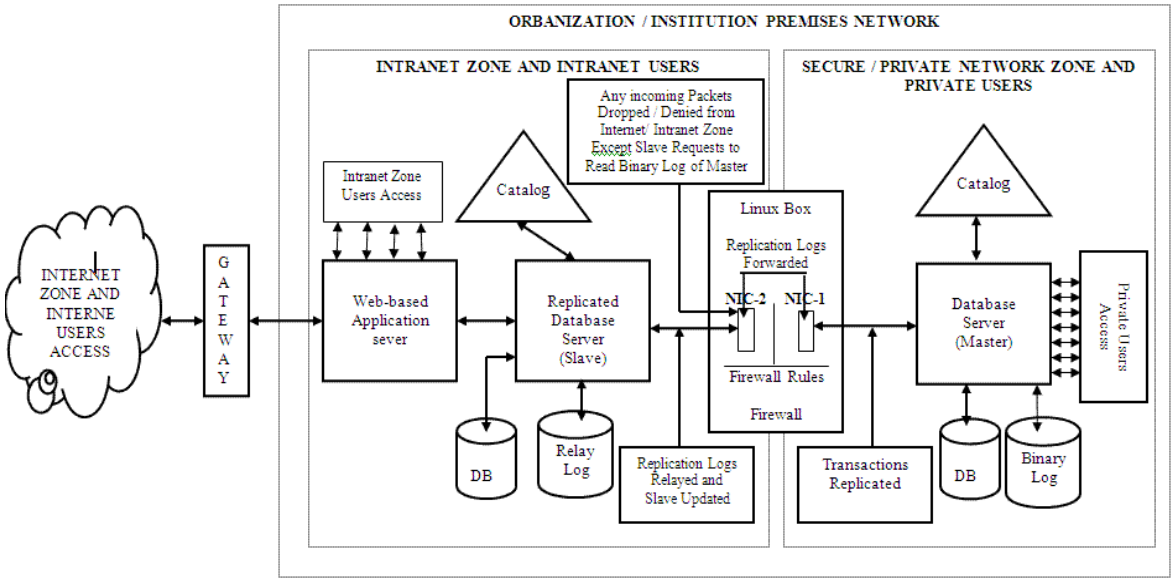

Fig.1. Aproposed framework

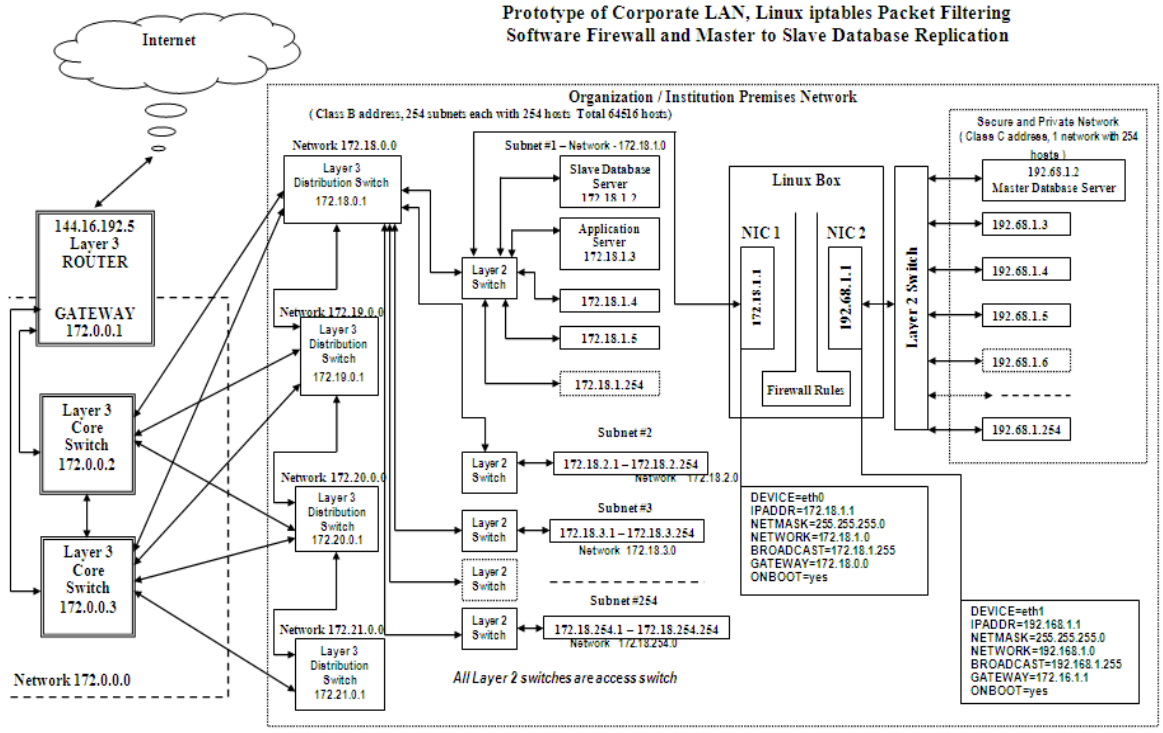

Fig. 2. A prototype of a corporate LAN, iptables packet filtering firewall and master to slave database replication

All IP packet headers contain the source and destination IP addresses and type of IP protocol message, (ICMP, UDP and TCP) as we discussed above. Beyond this, a packet header contains slightly different fields depending on the protocol type. ICMP packets contain a type field identifying the control or status message, along with a second code field for defining the message more specifically. UDP and TCP packets contain source and destination service port numbers. TCP packets contain additional information about the state of the connection and unique identifiers for each packet [24]. We finally decide Firewall Rules as stated in section III.B. Please see details [9,10,11,21,23,24] for writing and implementing firewall policy rules.

We extend the proposed framework by showing a prototype of a corporate LAN, implementing a Linux-based iptables packet filtering firewall within the same corporate LAN, implementing Master to Slave Database Replication from a private and secure network to the Corporate LAN, providing a high level network and database security and finally distributing load of the master database server as it is shown in figure 2. This will give an illustration of the framework in more meaningful manner.

The corporate LAN is based on class B addressing. We have shown four distribution switches and each distribution will have Subnet #1 to Subnet #254 and each Subnet with 254 hosts. The Subnet Mask for each Subnet is 255.255.255.0. We have also shown router/gateway and two core switches based on Virtual Router Redundancy protocol (VRRP). We do not go in-depth in this topic in this paper. Similarly, the private and secure network has class C address with 1 network with 254 hosts and the Subnet Mask is 255.255.255.0. We have not discussed about role of the proxy server(s) and DNS Server(s) in this figure.

We are sure that our proposed framework will certainly provide high level database and network security, will distribute load of the master database server from the private and secure users, intranet users and the internet users by implementing database replications and the packets filtering software firewall.

III. Experimental Proof of Master to Slave Database Replication

We explain the experimental proof of the Master to Slave Database Replications and evaluate the efficiency of the proposed framework.

A. Master and Slave Database Server Specifications

In our experiment we have used MySQL 4.1.7 Database Server for Master and Slave in two different Red Hat Enterprise Linux 3.4.3-9 and Kernel 2.6.9-5. The Master Database Server configuration is Intel(R) Xeon(TM) CPU 3.00GHz, CPU MHz is 2993.592 and Cache Size is 2048 KB and the Slave Database Server Configuration is Intel(R) Pentium(R) D CPU 3.00GHz, CPU MHz is 3001.387 and Cache Size is 2048 KB. In figure – 2 Master Database Server has IP 192.68.1.2 and Slave Database Server has IP 172.18.1.2. How MySQL replication works, we have shown in figure 3.

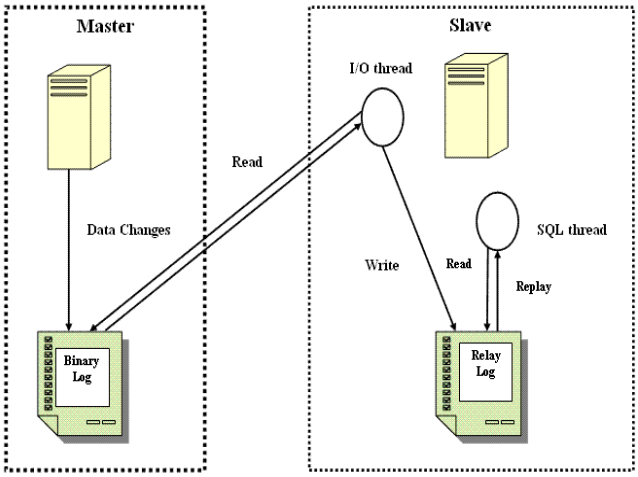

Fig. 3. How mySQL replication works

We categorize replication in three processes:

The master records changes to its data in its binary log. (These records are called binary log events)

The slave copies the master's binary log events to its relay log.

The slave replays the events in the relay log, applying the changes to its own data.

B. Firewall Server Specification and Firewall Policy

In between this two computer servers we have proposed to put a Linux based computer machine with two network interface cards (NIC1, NIC2) with system configuration is Intel(R) Pentium(R) D CPU 3.00GHz, CPU MHz is 3001.387 and Cache Size is 2048 KB which will act as iptables packet filtering software firewall. The IP address of the NIC2 is 192.68.1.1 and the IP address of the NIC1 is 172.18.1.1. Two network interface cards are configured properly and tested. NIC1 is connected to the Intranet of the Institution/Organization and NIC2 is connected to the private and secure network. Requests are originated through a slave I/O thread from the slave (172.18.1.2) via TCP/IP from port 3306 to connect to the master (192.68.1.2) on port 3306 to read binary logs and to store those logs in the relay log of the slave and finally all events in the relay log are executed in the slave with the help of the SQL thread and slave is updated. Therefore, we consider following rules in writing a firewall policy:

The source IP Address, Port Number and Protocol is known

The destination IP address, Port Number and Protocol is known.

I/O thread related packets of the slave database will be forwarded to the Master Database Server.

Master will relay binary log to the slave and therefore packets related to the relay log will be forwarded to the Slave Database Server.

All other incoming and outgoing packets are straight forward dropped.

Hence we can very easily write and implement firewall rules in this scenario.

C. Master and Slave Database Server Configuration

We create user account "slave_user" in the Master Database Server having IP address 192.68.1.2 with host name 172.18.1.2 which is the IP Address of the Slave Database Server and giving "repl_slave_priv**"** privilege**.** We replicate database named ResearchDB from the Master Database Server to the identical database ResearchDB in the Slave Database Server. We further configure MySQL configuration file "my.cnf" in both Master and Slave as per our requirements:

In the Master Database Server (additional parameters)

……...
server-id = 1
log-bin = /var/log/mysql/bin.log
replicate-do-db = ResearchDB
binlog-do-db = ResearchDB
binlog-ignore-db = mysql,test, other databases
master-port = 3306
……

In the Slave Database Server (additional parameters)

……
server-id = 2
master-port = 3306
master-host = 192.68.1.2
master-user = slave_user
master-password = password of the slave_user
relay-log = /var/log/mysql/relay.log
relay-log-info = /var/log/mysql/relay-log.info
master-info-file = /var/lib/mysql/mysql-master.info
……

Then we restart both servers. We check the master status to get initial binary log file number and log position. We stop the slave and change the parameters that the slave server uses for connecting to the master server, for reading the master binary log, and reading the slave relay log. This includes initialization of master log file and master log position as we get from master status. It also updates the contents of the master.info and relay-log.info files. Finally

we start the slave to start replications. We have observed there is no delay in data replications. If due to any reason slave is not running and transactions are on in the master, when slave restores all backlogs will be relayed and will be committed in the slave. Hence there is no room for database inconsistency. Slave will have always identical database as of master.

IV. Conclusion

In this paper we have proposed a robust framework for distributing load of the master database through database replications of the master database from the private and secure network to the database which is running in the Institution / organization intranet and making master database more secure and this will be done with the help of a software firewall. We shall address other issues in future on database and network security and database high throughput.

References

[1] E. Bertino, and L. M. Haas, "Views and Security in Distributed Database Management Systems".
[2] MySQL Replication extract from the MySQL 5.0 Reference Manual.
[3] L. Thalmann, M. Kindahl, "MySQL Replication Architecture".
[4] Sybase Heterogeneous Replication Guide 15.0.
[5] Sybase Replication Server Design Guide 11.5.
[6] Sybase Replication server 12.0.
[7] Sybase Replication Server Administration Guide 12.0.
[8] Configuring SQL Server 2005 Peer-to-Peer Replication.
[9] P. Harrison, "Linux Home Networking".
[10] M. Grennan, "Firewall and Proxy server HOWTO".
[11] M. Hamm"Firewall".
[12] K. Seifried, "Linux Administrators Security Guide".
[13] P. Cobbaut, "Linux System Admin".
[14] Sybase ASE System Administration Guide 12.
[15] MySQl Reference Manual for Version 5.0.3-alpha.
[16] Linux Network Administrators Guide.
[17] K. Fenzi, "Linux Security HOWTO".
[18] M. Sweeney, "Network Security Using Linux".
[19] D. Wreski, "Linux Security Administrator's Guide".
[20] Adaptive Server Anywhere Database Administration Guide
[21] G. Green , J. George "Linux Security".
[22] G. Mourani, "Linux Secure and Optimized Server-A guide for information system, configuration, optimization and network security professionals."
[23] Y. G. Beyene, "Firewall in Linux: Principles and Implementation"
[24] R.L. Ziegler, "Linux Firewalls, New Riders, Indiana, 470pp".
[25] Meg Coffin Murray, **"**Database Security: What Students Need to Know**",** Kennesaw State University, Kennesaw, GA, USA, Volume 9, 2010.
[26] Nagarro, "Case study: load balancing high transaction volume databases", June 2, 2010

Mohammad G. Ali He was borne in Bhagalpur, Bihar India. His date of birth is January 27, 1968.

He obtained the degree of Master Diploma in Computer Science (1991) and Master of Science in Mathematics (1993) with 1st class. He stood 1st in the Computer Science in the University.

He is a Fellow (FBCS), British Computer Society, the Chartered Institute for IT, UK. He is a life member of IAENG, Hong Kong and IACSIT, Singapore.

His two papers were published in the International Journal of Computer Science and Information Security, USA in the month of November 2009. Another paper was published in the Global Journal of Computer Science and Technology, USA in the month of April, 2010. Another paper was accepted in the International Conference, IASTED, ACIT-ICT 2010, Russia. Another paper was published in International Journal of Computer Applications, Foundation of Computer Science, New York, USA in the month of September 2010. Another paper is published in the International Journal of Computer and Electrical Engineering (IJCEE), International Association of Computer Science and Information Technology, Singapore. Another paper is accepted for publication in the International Journal of Computer Theory and Engineering (IJCTE), International Association of Computer Science and Information Technology, Singapore. His one paper was accepted in the international conference, (ICMLC-2011), Singapore which was held in Feb 26-28, 2011 (The conference was sponsored by the IACSIT, IEEE, IE).

He is a member of the Editorial Board of IJCA, USA and IJCTE, Singapore. He is a member of Reviewer Board of IAENG International Journal of Computer Science, Hong Kong. He was a Peer Reviewer of the International Conferences, ICMLC-2011, Singapore and IEEE ICCSIT 2011, China.

He is a System Engineer Grade I in the Indian Institute of Technology, Kharagpur, West Bengal, India. He is associated with IT project Management, System Analysis and Design Methods, System Development Life Cycle, Programming, Implementation and Maintenance of Client-Server DBMSs and Web Applications Development. He is also associated with Database Administration, Web Server Administration, System Administration and Networking of the Institute. He has deployed many small to big projects in the Institute Network. He has been guiding undergraduate and post graduate students of the Institute in their projects.

His areas of research are Parallel and Distributed Computing (Heterogeneous Distributed Databases), Software Engineering, Networking and Network Security.