

# Forward Engineering based Implementation of TOS in Social Networking

Nitish Pathak  
Research Scholar and  
Corresponding Author, UTU,  
Dehradun (INDIA)

Girish Sharma  
Department of CS&E, BPIBS,  
Govt. of NCT, Delhi (INDIA)

B. M. Singh  
Department of CS&E,  
College of Engineering  
Roorkee (INDIA)

## ABSTRACT

The present generation of youth begins the day with the Facebook or other social website. Hundreds of millions of people all over the world make use of social websites, Internet portals, blogs, Wikis, etc. These sites such as MySpace, Facebook and YouTube have the essential features and equipped with the necessary computing facilities to keep gigantic online communities get going with secure manner.

Due to rapid growth of networking, use of social networking sites in day to day life, data sharing, computer security has made a vital part of computer research & development. For maintaining the security in various applications like E-commerce Online goods and services, Banking, Marketplace services Advertising, Auctions, Comparison shopping, Mobile commerce Payment, Ticketing, An electronic payment system (EPS), Online insurance policy management, we have to use high secured operating systems. In this regard a number of extremely secure operating systems i.e. Trusted Operating Systems like SELinux, Argus, Trusted Solaris, Virtual Vault have been developed by companies such as Argus-Systems Group, Hewlett-Packard, and Sun Microsystems to handle the increasing need of security. Normally, due to high security reason these operating systems are being used in defense. But still these secure operating systems have limited scope in commercial sector and are not popular in corporate due to lower performance; actually this security will come at a cost.

In this paper we will propose SPF Model to maintain the balance between security and performance for these operating systems. This SPF model of TOS can be implement for various applications. For implementation of these SPF based trusted operating system we propose object oriented based Code generation i.e. forward engineering i.e. process of generating source code from one or more OO Rational Rose model for web application like social networking. In this research paper we will discuss the issues and UML-based software development solutions for SPF to manage the security, performance and modeling for Social networking sites.

## Keywords

**SPF**- Security Performance Flexibility, Software architecture, UML, Code generation, **SELinux**- Security-Enhanced Linux, OO Rose Mode-Object Oriented Rose Model, **TOS** -Trusted Operating System

## 1. INTRODUCTION

Social Networks are the common platform for the present generations to share ideas. The members of these sites have grown to billions in the last decade and much more at present.

Social Websites are supporting the exchange of thoughts and opinions through interaction [1] among the users. They became

the windows for all marketing, sales, services and information exchanges among the users.

The user can interact through the social websites in different ways such as login, bookmarking, tags, etc. Now, people from online communities, use them as means to extend their personal networks and for entertainment purposes [1].

The system of social network is recognized as a better component to improve the necessary factors in the fields of marketing, sales, and banking sectors. The main practice in these sites is to exchange thoughts on various services, product features had by the public (users) in the form of direct and indirect way [1][2]. Due to gravity and importance of security in web applications; we should use trusted operating systems for the same.

High secure operating systems called Trusted Operating Systems, these operating systems recommend a number of safety mechanisms that can assist defend information, make a system hard to break into, and detain attacks far better than conventional operating systems [3].

Hundreds of millions of people all over the world make use of social websites, Internet portals, blogs, Wikis, etc. trusted operating systems can be used for the same due to importance of security concern [4].

Actually, this security will come at a charge, since it can humiliate the performance of an operating system. This performance defeat is one of the reasons why TOS have not become popular in web applications. UML can be applied in many areas like embedded systems, web applications, commercial applications etc. Some UML tools generate program language code from UML. UML can be used for modeling the whole system independent of platform language. Trusted operating system can be designed by using UML based concepts.

While Trusted Operating Systems present an unbelievable quantity of security, observations about computing workloads recommend that only a few parts of the operating system security are really essential[5]. Web servers are the most excellent example.

For various web servers, the majority of the information on the server is freely readable and accessible on the Internet. so, if a Trusted Operating System is used on a web server, any security used to protected the privacy of the server's information is not needed. Any security used to protect the confidentiality of web server data can be considered a waste of computational resources [6][7]. The security needed in web servers is the security to guard the integrity of data, not the privacy of data. Based on this surveillance, this research article proposes the Security Performance Flexibility Model (SPF) of TOS to Web applications [5].

## 2. RELATED WORK

The citizens anticipate high standards of services, immediate access to information, well-organized transactions and support, whenever and wherever they need it but in a secure and safe fashion.

In this regard there is a requirement of high secured Trusted Operating Systems, which offer an unbelievable amount of security. Normally, due to high security reason these operating systems are being used in defense. But still these secure operating systems have limited scope in commercial sector and are not popular in corporate due to lower performance; actually this security will come at a cost.

According to SPF observation we feel that not all computing workloads need all the security in secure Operating Systems. SPF permits computer system administrators to selectively disable parts of the security protection in Trusted Operating Systems [9]. Whenever we disable some unnecessary parts of the Trusted Operating System security, performance of the computer system can potentially be increased [6].

System administrators can use the SPF framework to balance the security and performance needs in their particular computing environment [8]. First we have to identify which part can be disabled to achieve the maximum performance in specific web application. We should identify various operations in social networking to disable the undesirable operations to improve the performance in all respect. For identify the desired and undesired one we can float some option For example:- For which purposes do you use social media related to your work or profession (including your personal professional development)?

### Mark each that applies

- Specialized networking purpose
- Exchange and swaping of information with peers & friends
- Marketing or sales purpose
- Collecting market intelligence
- Individual professional development and growth
- News and information purpose
- Promotion of my personal "brand"
- Promotion of my office, company / organization

Based on Questionnaire and survey related to web applications like Social Networking sites, e-commerce sites, job portals, we can find following trend Hit rate, Max operations performed by users, Frequency of read, write, upload, share, comment, messaging operations, What is the chance that an event will occur?, Which patterns are significant?

After identify the answer of above question we can identify the frequency of operations performed by web users and according to our desired requirement we can disable some security checks in trusted operating systems to improve performance of web applications [1][21]. As we know all security checks are not important in all web applications. Selection of these criteria will vary from one application to another application.

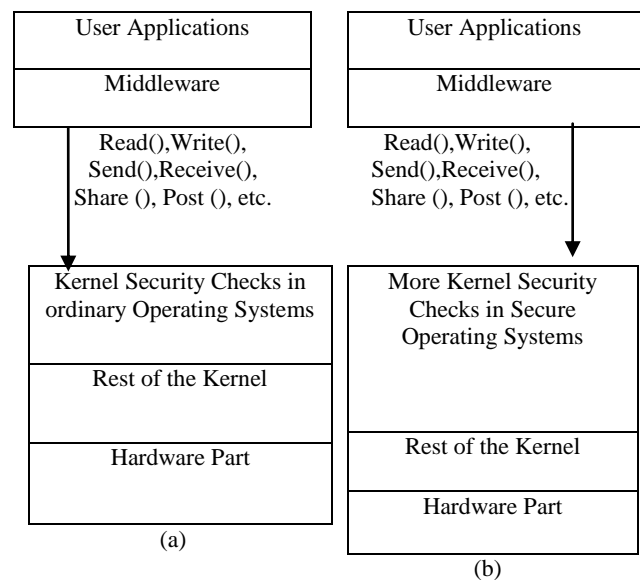
## 3. ARCHITECTURE

The fundamental structural design of Trusted Operating Systems is demonstrated under the Figure 1. As a prompt to the reader, the design of a social networking system is not the same as the execution. A mixture of implementations for trusted security mechanisms can be followed for the web applications.

This trusted operating system we can implement for social networking sites. For the designing and development we should use the concept of object oriented design and forward engineering for code generation in C++.In this paper I have generated code in C++ for forward engineering.

In TOS or secure operating systems, there is a larger amount of security and safety placed into the operating system. Figure 1.b indicates this security and safety with a much thicker coating of kernel security checks [11][7].What is within the kernel security check layer of a Trusted Operating System depends on the design & implementation of social networking web application.

We can see in figure 1.b, the kernel security layer in trusted operating system may comprise DAC, MAC, Least Privilege, auditing, or any more number of extra security features [5][10].



(a) Normal Operating Systems (b) Trusted or Secure Operating Systems

**Fig 1: Security Framework of TOS**

In figure 1.b the main point is that the kernel security checks are a large amount in nature than conventional operating systems. This large layer of security and safety causes Trusted Operating Systems to bear performance degradation. All system calls to the kernel of TOS must go through this thick layer of security checks before they can do any useful and valuable work [6][7].

## 4. PROBLEM DESCRIPTION

As the architecture in Figure 1.b shows, the additional security checks in the kernel will cause Trusted Operating Systems to be slower than traditional operating systems [5]. If we implement the same trusted operating system for web applications like social networking sites, e-commerce sites, job portals, then we get lower performance but more security

features as we are using the trusted operating system for the same. The main problem is to balance security and performance of secure operating system for web applications. In this research paper we will suggest Object-oriented analysis based designing using forward engineering. This approach is a popular technical approach to analyzing, designing an application, TOS or business by applying the object-oriented paradigm and visual modeling throughout the development life cycles to foster better stakeholder communication and product quality[5][6].

## 5. PROBLEM SOLUTION

This research article proposes that performance in Trusted Operating Systems can be increased if security & safety can be disabled in few parts of the operating system. Performance can be amplified. This approach gives the opportunity to system administrators by which he or she can balance their security and performance needs in various web applications [11][6].

With a variety of diagrams from UML we can demonstrate clear views of system thus reducing time-to-market for business and corporate problem solutions. If we model any problem before its progress there are less chances that our development sketch will go wrong, thus decreasing development costs. Modeling decreases the risk and danger of mistakes [14].

For better performance for specific computer system workloads, this research paper suggests the security performance flexibility (SPF) model for Trusted Operating Systems. Normally we perform read, write, like/unlike, comment, share, post, add photos operations in social networking sites. On the basis of these operations, we can implement the SPF model for social networking sites. For the designing and development we should use the concept of object oriented design and forward engineering for code generation in C++. The combination of SPF and forward engineering can help to developers, designers, analyst as well as web applications user. The architectural thought behind the SPF structure is illustrated in Figure 2. For example, a system administrator can use SPF to turn off all read security and safety checks in a web server. By turning off the read security checks of a web server, it is probable the web server's throughput can be increased and improved [6][18].

In this research paper I have generated code in C++ for forward engineering using IBM Rational Rose Enterprise edition.

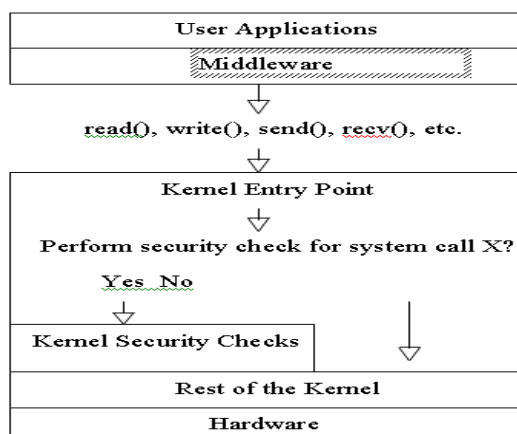


Fig 2: SPF Architecture for Web Applications

For object oriented modeling of web applications like social networking sites purpose, we can use UML for the above solution. This code generation based forward engineering of web applications can help the designers and software developers [3].

Taking the decision to adopt UML in the development process of social networking site does not mean that all UML diagrams must be used. It is up to the modeller to decide, which diagrams to use in the development process. The decision is usually based on the nature of the system; the domain, the functionalities, as well as the preferences of the modeller, since some UML diagrams might in some cases replace each other.

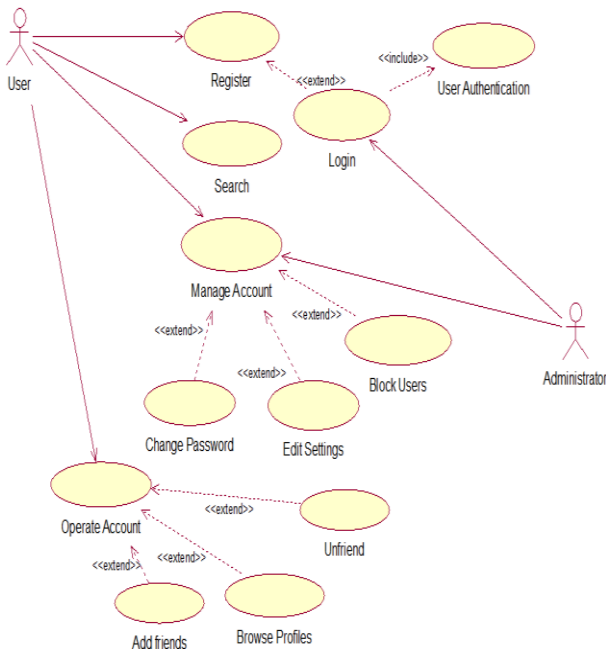
For web applications like social networking sites we will design use case diagram and class diagram [13].

We can implement the SPF and code generation in following case of web applications. Consider a Social Networking Website. The aim of the site is to let people network socially over the Internet. A user can register with the site. On doing so, he is connected with all other registered users of this site. To begin networking, he must search for names through a general search [2]. The system will display the public record of the user under the name. He can select a particular user and invite him to become a friend. On acceptance of the request, the latter's record will be visible in the Friends List of the user and vice versa [1][12].

The user can send and receive messages that can be of two types: public and private. A public message, when sent, will be visible to all the registered users browsing the public message list, whereas a private message will be visible only to the recipient. A registered user can upload his photographs and delete previously uploaded photographs. For the persons who do not have any knowledge of this site, an email can be sent, providing information of this site [15]. User can also delete an already added friend from his friend list. He is also allowed to send group messages to a group of friends [1].

First we have to design use case diagram because Use cases are a influential method for the elicitation and documentation of *blackbox functional* requirements. Use cases are printed in natural language, use cases are effortless to understand and provide an outstanding way for communicating with customers and users. Use cases diagram provide the groundwork on which to specify end-to-end timing requirements for real-time applications. Use cases can assist control the complexity of huge projects by decomposing the difficulty into most important functions (i.e., use cases) and by specifying applications from the users' perspective [9][19].

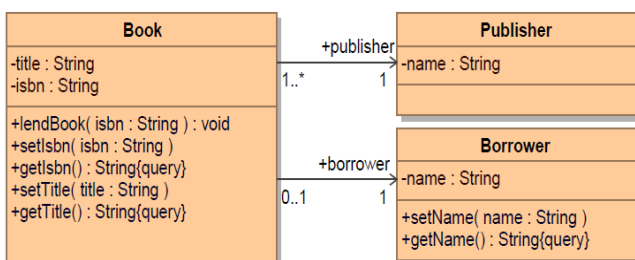
The generic Use case diagram of Social networking model is as follows –



**Fig 3: Generic Use case diagram of Social networking Application**

For the above web application we can design and develop the class diagram with the help of above use case diagram and can perform forward engineering for the same application with the help of IBM Rational Rose Software [3]. A class diagram is "a diagram that shows a set of classes, interfaces, and collaborations and their relationships; class diagrams address the static design view of a system; a diagram that shows a collection of declarative (static) behavior"

A class is seen as a container of objects that must have the same operations, attributes, and associations of that class, but with different attribute values. Class diagrams are used in any system development process that uses UML as a modeling notation [3]. They are helpful in the development process from the very beginning, where they can be used for identifying system requirements and its entities [20]. For example, they construct the initial system architecture in the analysis phase, while in the design phase; they are refined and extended to represent the complete system specifications. We can see in the following class diagram that any class diagram is mainly composed of classes and associations between them. There are many types of associations in the class diagram, e.g. inheritance, aggregation, composition, and dependency, which can be used for several purposes [3][14].

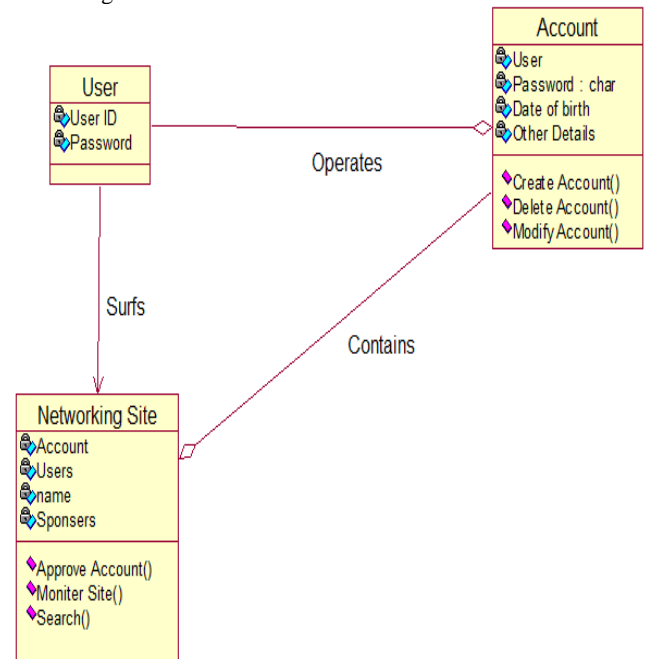


**Fig 4: Sample Class Diagram**

Figure 4 shows a sample class diagram, which is composed of three classes, i.e. Book, Publisher, and Borrower. The diagram shows the attributes and operations inside those classes. In addition, the diagram shows different elements, such as associations (i.e. Book-Publisher and Book-

Borrower), role names (i.e. publisher, borrower), and multiplicities (i.e. 1, 1...0, 1...\*).

On the basis of code and design we can implement the SPF model for trusted operating system i.e. well suited for various web applications. The generic class diagram of Social networking model is as follows –



**Fig 5: Generic Class Diagram for Social networking Site Forward Engineering for Social networking Site**

We know the Forward engineering in Rose is component-centered. Class wise code i.e. component-based programming is given below, this code can help the designers, developers during the development process of web applications. The class wise corresponding sample code segment in C++ is as follows-

```

Account Code
///##ModelId=51121D0B0335
class Account
{
public:
    ///##ModelId=51121E1803B9
    Create Account();
    ///##ModelId=51121E1D0112
    Delete Account();

    ///##ModelId=51121E2001B2
    Modify Details();
    Browse Account();

private:
    ///##ModelId=51121D420181
    char User Name;

    ///##ModelId=51121D4E0193
    int AccountID;

    ///##ModelId=51121D5D0166
    char Email ID;
    ///##ModelId=51121DCF0265
    char Password;
    
```

```
///  
ModelId=51121DDD0389  
int Date of Birth;
```

```
///  
ModelId=51121E080318  
Other Details;
```

```
};
```

### Networking Site Code

```
///  
ModelId=51121E2D01F4
```

```
class Networking Site
```

```
{  
public:  
///  
ModelId=51121E7D0161  
bool Approve Account();
```

```
///  
ModelId=51121E8102F7  
Site Maintenance();
```

```
///  
ModelId=51121E8601B2  
Monitor Site();
```

```
private:
```

```
///  
ModelId=51121E560188  
int No of accounts;
```

```
///  
ModelId=51121E61010D  
Accounts;
```

```
///  
ModelId=51121E6C02EB  
Users;
```

```
};
```

### User Code

```
///  
ModelId=51121D04036C
```

```
class User
```

```
{  
///  
ModelId=51121D2F0021  
int UserID;
```

```
///  
ModelId=51121D3503D4  
char Password;
```

```
};
```

Here we have converted the object oriented model into the target language (e.g. C++, Visual Basic, COBOL, and Java). In above example we generated code in C++. In this approach main benefit is higher flexibility that the developer is not afraid to repeat the process several times and is therefore more adventurous.

## 6. CONCLUSION

So it's my conviction that simply doing reverse engineering does not provide one sufficient opportunities to study about the true nature of the business nor to completely appreciate and study from past implementation shortcomings. Thus I'd recommend reasonable modeling with forward engineering whenever possible.

In this research paper, we specifically focused on UML (Unified Model Language) and component-based programming for forward engineering for web applications in trusted operating systems. As we know modeling provides arrangement for problem solving. If we model a social networking based web application system, we may notice

multiple solutions. Modeling assist us supervise oversee the difficulty of systems before really starting development work.

The main proposal in this research article is that all security in a Trusted Operating System is not necessary. Some non-essential security checks can be skipped in a Trusted Operating System may raise system performance. These non-essential security checks we can identify in any web application. This SPF model and forward engineering based approach gives system administrators the capability to balance security and performance needs of a web application [5][18]. This approach can be followed for any web applications.

This code generation based forward Engineering of web applications can help the designers and software developers during the development of SPF based web applications. With this approach we can also generate the database tables from class for any web applications [3].

The ideas and concepts discussed in this research paper are preamble to issues and solutions for modeling SPF based Web application using UML. The ambition of this effort is to present a coherent and complete way integrates the modeling of Web-specific elements with the rest of the application such that the level of detail and abstraction is suitable for designers, implementers, and architects of Web applications.

## 7. REFERENCES

- [1] M. Saravanan, G. Prasad, Karishma Surana, D. Suganthi, "Labeling Communities using Structural properties", International Conference on Advances in Social Networks Analysis and Mining, 2010.
- [2] Roy A, Karforma S, A Survey on E-Governance Security, International Journal of Computer Engineering and Computer Applications (IJCECA), Fall Edition 2011, Vol 08 Issue No. 01, Pp: 50-62, ISSN 0974-4983.
- [3] Mark Aldrich. Secured Systems and Ada: A Trusted System Software Architecture. Proceedings of TRI-Ada, 2001.
- [4] Philip Raeth, Maurice Kügler, Stefan Smolnik, "The Impact of Organizational Social Web Site Usage on Work Performance: A Multilevel Structural Interaction Perspective", 45th Hawaii International Conference on System Sciences, 2012.
- [5] ABRAMS95 Michael Abrams, et. al. Information Security: An Integrated Collection of Essays. IEEE Computer Society Press, 1995.
- [6] <http://cairo.cs.uiuc.edu/publications/papers/thesis-achu.pdf>
- [7] <http://www.ijric.org/volumes/Vol8/Vol8No8.pdf>
- [8] Booch, G., Jacobson, I., Rumbaugh, J. The Unified Modeling Language Users Guide. Addison Wesley, Reading, MA, 1998.
- [9] Argus Systems Group, Inc. <http://www.argussystems.com/>.
- [10] John Barkley. Managing Role/Permission Relationships Using Object Access Types. Third ACM Workshop on Role-Based Access Control, 1998.

- [11] Paul C. Clark. Policy-Enhanced Linux. Proceedings of the 23rd National Information Systems Security Conference, 2000.
- [12] Ivar Jacobson, Magnus Christerson, Patrik Jonsson, and Gunnar Övergaard, Object-Oriented Software Engineering: A Use Case Driven Approach, Addison-Wesley, Wokingham, England, 1992. David Ferraiolo and Richard Kuhn. Role-Based Access Control. Proceedings of the 15th National Computer Security Conference, 2004.
- [13] Christian Friberg and Achim Held. Support for Discretionary Role-Based Access Control in ACL-oriented Operating Systems. Proceedings of the Second ACM Role-Based Access Control Workshop, 2005.
- [14] James B.D. Joshi, Walid G. Aref, Arif Ghafour, and Eugene H. Spafford. Security Models for Web-Based Applications, 2001.
- [15] [http://en.wikipedia.org/wiki/Applications\\_of\\_UML](http://en.wikipedia.org/wiki/Applications_of_UML)
- [16] Tamar Richner and Stéphane Ducasse. Recovering high-level views of object-oriented applications from static and dynamic information. In Hongji Yang and Lee White, editors, Proceedings ICSM'99 (International Conference on Software Maintenance). IEEE, September 1999.
- [17] K. Koskimies, T. M'annist'o, T. Syst'a, and J. Tuomi, "Automated Support for Modeling OO Software", IEEE Software, 15, 1, January/February 1998, pp. 87–94.
- [18] <http://www.ksc.com/articles/usecases.htm>
- [19] Donald G. Firesmith, "Modeling the Dynamic Behavior of Systems, Mechanisms, and Classes with Scenarios," Report on Object-Oriented Analysis and Design (ROAD), SIGS Publications, Vol. 1, No. 2, New York, New York, July/August 1994, pages 32-36.
- [20] Power, J. F. & Malloy, B. A. (2000), an approach for modeling the name lookup problem in the C++ programming language, in 'ACM Symposium on Applied Computing', Como, Italy.
- [21] Gómez J., Cachero C., Pastor O. Extending a Conceptual Modeling Approach to Web Application Design. In Proc. Conference on Advanced Information Systems Engineering (CAiSE), LNCS 1789, Springer-Verlag, 2000, pp. 79-93.

## 8. AUTHOR'S PROFILE

**Dr. Girish Sharma** is currently working as Associate Professor & Head of BPIBS, GNCT OF DELHI. He has More than 19 Years of academic/research and

administrative experiences in reputed organizations like Hindu College, Sonapat, ABES Engineering College, Ghaziabad, and College of Engineering Roorkee. He completed his PhD (2005) in Mathematics (Inter Disciplinary) at Department of Mathematics, Gurukul Kangri University, Haridwar. He completed his M.Tech in Computer Technology and Application (2003) from Delhi College of Engineering (DCE), Delhi University, Completed his MCA and M.Sc(Physics) in 1993 and 1996 respectively from Gurukul kangri University, Haridwar. His current research interests include different aspects of Distributed systems, Trusted operating Systems, Object oriented based Forward Engineering and Data Mining. He has published more than 55 research papers in various reputed national / international journals.

**Dr. Brijmohan Singh** is currently working as Associate Professor & Head of computer science and engineering department at COER, Roorkee. He has More than 12 Years of academic/research and administrative experiences in reputed organizations. He completed his PhD (2012) in Computer science and Engineering from Uttarakhand Technical University, Dehradun, UK. He completed his Master of Technology (Computer Science) with First Class (76.8%, Grade 'A') from Rajasthan Vidhya peeth Deemed University, Udaipur Rajasthan, in the session 2005 to 2007. He did his Bachelor of Engineering (Computer Science & Engineering) with First Class (74.89%) from College of Engineering Roorkee, in the session 1999 to 2003. His current research interests include different aspects of Digital Image Processing, Mobile Computing, Artificial Intelligence, Pattern Recognition; Multimedia Systems etc. He has published more than 30 research papers in various reputed national / international journals.

**Nitish Pathak** received his M.Tech in Computer Science and Engineering (2010) from SIET, Meerut, did his MCA in 2005 from UPTU, Lucknow and completed M.Phil. (Computer Science) in 2007 from Madurai Kamraj University, Madurai, Tamilnadu, India. Now he is Assistant Professor of Computer Science department in BVICAM, GGSIPU, Delhi (INDIA). He has More than Nine Years of extensive Corporate as well as Professional Teaching Experience at Graduate and Post Graduate level. He got Directorate Award (Three times on teacher's Day) for best teaching performance since 2007 to 2009, at ABES Engineering College, Ghaziabad, India.

His current research interests include different aspects of Trusted operating Systems; Object oriented based Forward Engineering and Data Mining. He has published more than 15 research papers in various reputed national / international journals.