

A Major Project Report On  
**INFORMATION SECURITY ENHANCEMENTS USING  
ENRICHED FRAMEWORKS IN CLOUD COMPUTING**

Submitted in partial fulfillment of the requirements  
for the award of the degree of

**MASTER OF TECHNOLOGY  
IN  
COMPUTER ENGINEERING**

By

**Shweta Sharma**

(Roll No. 2K12/CSE/20)

Under the guidance of

**Mr. Manoj Kumar**

Department of Computer Engineering  
Delhi Technological University, Delhi



**Department of Computer Engineering  
Delhi Technological University, Delhi  
2012-2014**



## **DELHI TECHNOLOGICAL UNIVERSITY CERTIFICATE**

This is to certify that the project report entitled **INFORMATION SECURITY ENHANCEMENTS USING ENRICHED FRAMEWORKS IN CLOUD COMPUTING** is a bona fide record of work carried out by Shweta Sharma (2K12/CSE/20) under my guidance and supervision, during the academic session 2012-2014 in partial fulfillment of the requirement for the degree of Master of Technology in Computer Engineering from Delhi Technological University, Delhi.

To the best of my knowledge, the matter embodied in the thesis has not been submitted to any other University/Institute for the award of any Degree or Diploma.

Mr. Manoj Kumar  
Associate Professor  
Department of Software Engineering  
Delhi Technological University  
Delhi



## DELHI TECHNOLOGICAL UNIVERSITY

### ACKNOWLEDGEMENT

With due regards, I hereby take this opportunity to acknowledge a lot of people who have supported me with their words and deeds in completion of my research work as part of this course of Master of Technology in Computer Engineering.

To start with I would like to thank the almighty for being with me in each and every step of my life. Next, I thank my parents and family for their encouragement and persistent support.

I would like to express my deepest sense of gratitude and indebtedness to my guide and motivator, **Mr. Manoj Kumar**, Associate Professor, Department of Computer Engineering, Delhi Technological University for his valuable guidance and support in all the phases from conceptualization to final completion of the project.

I wish to convey my sincere gratitude to **Prof. Rajeev Kapoor**, Head of Department, and all the faculties and PhD. Scholars of Computer Engineering Department, Delhi Technological University who have enlightened me during my project.

I humbly extend my grateful appreciation to my friends whose moral support made this project possible.

Last but not the least, I would like to thank all the people directly and indirectly involved in successfully completion of this project.

Shweta Sharma

Roll No. 2K12/CSE/20

## TABLE OF CONTENTS

Certificate	2
Acknowledgement	3
Table of Contents	4-5
List of Figures	6-7
List of Tables	8
Abstract	9
Questions & Answers	10-11
<b>Chapter 1: INTRODUCTION</b>	<b>12-17</b>
1.1 General Concepts	13
1.2 Motivation	14
1.3 Related Work	15
1.4 Proposed Work	16
1.5 Thesis Statement and Outline	17
<b>Chapter 2: CLOUD COMPUTING INFRASTRUCTURE</b>	<b>18-31</b>
2.1 Cloud Computing & Its Characteristics	18
2.1.1 Various cloud models categorization based on usage	18
2.1.2 Features of Cloud Computing	19
2.2 Security issues present in Cloud Computing	21
2.2.1 Prevalent security related concerns in Cloud	21
2.2.2 Security Attacks on Cloud	21
2.3 Different Frameworks for Cloud Security & Data Storage	22
2.3.1 Security Framework for data security	22
2.3.1.1 Deduced drawbacks of Framework	23
2.3.2A Light weight file monitoring approach for securing files	24
2.3.2.1 Sequence of events while client server interaction	26
2.3.3 I3FS: An In-Kernel Integrity checker & Intrusion detection	27
2.3.4 Flogger: An Intrusion detection system & file centric logger	31
<b>Chapter 3: PROPOSED SECURITY FRAMEWORKS</b>	<b>32-39</b>

3.1 For Secure Information transmission using cryptography	32
3.1.1 Algorithm Steps for Proposed Security Framework	33
3.1.2 Features of Proposed Security Framework	33
3.1.3 Cryptographic Techniques used	34
3.2 File Integrity Maintenance tool	35
3.2.1 Sequence of events of above mentioned tool	38
3.2.2 Algorithm Steps for New File Integrity Maintenance tool	39
<b>Chapter 4: IMPLEMENTATION WORK</b>	<b>42-54</b>
4.1 Implementation of light weight integrity tool	42
4.1.1 Outputs of tool	42
4.2 Implementation of newly Proposed File integrity tool	47
4.2.1 Outputs of Newly Proposed File Integrity Tool	48
4.3 Comparison between Different File Integrity models	54
<b>Chapter 5: CONCLUSION &amp; FUTURE WORK</b>	<b>55</b>
<b>APPENDIX</b>	<b>56-71</b>
<b>REFERENCES</b>	<b>72-74</b>

## LIST OF FIGURES

Figure 1:Cloud Architecture	18
Figure 2:Cloud Computing Types	20
Figure 3: Data Security Issues in Cloud Computing	23
Figure 4: Pie Chart for Security Concerns	24
Figure 5:Security Framework	25
Figure 6:Broadview of CFMT(Cloud File Monitoring Tool)	27
Figure 7:I3FS Architecture	29
Figure 8:Flowchart for I3FS Permission Checks	30
Figure 9:Proposed Security Framework Diagram	32
Figure 10: General Design Architecture for AES algorithm	34
Figure 11: Cipher Feedback Mode (CBC) k-bit version	35
Figure 12: RSA Digital Signature Scheme design	35
Figure 13:File Integrity Maintenance Tool Architecture	36
Figure 14:Proposed Model for File Integrity Maintenance Tool	37
Figure 15:Screenshot of Server Process during Initialization Mode	42
Figure 16:Screenshot of Client Process during Initialization Mode	43
Figure 17: Screenshot of Client Process during Integrity Establishment	44
Figure 18: Screenshot of Client Process during Integrity Establishment	45
Figure 19:Screenshot of Client Process during Integrity Monitoring	46
Figure 20:Screenshot of Server Process in Initialization Mode	47
Figure 21:Screenshot of Client Process in Initialization Mode	48
Figure 22:Screenshot of Client Process during Integrity Establishment for File	49
Figure 23: Screenshot of Client Process during Integrity Monitoring for Files	49

Figure 24: Screenshot of Performance analysis during Integrity Establishment	50
Figure 25:Screenshot of Performance analysis during Integrity Maintenance	51
Figure 26:Screenshot of File Restoration functionality	52
Figure 27: Screenshot of original client data stored on server	53
Figure 28:Screenshot of original client data stored after Integrity Calculation	53
Figure 29:Screenshot of server source file	56
Figure 30:Screenshot of server source file	57
Figure 31:Screenshot of client source file	58
Figure 32:Screenshot of common source file	59
Figure 33:Screenshot of Make file source file	60
Figure 34:Screenshot of des source file	61
Figure 35:Screenshot of des header source file	62
Figure 36:Screenshot of server source file	63
Figure 37:Screenshot of server header source file	64
Figure 38:Screenshot of client source file	65
Figure 39:Screenshot of common source file	66
Figure 40:Screenshot of des source file	67
Figure 41:Screenshot of des header source file	68
Figure 42:Screenshot of Make file source file	69
Figure 43:Screenshot of rsa header source file	70
Figure 44:Screenshot of sha2 header source file	71

## LIST OF TABLES

Table 1: Tools Developed & Characteristics	16
Table 2: Various Cloud Models Categorization based on Usage	19
Table 3: Security Issues in Cloud Computing	21
Table 4: Types of available cloud security attacks & Characteristics	22
Table 5: I3FS Database Schemas	29
Table 6: Comparison among various file integrity models	54



## ABSTRACT

Over growing years, the changing trend of IT infrastructure has introduced the concept of new technologies to meet the customer/client requirements. The usage of information and communication techniques has been widely distributed. The internet evolution has revolutionized the system configurations, network usage and service availability criteria. This leads to the emergence of Cloud Computing era. Cloud Computing provides a means to enable the distributed system resources to collaborate together and provides certain services related to infrastructure and platform etc. It's a pay per usage policy. Clients could consume services as per its needs and enroll for payment accordingly. This technology brought certain factors/issues to be considered like security threats, data security issues etc.

Cloud Security has become most challenging, crucial and important work in today's Cloud Computing era. The information security has become the top most priority for the cloud service provider along with the client as well. There have been various kinds of security threats imposed on system infrastructure. Certain surveys have been processed and a huge amount of research work is on progress to solve such potential threats through outsider attacks.

In this work, Cloud Security based research work has been done significantly and various approaches to improve security levels have been designed, implemented and evaluated accordingly.

## QUESTIONS & ANSWERS

### **Q1. Is your project a research work or is to be considered for patent work?**

**Ans:**This project is completely based on research work. Various researches based on cloud computing security have been studied and validated during this semester. Major security weaknesses have been shortlisted and dealt with. Certain security frameworks have been proposed and implemented in this thesis work presentation. In order to enhance the information security level in cloud computing environment, following approaches have been proposed:-

- FILE INTEGRITY MAINTENANCE TOOL FOR SECURE INFORMATION STORAGE IN CLOUD.
- A MODEL FOR SECURE INFORMATION TRANSMISSION USING CRYPTOGRAPHIC ALGORITHMS IN CLOUD COMPUTING.

### **Q2. Define usefulness of project work for the society/Industry?**

**Ans:**This project work is based on cloud computing information security issues. Nowadays, information security has become the topmost priority for organizations/corporate world. With the advancement in internet technologies, various online applications/web resources are accessed through internet users and most transactions happen online through client- server mechanisms. This brings the urgency for information security over internet. In Cloud Computing, there may occur various scenarios where client's data has been stored on storage server and it needs to be protected against any outsider/thwart attacks. Loss of confidential information could lead to higher risks related to financial losses, legal issues among organizations and it deteriorates the trust level relationship between client and cloud service provider. This all could lead to business losses and in turn proves fatal to further developing technologies in the world. Hence, we have proposed security frameworks to enhance the information security level with in cloud computing, which should serve the society and industrial world as a whole.

**Q3. Describe division of work /contributing among individuals(if any involved).**

**Ans:**This project work has been performed by myself (Ms.ShwetaSharma, RollNo-2K12/CSE/20) under the guidance of my learned supervisor Mr. Manoj Kumar, Associate Professor, Department of Computer Engineering, Delhi Technological University. I would like to thank him for his invaluable guidance, patient reviews and encouragement.

# CHAPTER 1

## INTRODUCTION

In today's economic world, Cloud Computing has revolutionized the ways businesses are executed and cost advantages cannot be ignored anymore. The governments have come forward to provide latest technology in the market to ensure information technologies are accessed using larger scalable and efficient systems along with the cost benefits. The Organizations have started to adopt Cloud Computing technology to perform their regular processes such as Application Development/Testing, Messaging/Email Services, Data Storage, Collaboration Software, Application Hosting etc.

The Cloud provides better scalability and improved computing flexibility. Cloud Security becomes a high priority when a cloud service provider is to be evaluated. Other risk factors may arise includes how and when encryption schemes are applied to the data under the infrastructures. Besides gaining cost and efficiency benefits, it becomes necessary for the industrialists to deal with cloud security associated risks and thwart attacks/threats as well.

There exists various security concerns including Data protection, enforcement of security policies and information related losses. Data Protection is a relatively more risk associated factor of Cloud Computing. Organizations need to ensure their 'big' confidential data is secured within cloud. Their security policies should be in accordance with cloud service providers. Even data under public domain needs protection as well as it can be used in decision support systems and could affect the whole business deals and scenarios. There have been various functional models of Cloud. In case of private cloud, there exists confidential and sensitive information such as –credit cards, Intellectual property/trade secrets, Financial, Health, State/Government secrets, Proprietary/Sensitive and Personally Identifiable. Thus, Enhancing Information Security levels become the top most priority to support Cloud Computing for running businesses in IT trade.

*We have done research work to alleviate the security of stored information with in the cloud and also to protect it from any outside attack during information exchange between client and server.*

## 1.1 GENERAL CONCEPTS

According to the National Institute of Standards and Technology in the US Department of Commerce, cloud computing means:

*‘A model for enabling convenient, on-demand network access to a shared pool of configurable Computing resources (e.g. Networks, servers, storage applications etc.) that can be rapidly provisioned and released with minimal management effort or cloud provider (i.e., Internet Service Provider) interaction’ [1].*

In order to utilize the facility of cloud, certain technical parameters are required to be considered. The software and hardware configurations of the cloud system must match with the client computerSystem. Both the systems must be synchronized before start of setup. Cloud Computing provides reliable pool of computing resources which includes configurable system and network resources [1].

In cloud computing, organizations outsource the computing resources from other cloud vendors. The companies may decide to transfer their business applications/tools/databases on Cloud platform. They are required to follow certain configurations and technical parameters related to their software and hardware needs before deciding to move to the cloud.

According to [2], there have been various developed features of cloud computing, have been enforced and provided to the organizations if security and privacy risks are minimized. These features are as:-

- **Limitless Flexibility:**The various software applications/databases can easily be accessed through cloud computing platform. This provides better scalability.
- **Reliability &Security:** Both these features are imparted to the clients dealing with cloudPlatform. These requirements serve the purpose of quality improvements in terms of service delivery to the users.
- **Collaboration of application units:**Various software applications/UIs/Executable gets collaborated together on same platform to perform desirable computing functions. Thus benefits the user’s adaptability.
- **Portability:**Remote servers’stores and access client’s data based on constraints. The client may avail its stored data and applications as per his/her requirements.

- **Simpler Devices:** Devices such as PDAs, cellphones, video recorders etc may be used to interface with cloud platform and thus becomes easily accessible. The true benefits of cloud computing can be utilized if real time privacy and security issues can be addressed to protect information part of the cloud platforms.

## 1.2 MOTIVATION

In order to consolidate cloud computing to become robust and feasible multi-purpose solution, Security is a key requirement. According to academia researchers, organizations and various distinct groups, it has been agreed upon that cloud security is a crucial concern and an obstacle in order to make an effective and efficient system[3].

There are various security concerns as confidentiality, service availability and information security. Cloud computing security types [4] -

- **Network Security:** It emphasizes problems associated with configuration settings and network communications. Under cloud infrastructure, remote servers get connected to form internal networks which may lead to various network related issues, which becomes a challenge for the researchers to be worked upon. It includes transfer security, firewalls and security configurations.
- **Interfaces:** APIs (Application programming interfaces) are utilized to access IaaS/PaaS services. It may cause security issues when user interfaces are also accessed to reach virtualized servers and their resources. In fact, authentication mechanism may also lead to certain security issues within cloud.
- **Data Security:** The quality standards of data need to be maintained. CIA (Confidentiality, Integrity and Authentication) criteria should be fulfilled in cloud. The secure storage facility must be provided through integrity maintenance. Cryptographic Techniques are used to solve such issues and may be implemented to any security based domain where certain level of security is desired. Data loss must be avoided and redundancy must be detected and controlled.

- **Virtualization:** The virtual machines (VMs) share same hardware and software resources. There may occur crossVM conflicts, data leakage or any form of data exploitation while using virtualization technique in cloud setup.
- **Governance:** Governance related issues include service control, data control mechanisms wherein SLAs (Service Level Agreements) are followed as per customer services with cloud. Loss of government information may cause serious security threats to the country reputation.
- **Compliance:** Service level compliance is compulsory for customer dealings. Service Outages must be performed on time regularly to avoid any kind of service conflicts. Customer audits may also be done in timely fashion to check security and protection policies as well. And so; SLAs must be followed religiously to maintain running cloud businesses.
- **Legal Issues:** Law enforcement measures are applied to client data stored on a geographical location. It must be as per judiciary laws of a given country. Potential security attacks/threats are possible from insiders of cloud platform.

### 1.3 RELATED WORK

As per [3][5], the Cloud Security includes around 9% data security issues which further categorizes into data redundancy, data loss, data availability related issues [3]. Data security becomes the top priority concern for the researchers as it may affect the whole cloud computing business. The highly secured applications/software require the access of confidential information of client and may also require secure cloud storage services through cloud platform.

Cloud computing demands certain security models which could enhance the quality standards of secure information transmission between cloud server and client. There have been certain amounts of research work conducted in this direction. Certain security frameworks have been proposed and implemented [6] to contribute the Confidentiality, Integrity and Authentication (CIA) criteria.

Various IEEE based researches have been performed recently for file based integrity providence [8] [9]. Secure and light weight approaches have been suggested for securing files in cloud environment. Various tools have already been implemented to sustain integrity of user's confidential information [8]. Few are listed as shown in Table 1:-

<b>TOOLS DEVELOPED</b>	<b>CHARACTERISTICS</b>
VMFence[11] proposed by Hai Jin et al	Monitors network flow and integrity in real time.
Storage-basedIDS propose by Pennington et al. [12]	Allowsthe storage systems to watch for data modification.
I3FS[13]	Intercepts file system calls and injects its integrity checking operations in kernel mode.
Xen FITs[14]	Monitored system consists of breakpoints that intercept file system calls. E.g. open, close, write.
Flogger[15]	File centric logger for monitoring file access and transfers within cloud.
Tripwire[16]	A Host based IDS that alerts on macro changes to the files and folders.

They monitor network flow through cloud infrastructure as various clients could connect to cloud server through virtual machines(VMs).Various network based attacks may occur on server. There is an immense requirement for light weight tools which could provide integrity checks over stored data.

We analyzed that for cloudcomputing, there is a requirement for a light weight, efficient, low operational cost and secure optimized solution. In tools such as file integrity loggers, there exists an immense database availability which could be an overhead to the cloud server memory utilization functions.

#### **1.4 PROPOSED WORK**

The Cloud Security frameworks which we will present here includes:-

- FILE INTEGRITY MAINTENANCE TOOL FOR SECURE INFORMATION STORAGE IN CLOUD.
- A MODEL FOR HIGH LEVEL INFORMATION SECURITY IN CLOUD COMPUTING USING CRYPTOGRAPHIC ALGORITHMS.

The above mentioned techniques provide certain benefits for cloud computing as:-

- First scheme provides optimized information integrity check functionality with minimum memory requirements (w/o any database support on remote server)



and confirms trust based relationship between client and server (while includes client involvement during the entire process).

- Another scheme represents the extension of first scheme in order to enhance security level during information transmission between client and cloud server. For confidential information to be transferred, cloud server may implement such models to avoid data loss and other substantial outsider/thwart attacks as well.

## **1.5 THESIS STATEMENT & OUTLINE**

The remainder part of the thesis is organized in the following sections:

**Chapter 2:**It represents research background based on cloud computing & its characteristics, along with Cloud Security and corresponding possible attacks. This section elaborates frameworks for information transmission and other tools related to storage based security under cloud.

**Chapter 3:**This section describes the research methodologies (proposed cloud security frameworks) to improve the quality standards for stored/ transmitted information in cloud computing. It includes the description of all modules and techniques applied to develop integrity tool.

**Chapter4:**This section consists of research analysis and represents implementation details of integrity tools.

**Chapter 5:**In this section,conclusion and future work of the thesis is given.

**APPENDIX:** It includes snapshotsof source code for both File Integrity based projects.

## **CHAPTER 2**

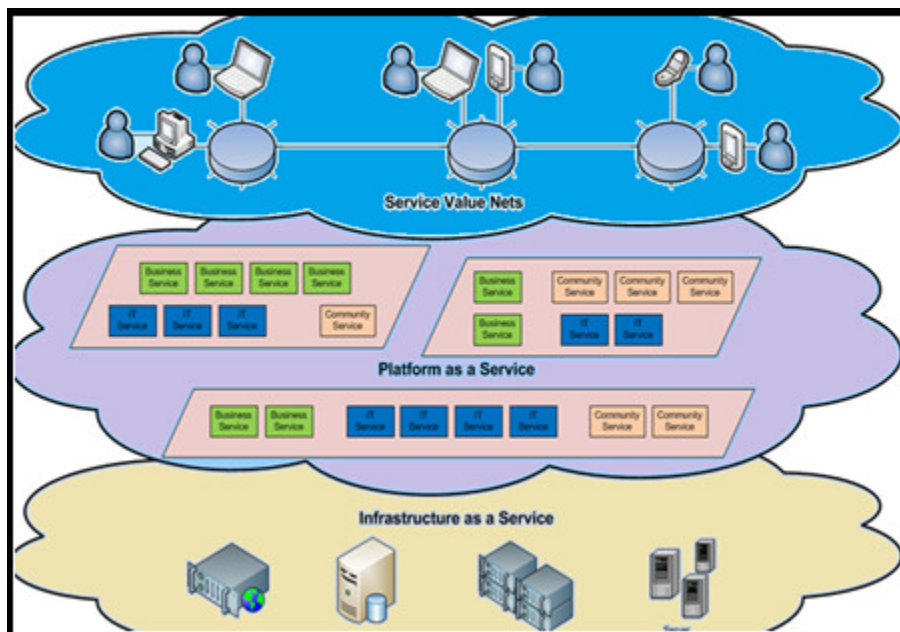
### **CLOUD COMPUTING INFRASTRUCTURE**

The Cloud Computing infrastructure includes storage based remote servers, network and configurationally resources,applications/software/executable & virtual machines (VMs)

availability to the clients. The organizations access cloud facility to ensure cost effective profitable business returns. The services (applications, software's) are deployed to the cloud servers before running their businesses. They are being tested as per network resource availability.

## 2.1 CLOUD COMPUTING & ITS CHARACTERISTICS

Cloud Computing signifies a collaboration of distributed systems with configurationally network Resources to perform distinctive tasks. As per our research study [17], following table describes the features provided by Cloud Computing to its users:



**Figure 1: Cloud Architecture**

### 2.1.1 VARIOUS CLOUD MODELS CATEGORIZATION BASED ON USAGE

S.NO.	TYPES OF CLOUD MODELS	SPECIFICATION & USAGE
1.	IaaS (Infrastructure as a Service)	Availability of virtual servers to clients for configuration and management.
2.	PaaS (Platform as a Service)	Supply of servers to customers for development related purposes.
3.	SaaS (Software as a Service)	Software/Application based services are provided
4.	SaaS (Security as a Service)	Facility of security solutions.
5.	IDaaS (Identity as a Service)	Management of identities in the cloud.
6.	CaaS (Communication as a Service)	The consumer can utilize Enterprise level VoIP, VPNs in economic scales.
7.	MaaS (Monitoring as a Service)	Application /server status are to be checked through monitoring tools during downtime.

### 2.1.2 FEATURES OF CLOUD COMPUTING

#### A. PUBLIC CLOUD

In Public cloud, the infrastructural resources are shared in public mode to all available clients. A public cloud is governed through the cloud service provider and is outside of user's organizations. There are enhanced security associated risks in such clouds as available information can be retrieved by any entity through outsider attacks. This cloud possesses multitenancy as stored data may belong to more than one organization.

#### B. PRIVATE CLOUD

In Private cloud (also known as internal cloud); the infrastructural resources are exclusive for particular company/organization. The private cloud is specific to provide services to limited number of users only. They may be maintained through particular organizations or third party vendors'. The private cloud is generally owned by large organizations for their operational functions and stores confidential information related to them.

### C. HYBRID CLOUD

The Hybrid cloud constitutes both public and private clouds and provides services to users in terms of interoperability. It represents that private cloud services can be extensible to public cloud services by cloud service provider. Both functionalities have been collaborated to prepare hybrid cloud.

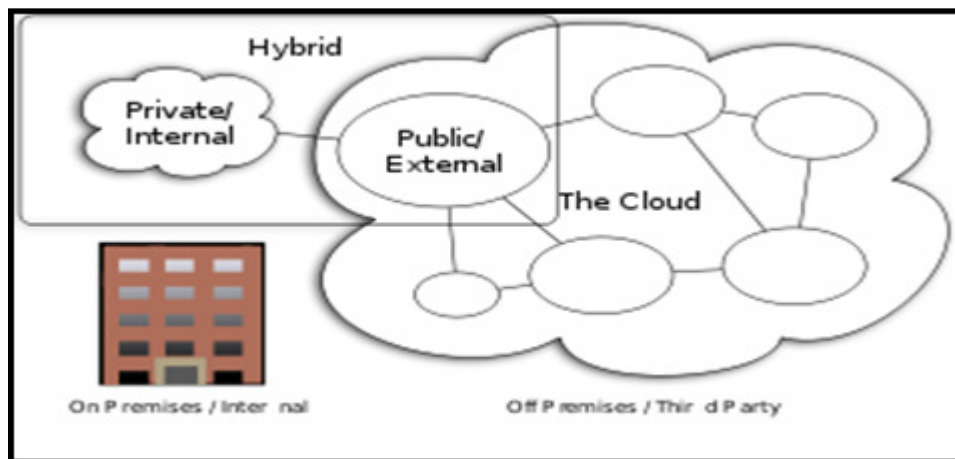


Figure 2: Cloud Computing Types

### D. COMMUNITY CLOUD

The Community cloud is beneficial for some community/government agencies. The government agencies can provide its services to another government agency of the same kind. And this way, cooperation and trust mechanism gets established among organizations.

## 2.2 SECURITY ISSUES PRESENT IN CLOUD COMPUTING

Following table [3] shortlists the prevalent security related concerns in Cloud Computing [18][19]:

S.NO	SECURITY ISSUES IN CLOUD	DEFINITION
1.	Data Privacy & Confidentiality	Safeguard of sensitive information between clients and cloud service providers.
2.	Backup	Maintenance of originality of data through data replication over cloud server.
3.	Authentication	Identification of client/server as trusted entities among themselves.
4.	Integrity	Preservance of intact original information.
5.	Interception of Data	Data modifications may occur on cloud server through security breaches.
6.	Intermediary	Intermediate parties legal rights must be protected to ensure proper transactions among third parties involved in cloud system.
7.	Data Storage Location	Customer ensures the data storage location within cloud and liabilities in case of data exposure must be decided beforehand.
8.	Governing Laws and Jurisdiction	Legal procedures to be followed while performing transactions among different companies with distinct countries involved.
9.	Vendor Contracts	Organizations providing cloud services may not warranty security constraints to users.
10.	Willingness to Cloud	Weak internet facility does not lead to data migration on cloud.
11.	Standardization	Clash of policies over cloud computing agreements among organizations.

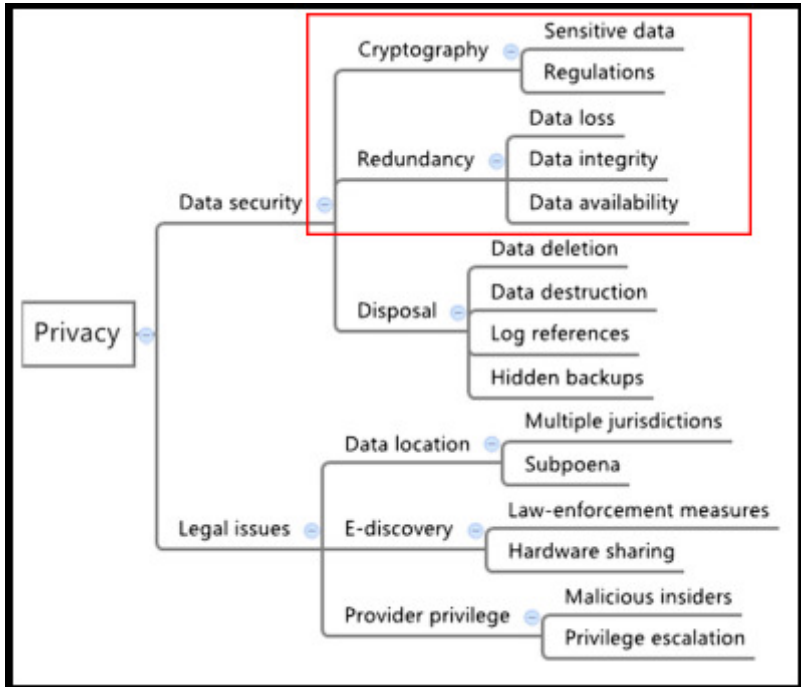
### 2.3 SIGNIFICANT SECURITY ATTACKS ON CLOUD

The below Table [4] represents and defines the collection of cloud security attacks

[20][21]-

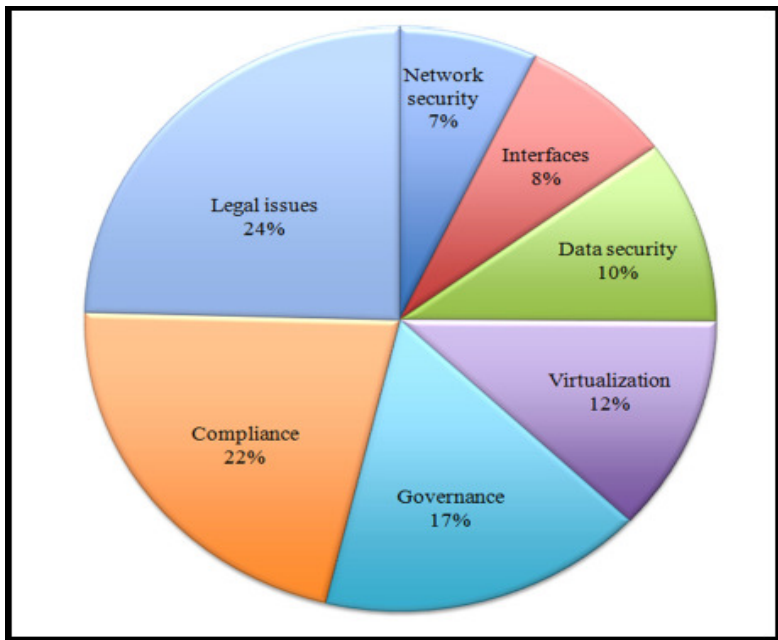
<b>S.NO</b>	<b>TYPES OF AVAILABLE CLOUD SECURITY ATTACKS</b>	<b>CHARACTERISTICS</b>
1.	Distributed Denial of Service Attack(DDoS)	Indefinite suspension of services of clients connected through internet. Leads to loss of personal information of users in specified conditions.
2.	Masquerading	Attacker impersonates another person.
3.	Replaying	Attacker obtains an original copy of message from sender and tries to retaliate later.
4.	Repudiation	Message sender/receiver may deny being the same for a given exchanged message.
5.	Insider Threats	Superuser privileges may get misused through cloud service provider's database administrators.
6.	Software & Security Management Risks	Dormant virtual machine aggravates the possibility of worms, viruses& malwares in cloud.
7.	Side Channel Attacks	Inherent cache monitoring techniques to check for information flow among clients and cloud service providers.
8.	Cloud Dependency Stack	Impact on security levels of business domains due to issues present in lower levels of cloud stack such as SaaS.
9.	Geographical Implications	The mobilization of virtual instances leads to loss of company's sensitive information through government agencies.
10.	Phishing	Act of acquiring confidential information from user by pretending as

	a trusted entity in cloud.
--	----------------------------



**Figure 3: Data Security Issues in Cloud Computing [3]**

The above marked portion has been studied and an approach has been proposed to resolve such issues.



**Figure 4: Pie Chart for Security Concerns [3]**

## 2.4 DIFFERENT FRAMEWORKS FOR CLOUD SECURITY AND DATA STORAGE

### 2.4.1 SECURITY FRAMEWORK FOR DATA SECURITY IN CLOUD COMPUTING USING CRYPTOGRAPHY [6]

- Information security is a crucial issue in cloud computing environments. Clouds have unlimited boundaries and the data may exist at any physical location. The cloud computing requires proper authentication schemes, data integrity and confidentiality.
- In this research paper, as it is proposed to implement an enhanced novel secure security algorithm in order to optimize the information security ensuring CIA – Confidentiality, Integrity and Authentication while storing and accessing the data from and to data centers and also in peer interactions.
- In this paper, a simple security framework using cryptographic algorithm the data protection is optimized by incorporating both public and private key cryptosystems for various cloud applications. We have examined the performance and have verified the test cases of our models in a simple cloud setup. We have achieved enhanced data security using AES, RSA and SHA algorithm with the minimal cost and effort.

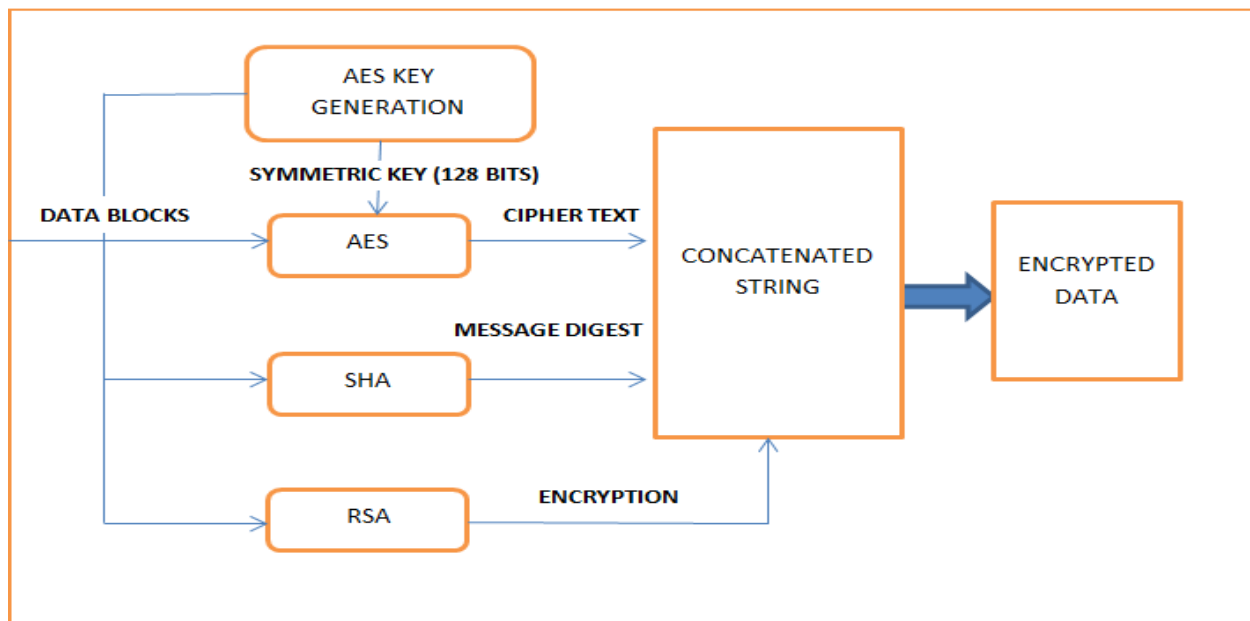


Figure 5: Security Framework [6]



#### **2.4.1.1 DEDUCED DRAWBACK OF THE ABOVE FRAMEWORK**

- No valid Client-Server Authentication scheme has been implemented.
- Server storage security criteria has not been included while client-server interaction scenario.
- Less secure framework and weak(as concatenations can be more vulnerable for brute force attacks).
- Various other randomness related security factors have not been included.
- Can't be preferred for confidential data related to banking applications and other brokerage activities.

#### **2.4.2 A LIGHT WEIGHT CENTRALIZED FILE MONITORING APPROACH FOR SECURING FILES IN CLOUD ENVIRONMENT [8]**

- This suggests development of a lightweight platform with low cost file monitoring approach and tool for securing important files from modifications in Cloud environment.
- The tool resolves the tampering problems with important files from VM users.
- It does not need any support for file signature which requires hash databases for storage of file integrity.
- It can be applied to any environment with minimum changes and support.
- This tool is a centralized utility in cloud and runs on a privileged entity (VM) in cloud environment.
- The tool uses cryptographic checksum for appending the integrity of files to them and for their verification.
- The task of integrity establishment and checking for a file is periodic in nature.
- It does not require any database support to protect the integrity of the stored files.

#### **2.4.2.1 SEQUENCE OF EVENTS DURING CLIENT SERVER INTERACTION**

- It can run on privileged VM of a Cloud and is hence a centralized utility that can be accessed by administrator of privileged domain.
- The integrity establishment is one time (until the file has no authorized modifications or unwanted subversions) while monitoring is periodic in nature.
- Monitoring the file integrity includes again calculating the encrypted cryptographic checksum of the original contents of the file and comparing it with the checksum extracted from the file between the predefined tags.
- Integrity Establishment module adds the integrity which is currently calculated encrypted hash value of the contents of the file in between well-defined tags in the file itself. This operation is performed over all the files in the folders that are configured for integrity establishment.
- Integrity Monitoring Module checks for Integrity of the files in the folder. For each file the encrypted hash is recalculated and checked for equality with values stored in them inside predefined tags. Server checks for the file integrity (by recalculating the hash value for the file and matches with originally stored hash value).
- If values match, it signifies that file is intact otherwise has been modified by any outside intruder.
- After establishment of integrity, the file is decrypted by server and encrypted using model of symmetric and asymmetric algorithms for secure and confidential transfer of data from server to client.
- The generation of session key takes place and is shared through server for every new created session b/w server and client.

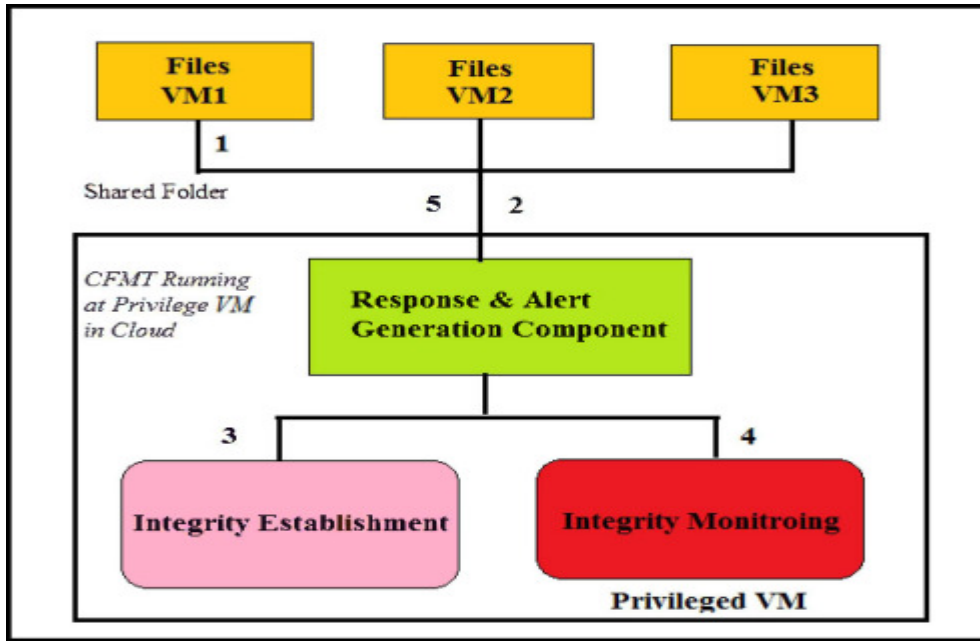


Figure 6: Broad View of CFMT (Cloud File Monitoring Tool) [8].

### 2.4.3 I3FS: AN INTRUSION DETECTION FILE SYSTEM WITH IN-KERNEL INTEGRITY CHECKER [13]

- It's an in-kernel approach to detect file modifications and any intruders. After performing integrity checks, it denies file access in case of any found modifications. I3FS has been implemented inside the kernel module of OS. It uses cryptographic checksums to calculate integrity and also stores security policies and checksums in different kernel Berkeley databases.
- Security policies are created by the administrator and managed and stored in relevant databases.
- Berkeley DB is a scalable, high-level performance, transaction-protection-based management of data which efficiently and persistently stores hash keys, value pairs using hash tables, B+ trees, or queues.
- FS stores four databases in the B+ tree format, so that we benefit from locality.
- Various get policy and cache policies have been framed and followed during computations to ensure secure access to files.
- The file is checked against I3FS permissions, and then it is computed against the cached policy and results, and after that, the checksum is computed for file information and is stored back to the respective databases.

- This model ensures secure file integrity checks but its biggest disadvantage is the requirement of , large databases on the server & which ,in turn, requires large memory availability and leads to an overhead for the cloud service provider to meet such memory expectations.

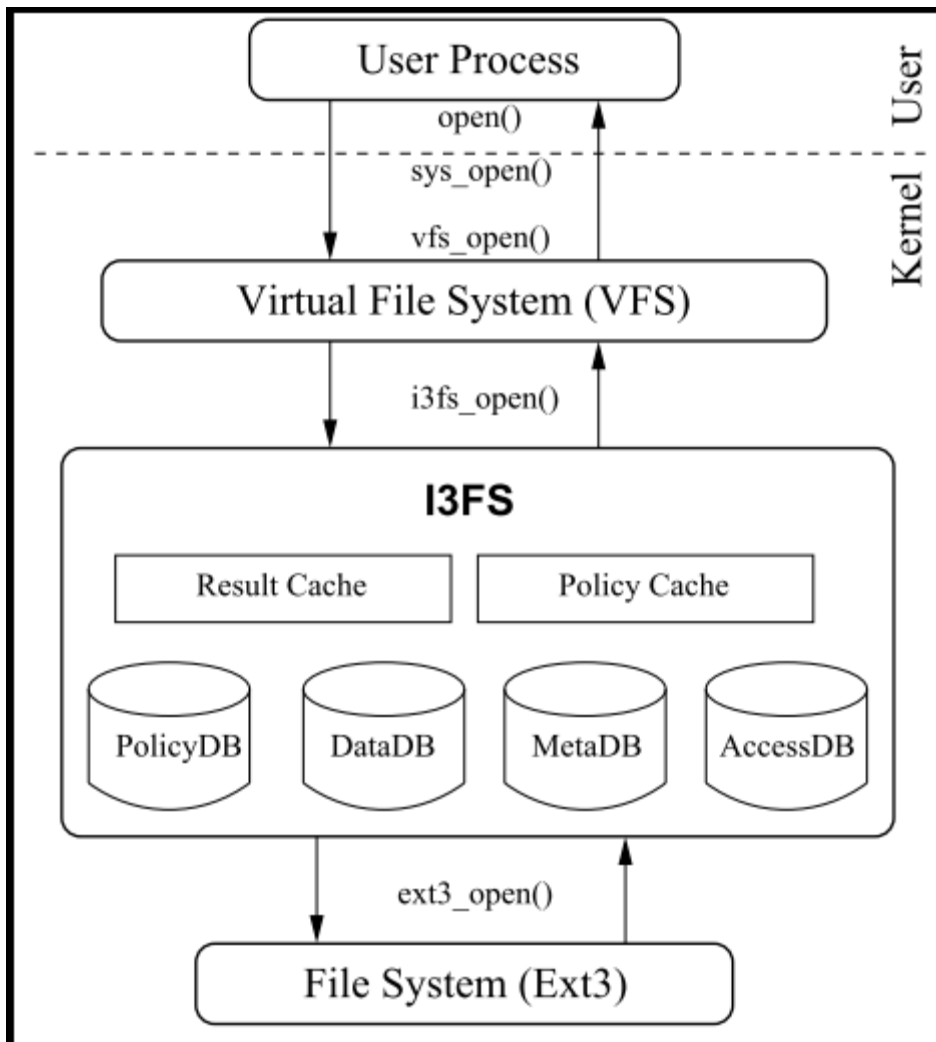


Figure 7: I3FS Architecture [13].

Table 5: I3FS Database Schemas [13].

S.NO	DATABASE	VALUE	KEY
1	policy dB	Policy bits,freq#	Inode#
2	data dB	Checksum value	Inode#,page#
3	metadb	Checksum value	Inode#
4	access dB	Counter value#	Inode#

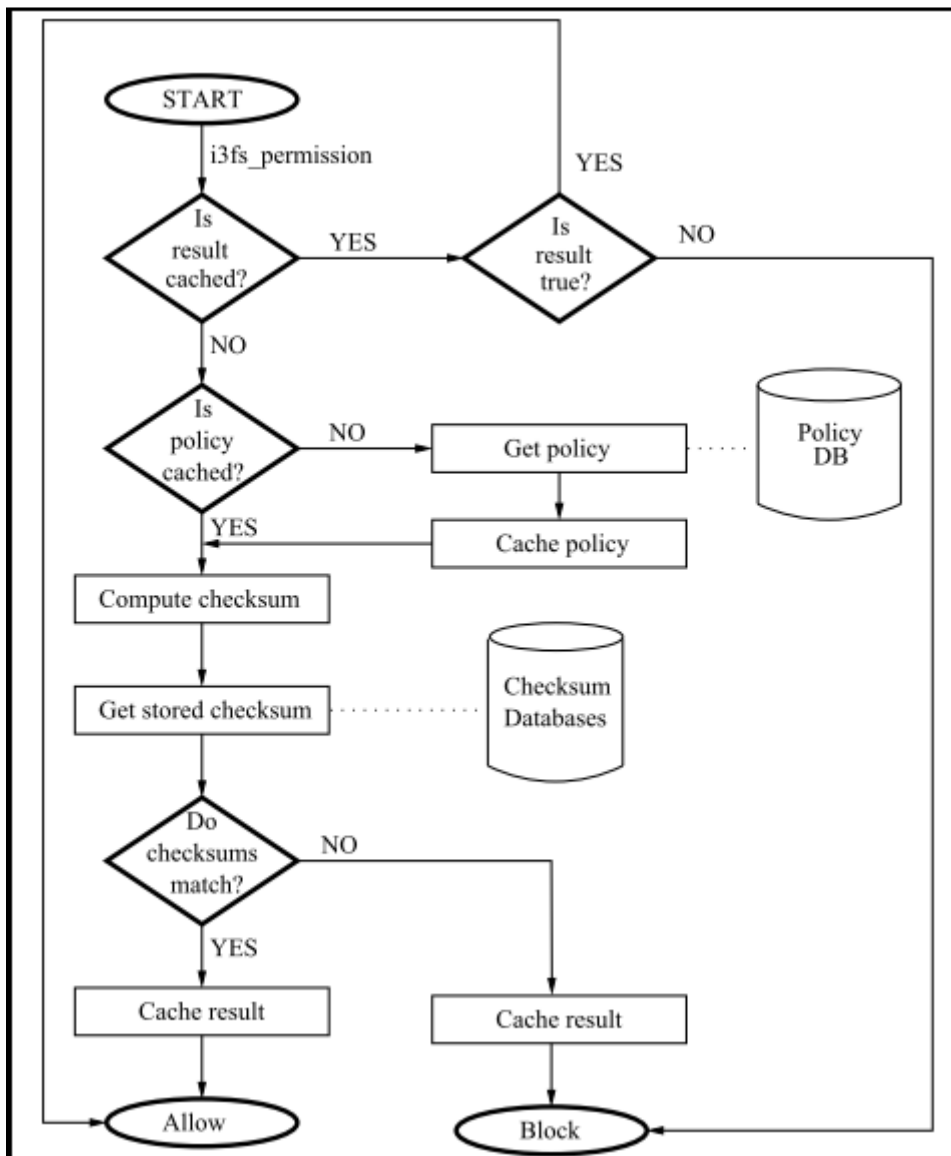


Figure 8:Flowchart for I3FS Permission Checks[13]

#### **2.4.4 FLOGGER: A FILE CENTRIC LOGGER TO MONITOR FILE ACCESS AND TRANSFERS IN CLOUD COMPUTING ENVIRONMENT [15]**

- Flogger is a centralized tool to compute file access logs using virtual and physical subnets.
- It has been implemented in Linux kernel and uses both VMs and PMs for logs based information.
- It intercepts every file access using windows/Linux floggers. It stores information based on various attributes such as- VM file access date/time, VM IP address, MAC address, ID & GID of various file owners of accessed files etc.
- The implementation includes various components such as windows Flogger, Linux Flogger, File Sender Client Program, File Sender Daemon, File Sender Server Program, Database Loader Daemons.
- Windows Flogger: A device driver which runs on PM and intercepts file operations and writes logs.
- Linux Flogger: A Linux based kernel module which intercepts network and file operations and writes events as PM logs.
- File Sender Client Program: It runs on VM and transfers VM logs from VM to PM through a communication channel.
- File Sender Daemon: It runs File Sender Client Program.
- File Sender Server Program: It runs on PMs and works on VM file logs, sent by File Sender Client program.
- Database Loader Daemons: It runs on PM. It transfers VM/PM based file logs to remote Database servers.

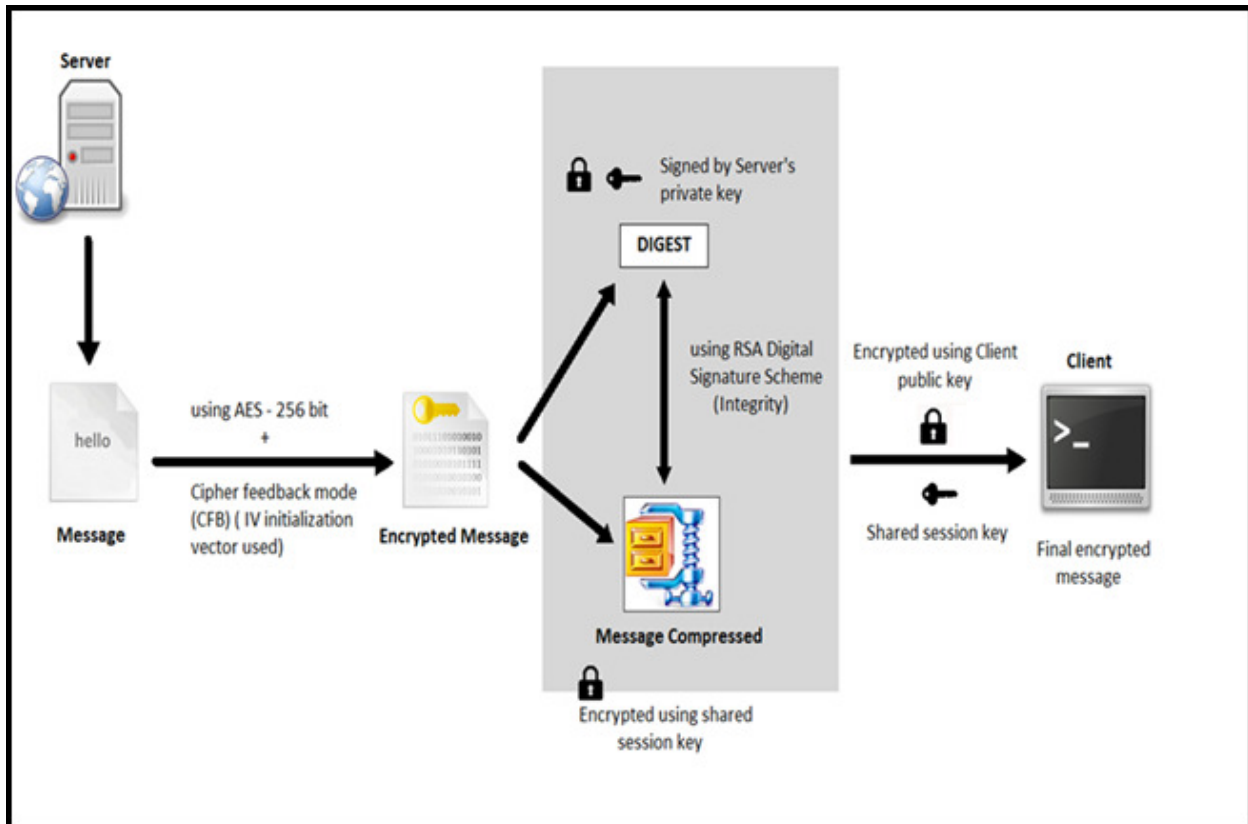
- File logger processes file centric based VM/PM logs & computes authorized/unauthorized file accesses by the owners. And separate remote database servers are required for close observation and storage of logs (includes archival based information as well).It proves to be a limiting factor for File Logger in real time scenarios where memory requirements becomes a huge constraint.

## **CHAPTER 3**

### **PROPOSED SECURITY FRAMEWORKS**

#### **3.1 PROPOSED FRAMEWORK FOR SECURE INFORMATION TRANSMISSION USING CRYPTOGRAPHIC ALGORITHMS IN CLOUD COMPUTING**

In cloud computing, whenever server wants to transfer the client data to the respective client ,there arises a requirement for secure transmission of data (avoiding any kind of leakage).Following are the proposed steps to be executed in the given scenario of cloud computing during server- client interaction for data:-



**Figure 9: Proposed Security Framework Diagram**

### **3.1.1 ALGORITHM STEPS FOR ABOVE PROPOSED SECURITY FRAMEWORK**

- A. Initialize server process SP and client process CP respectively.
- B. The server process listens to client port and establishes connection.
- C. SP generates a shared session key SS and server public key (e, n) and sends to client.
- D. CP generates its public and private key pair and transfers public key CK to SP.
- E. Server Encryption process {
- F. Apply AES-256 bit (Advanced Encryption Standard) Encryption on given message M
- G. Apply CFB (cipher feedback mode) on above.
- H. Apply RSA Digital Signature Scheme using server's private key d.
- I. Encrypt using CK.



- J. Encrypt the final output using SS to produce X (final encrypted data).
- K. X is sent to CP. }
- L. Client Decryption process {
- M. Decrypt using SS on X.
- N. Decrypt using client's private key.
- O. Apply RSA Digital Signature Scheme verifying process using server's public key (e, n).
- P. Apply Cipher feedback mode decryption process on intermediate result.
- Q. Decrypt using AES-256(Advanced Encryption Standard) bits decryption process.
- R. Output received is M.
- S. Exit }

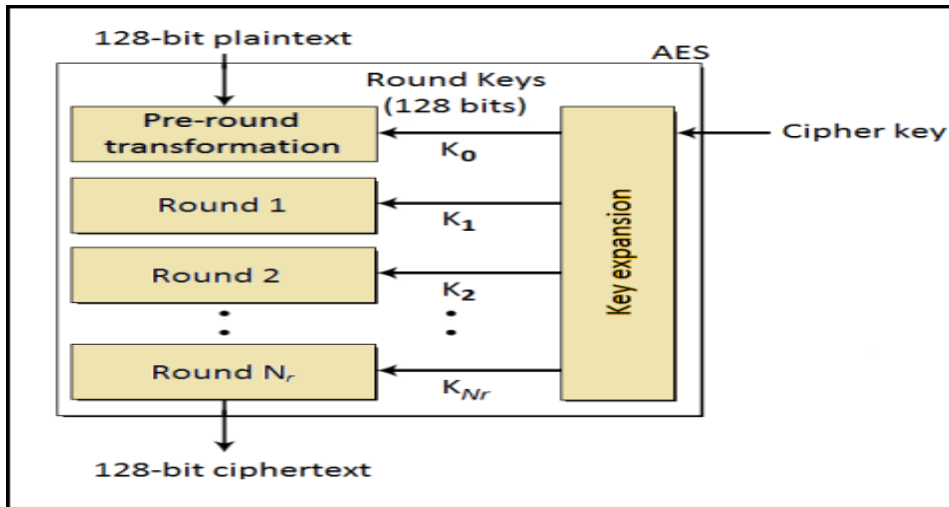
### **3.1.2 FEATURES OF PROPOSED SECURITY FRAMEWORK**

- It ensures secure transmission of information from server to client.
- It provides authentication and data integrity by using RSA digital Signature scheme.
- AES-256 bit encryption scheme (for confidentiality purpose).
- SHA-1/HMAC in order to generate digest (alleviates security and integrity).
- ZIP algorithm (compression algorithm) in order to reduce traffic flow between client and server (optional).
- Double encryption and cyclic codes were used to increase randomness in the cipher text.
- Shared Session Key provides an extra layer of security.
- Timestamp may be considered to be incorporated in the model.
- The model satisfies the standard requirements for data security (includes confidentiality, Data Integrity, Authentication).
- This framework increases the level of security which can be achieved for data transmission.

### **3.1.3 CRYPTOGRAPHIC TECHNIQUES USED IN ABOVE FRAMEWORK**

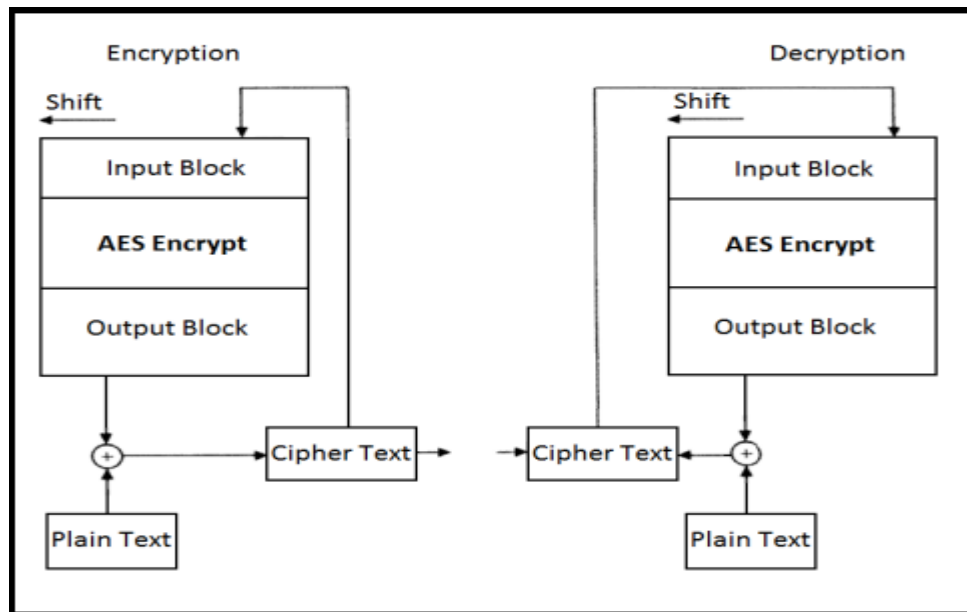
In above framework, following techniques have been utilized for secure transmission of message-

- **AES-256 bit (Advanced Encryption Standard) Algorithm:** It's a symmetric key cryptographic algorithm and uses 256 bit key size and 14 rounds to perform encryption. It can be applied to 128 bit input block.



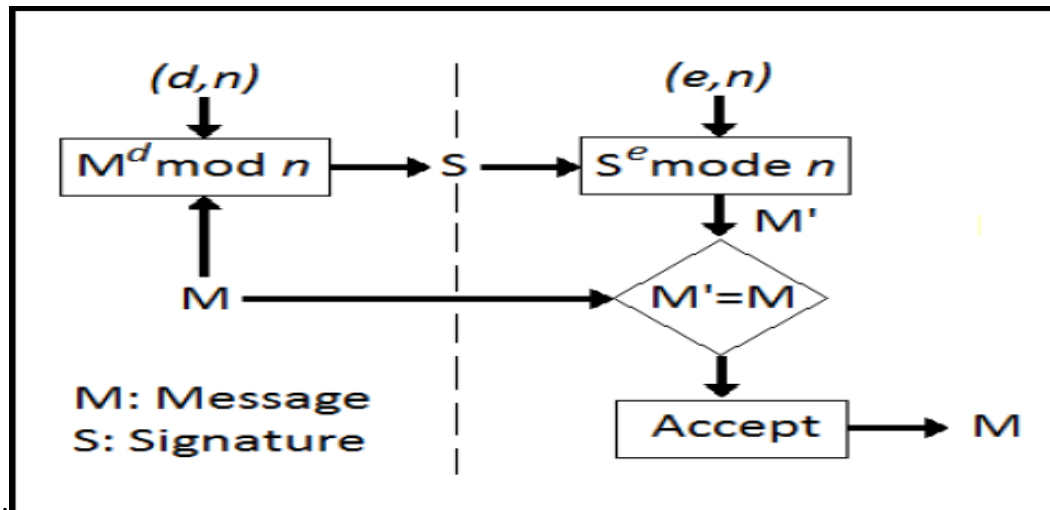
**Figure 10: General design for AES Encryption Algorithm**

- **Cipher Feedback Mode (CBC) Algorithm:** This algorithmic process produces randomness within cipher text produced using AES-256 bit. It decreases the chances of detection of information through any outsider attack.



**Figure 11: Cipher Feedback Mode K-bit version**

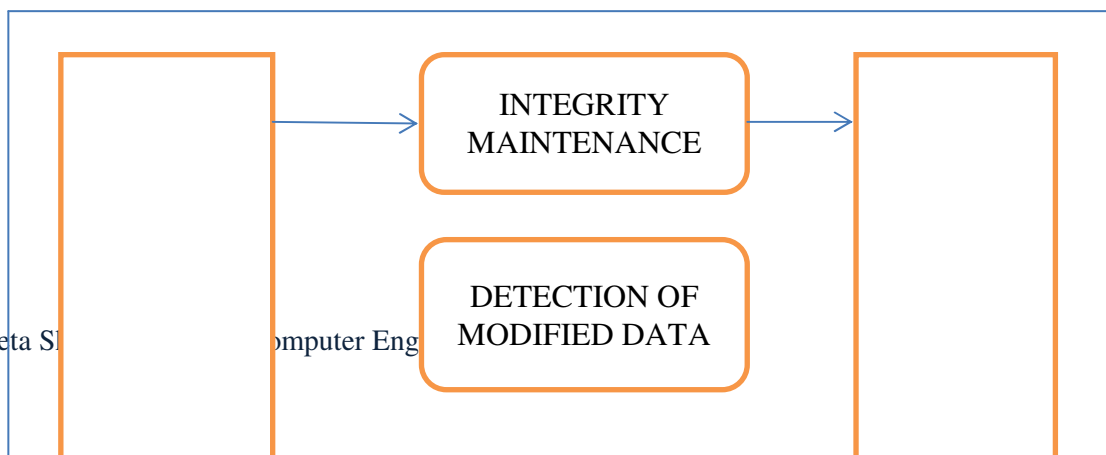
- **RSA Digital Signature Scheme:** This scheme provides integrity of data and authentication of client as well. It uses RSA (asymmetric key cryptography) scheme. The server applies signing process on  $M$  using its private key  $d$  and produces signature  $S$ . During verifying process (at client side), it applies server's public key  $(e, n)$  to produce  $M'$ . If  $M'$  equals  $M$  then client is verified and message integrity is considered to be intact.

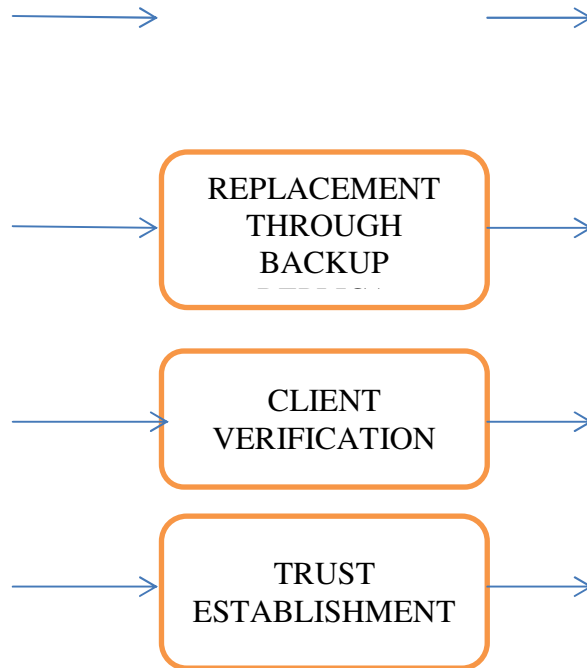


**Figure 12: RSA Digital Signature Scheme**

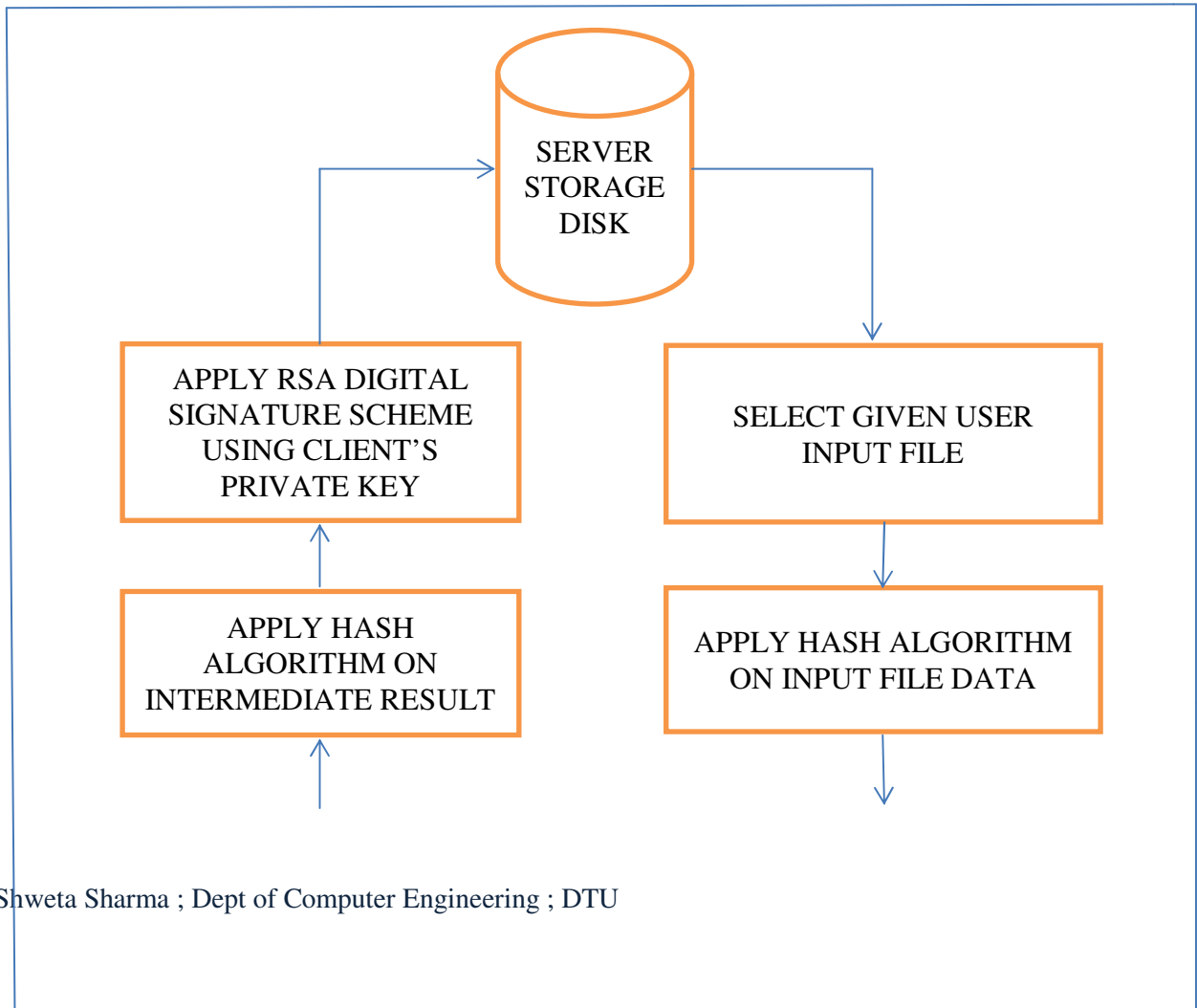
### 3.2 NEW FILE INTEGRITY MAINTENANCE TOOL FOR SECURE INFORMATION STORAGE IN CLOUD

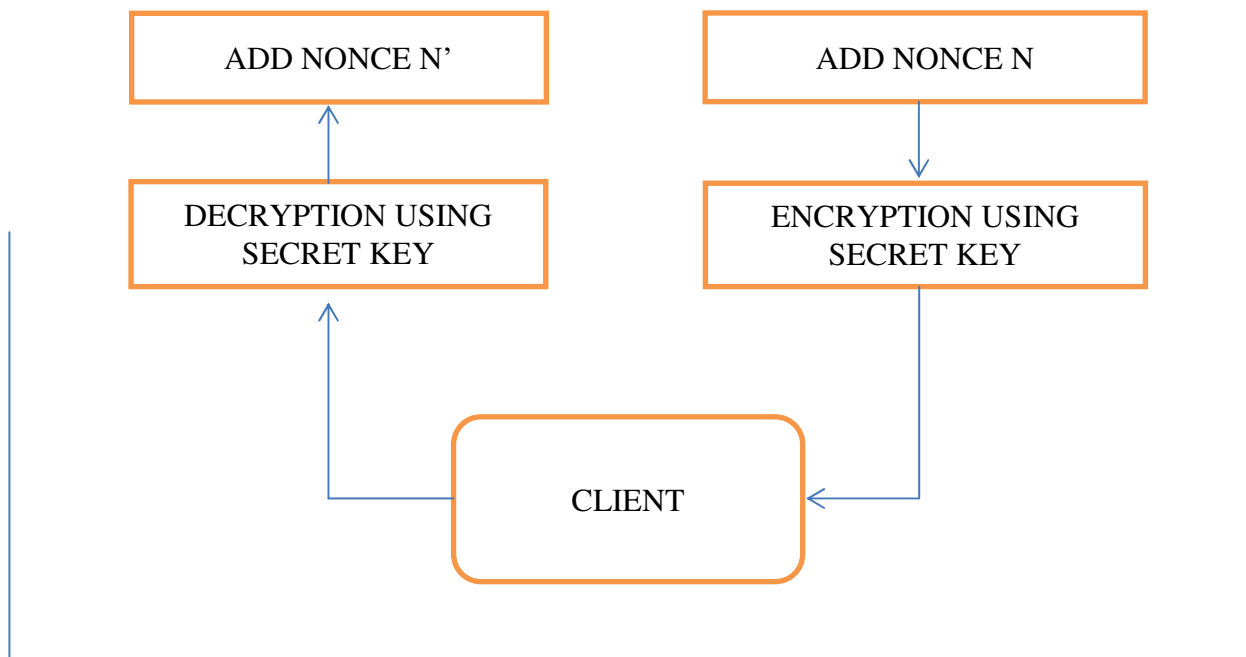
In a server- client environment, there occurs substantial data exchange between both entities. Client may require information storage in certain shared files/folders at remote server location for easy access and information retrieval. In those scenarios, the file data integrity becomes the topmost priority for the server. Certain schemes have been suggested with its regards [13][14]. Here, we would like to propose a new design to protect information integrity among client –server process exchanges.





**Figure 13: File Integrity Maintenance Tool Architecture.**





**Figure 14: Proposed Model for File Integrity Maintenance Tool for Secure Information Storage under Cloud.**

The above diagram represents the principle idea for proposed design scheme. Through the proposed design, we can achieve following functionalities:-

- **Integrity Maintenance-** It leads to establishment of information integrity within a given file.
- **Detection of Modified Data-** If attacked through outside intruders, it will be identified through proposed design scheme.
- **Replacement through Backup replica-** Modified data will be replaced through original copy of information (stored at server end in case of emergency).
- **Client Verification Process-** Valid client verification would happen using proposed design.
- **Alleviated Trust level Mechanism-** This scheme involves both client and server in order to finalize the contents to be stored within the file as final output of proposed cryptography design. This feature greatly enhances the quality of design. Here, no entity can be the intruder in worst case scenario and thus, certain level of quality trust establishes between both entities in a cloud computing environment.

### **3.2.1 SEQUENCE OF EVENTS FOR FILE INTEGRITY MAINTENANCE TOOL**

- During client–server interaction, to establish and monitor integrity of the stored client data. We have designed this model.
- Initially, the client stores its data in server specified directory.
- For integrity establishment, server accesses the file and applies a hash algorithm (sha-2) to generate message digest (MD).
- It appends (already shared random numbers called as Nonce N and N') N along with MD and encrypts using client- server shared secret key.
- The encrypted data gets transferred to client.
- Client receives the file and decrypts it using secret key and appends another nonce N' to the data and applies hash algorithm (sha2) to generate H(X).
- The Client implements RSA digital Signature Scheme to encode and transfers the final string to server for storage.
- At the same time, server may also verify the H(X) using client's public key.
- This way, client's involvement in integrity establishment gets achieved and server may also verify the client's authenticity using Digital Signature verification process.
- It enhances the trust level among server and client as they act as peers in determining integrity of stored data.
- And thus, more secure designed model for storage and maintenance of server data at server location.

### **3.2.2 ALGORITHM STEPS FOR NEW FILE INTEGRITY MAINTENANCE TOOL**

- A. Initialize server process SP and client process CP respectively.
- B. The server process listens to client port and establishes connection.
- C. Initialize the input file F's location on server hard disk.
- D. Integrity Establishment function
- E. {
- F. The Server process SP applies SHA-2 algorithm on F.
- G. Add Nonce N and apply AES (Advanced Encryption Standard) algorithm to produce intermediate result.

- H. The intermediate result stream is sent to client process.
- I. The client process CP applies AES decryption algorithm.
- J. Adds Nonce  $N'$ .
- K. Apply SHA-2 on intermediate result to produce  $H(X)$ .
- L. Apply RSA digital Signature scheme (signing process).
- M. CP transfers output stream to server process.
- N. The Server process SP stores the final output  $X$  in <secure> tags within  $F$ .
- O. }
- P. Integrity monitoring function (Invoked by SP/CP)
- Q. {
- R. Call Integrity Establishment function to produce output  $X'$ .
- S. Compare  $X$  and  $X'$ .
- T. If ( $X$  equals  $X'$ ) then print- File intact. Monitoring Completed.
- U. Else Replace  $F'$  with originally stored  $F$ .
- V. Call Integrity Establishment function
- W. }
- X. Client Verification Process
- Y. {
- Z. Initialize SP and CP.
- AA. SP applies RSA Digital Signature scheme (verifying process) on  $X$ .
- BB. Output is stored as  $Y$ .
- CC. If ( $Y$  equals  $H(X)$ ), then Client verified.
- DD. Exit }

## CHAPTER 4

### IMPLEMENTATION WORK

This model has been implemented in windows7 platform using Linux based Oracle VM virtualbox. This tool provides the facility of client- server environment on existing machine only. There have been various processes including client and server processes. The server process executes and establishes connection with client process using specified socket address structures. There are two types of processes which gets generated i.e. client and server processes.

#### 4.1 IMPLEMENTATION OF FORMER LIGHT WEIGHT FILE INTEGRITY MODEL [8][9]

This file integrity based model had been implemented using C programming in Linux platform. It provides certain functionalities (@International Conference for Internet Technology and Secured Transactions (ICITST-2012))-

- The client stores a particular file at server location and asks server process for integrity establishment. The server computes cryptographic checksum of the information stored inside given file.



- It applies DES encryption algorithm and then apply SHA-2 algorithm to generate final checksum and stores it inside file using secure tags.
- The server is required to monitor the stored checksum on a regular basis by recompilation of the checksum for the file and compares it with originally stored checksum.
- In this way, the file information gets protected from any outsider/thwart attacks on the server location.

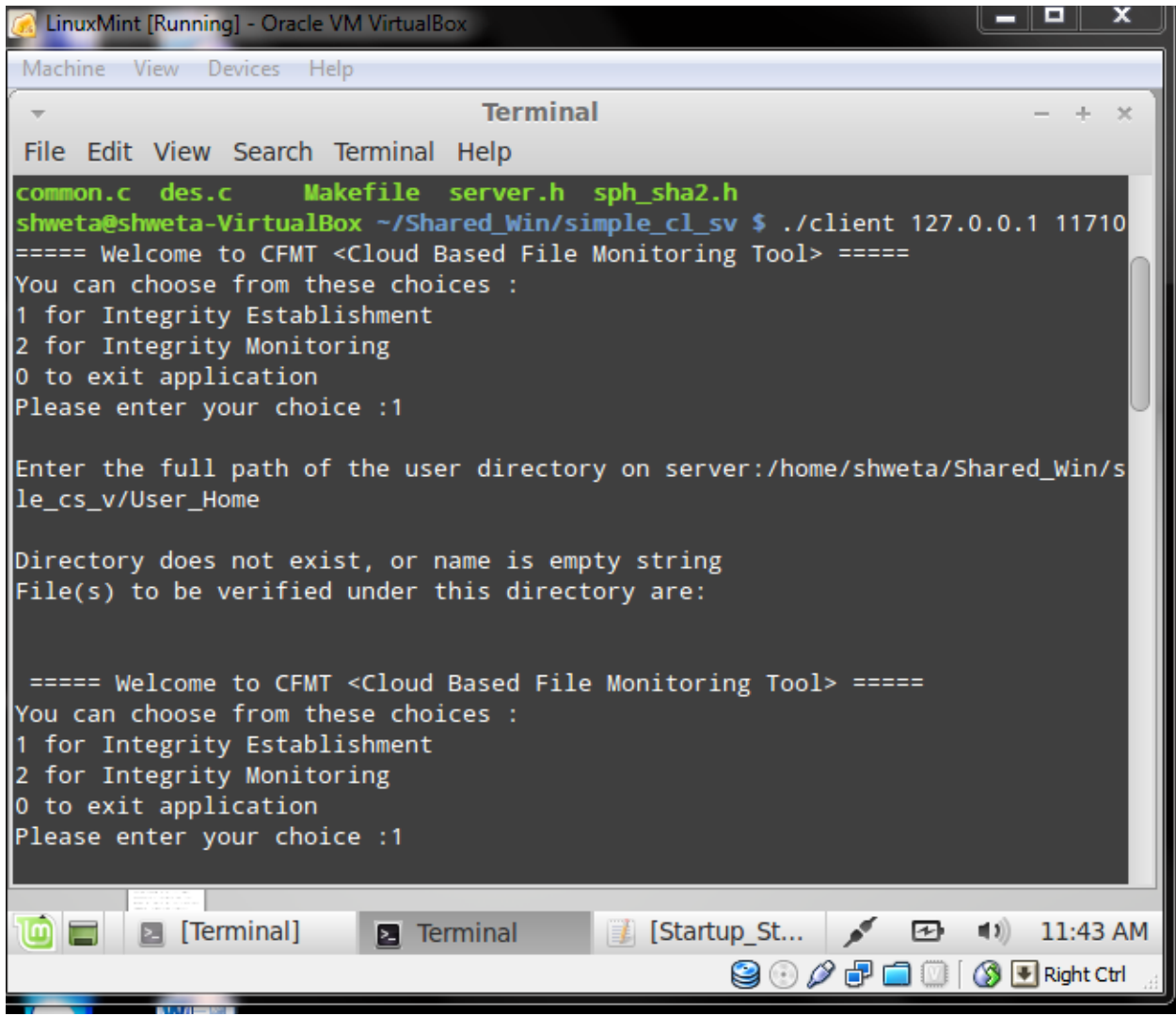
But, there are certain drawbacks associated with this approach. We have analyzed and shortlisted them as below:-

- There is no client involvement in the integrity establishment process.
- Client-server trust based relationship doesn't exist in this scenario.
- Attacker could be in form of server as only server is the active entity for processing computations.
- This model seems weak in terms of transparency and trust relationship between client and server. Thus, there is a scope of improvement here to enhance its quality.

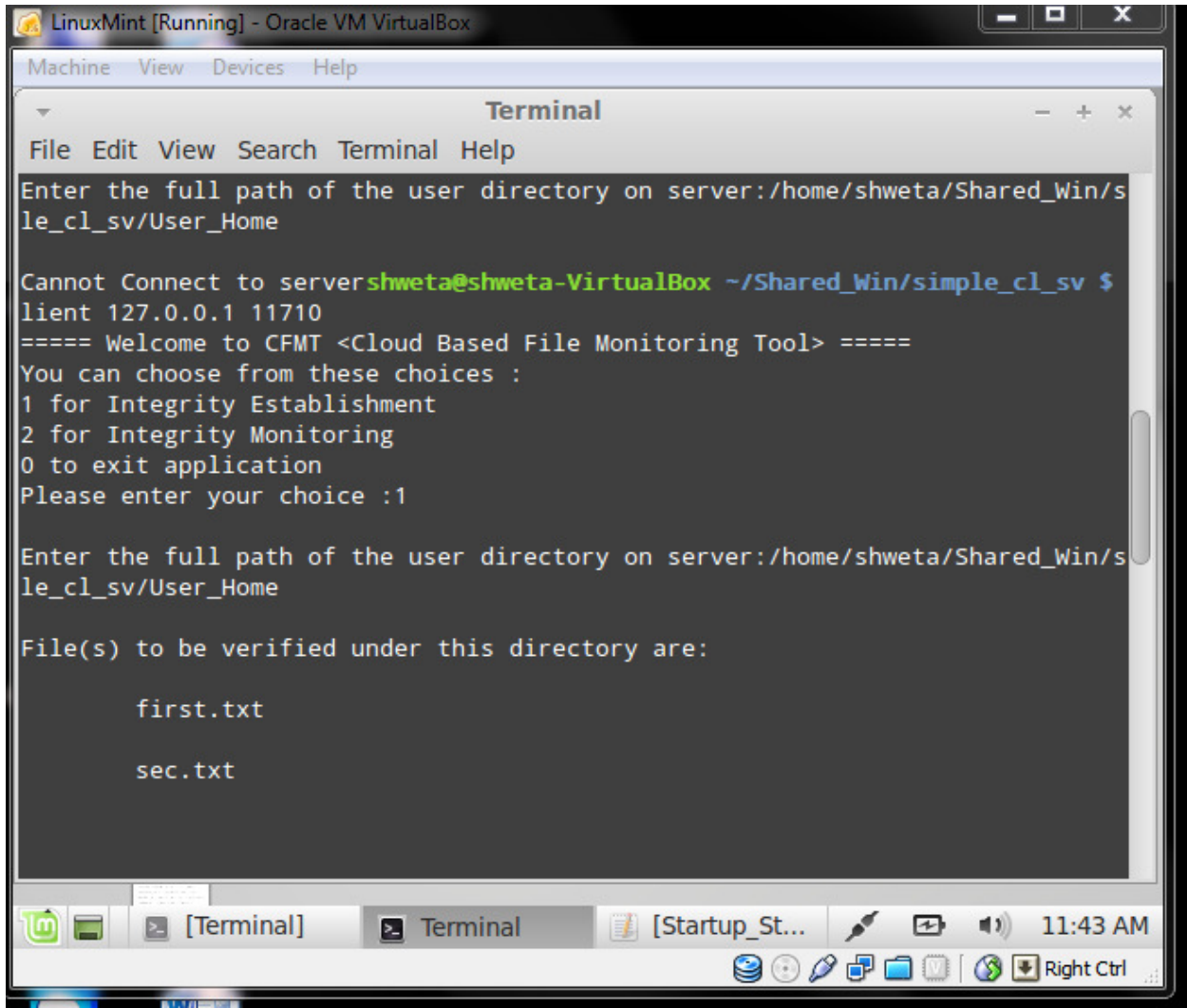
#### **4.1.1 OUTPUTS OF TOOL**

```
LinuxMint [Running] - Oracle VM VirtualBox
Machine View Devices Help
Terminal
File Edit View Search Terminal Help
[sudo] password for shweta:
shweta@shweta-VirtualBox ~ $ cd /home/shweta/Shared_Win
shweta@shweta-VirtualBox ~/Shared_Win $ cd simple_cl_sv/
shweta@shweta-VirtualBox ~/Shared_Win/simple_cl_sv $ ls
client      common.h  des.h     server    server.o  sph_types.h
client.c    common.o  des.o     server.c  sha2big.o User_Home
common.c    des.c     Makefile  server.h  sph_sha2.h
shweta@shweta-VirtualBox ~/Shared_Win/simple_cl_sv $ make client
make: `client' is up to date.
shweta@shweta-VirtualBox ~/Shared_Win/simple_cl_sv $ make server
make: `server' is up to date.
shweta@shweta-VirtualBox ~/Shared_Win/simple_cl_sv $ ./server
Server:waiting-----
Segmentation fault
shweta@shweta-VirtualBox ~/Shared_Win/simple_cl_sv $ ./server
Server:waiting-----
Finished Serving One Client
Server:waiting-----
Finished Serving One Client
Server:waiting-----
Finished Serving One Client
Server:waiting-----
Terminal [Startup_St... 11:47 AM
Right Ctrl
```

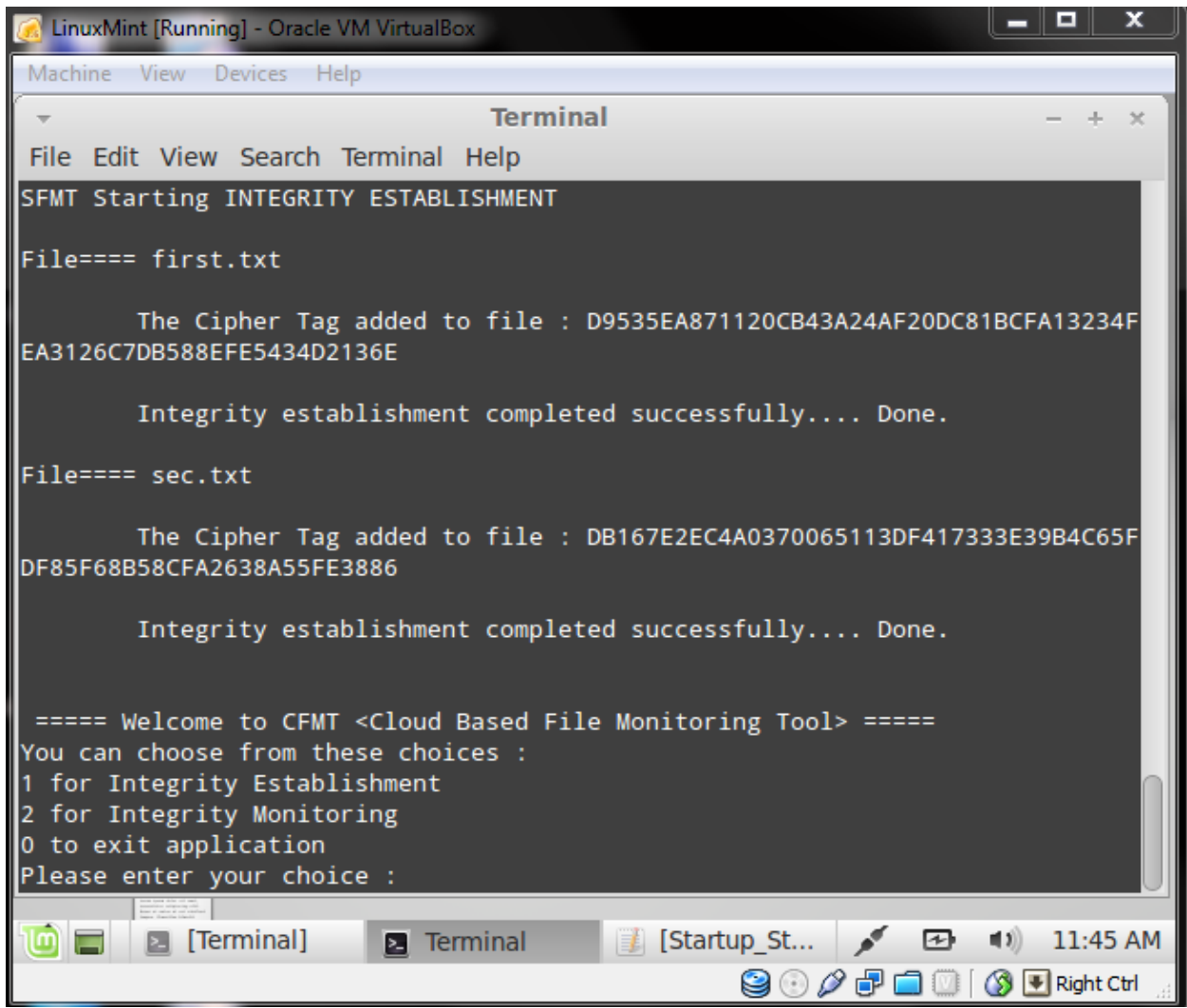
**Figure 15: Screenshot of Server Process during Initialization mode.**



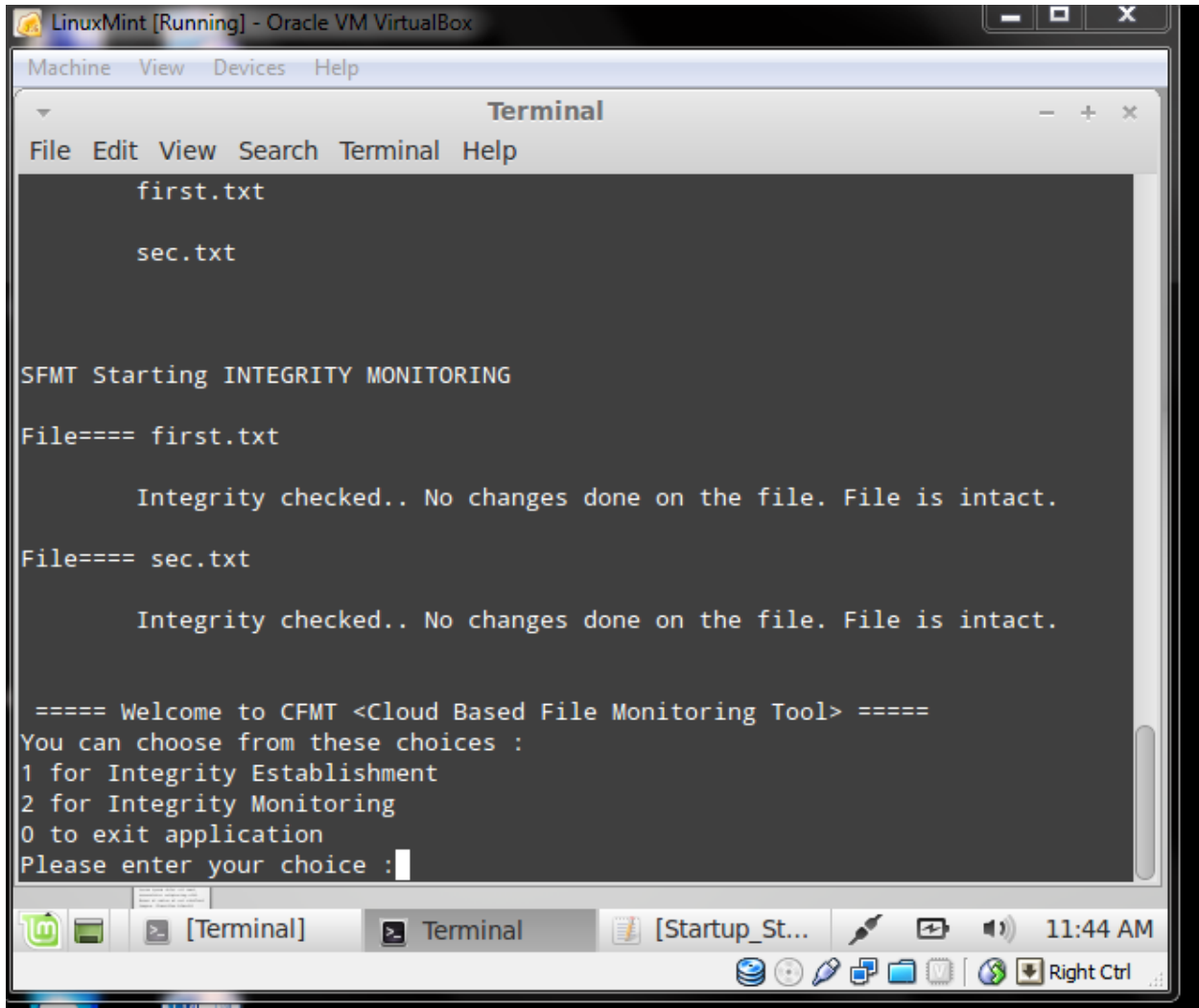
**Figure 16: Screenshot of Client Process during Initialization mode.**



**Figure 17: Screenshot of Client Process during Integrity Establishment.**



**Figure 18: Screenshot of Client Process during completion of Integrity Establishment.**



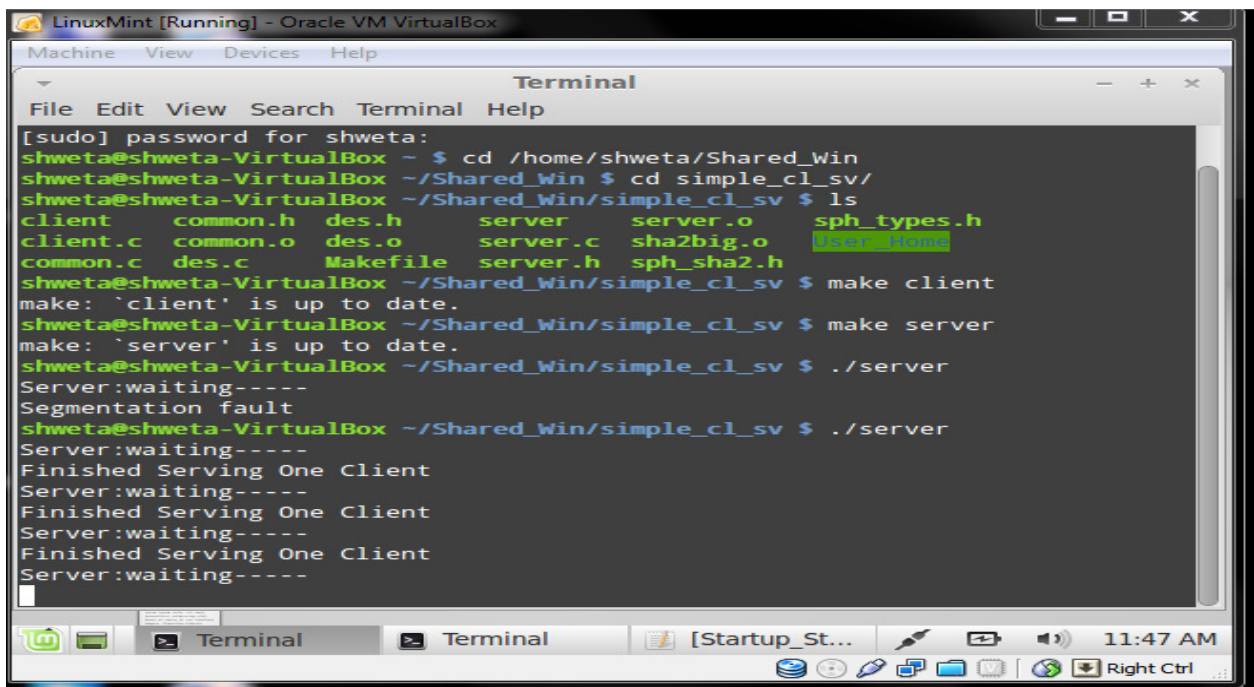
**Figure 19: Screenshot of Client Process during Integrity Monitoring.**

## 4.2 IMPLEMENTATION OF NEWLY PROPOSED FILE INTEGRITY MAINTENANCE TOOL FOR SECURE INFORMATION STORAGE UNDER CLOUD

This model has also been implemented in windows7 platform using Linux based Oracle VM virtual box. This tool provides the facility of client- server environment on existing machine only. There have been various processes including client and server processes. The server process executes and establishes connection with client process using specified socket address structures. There are two types of processes which gets generated i.e. client and server processes. Firstly, the server process executes and allows the client process to perform following functionalities-

- **Integrity Calculation**-This functionality calculates the final checksum value for the original file content and stores it in safe tags within the original file only.
- **Integrity Maintenance** -This function may be scheduled to perform integrity checks regularly for stored confidential client files (already stored by client on the server).

### 4.2.1 OUTPUTS OF NEW FILE INTEGRITY MAINTENANCE TOOL



```
[sudo] password for shweta:
shweta@shweta-VirtualBox ~ $ cd /home/shweta/Shared_Win
shweta@shweta-VirtualBox ~/Shared_Win $ cd simple_cl_sv/
shweta@shweta-VirtualBox ~/Shared_Win/simple_cl_sv $ ls
client      common.h   des.h      server     server.o   sph_types.h
client.c    common.o  des.o      server.c   sha2big.o  user_home
common.c    des.c     Makefile   server.h   sph_sha2.h
shweta@shweta-VirtualBox ~/Shared_Win/simple_cl_sv $ make client
make: `client' is up to date.
shweta@shweta-VirtualBox ~/Shared_Win/simple_cl_sv $ make server
make: `server' is up to date.
shweta@shweta-VirtualBox ~/Shared_Win/simple_cl_sv $ ./server
Server:waiting-----
Segmentation fault
shweta@shweta-VirtualBox ~/Shared_Win/simple_cl_sv $ ./server
Server:waiting-----
Finished Serving One Client
Server:waiting-----
Finished Serving One Client
Server:waiting-----
Finished Serving One Client
Server:waiting-----
```

Figure 20: Screenshot for Server Process Initialization Mode.

```
shweta@shweta-VirtualBox ~/Shared_Win/MajorPro_v1.0 $ ./client 127.0.0.1 11710 1
Debug logs enabled

===== Welcome to CLIENT - SERVER FILE INTEGRITY TOOL > =====
You can choose from these choices :
1 for Integrity CALCULATION
2 for Integrity MAINTENANCE
0 to exit application
Please enter your choice :1

Enter the full path of the user directory on server:cd Shared_Win/MajorPro_v1.0/
user_dir/

CHOICE PORT PATH
1 11720 cd
Number of files found in user directory : 3

1. File =====
```

**Figure21:Screenshot for Client Process Initialization Mode**



```

4. Applied Hash Algorithm          H(x) = H(MD + N + N')

5. Applied RSA Digital Signature.  E[H(x)^d mod n]
Sending this RSA Digital Signature to server:
778926cdb3e4b135ef1d4361cd1a9159222f4e10210337441db61b41d31282934d144110442044d7
311a10111cd8bce9251081a84d88525118121182552041b8f95c7175841881eb13465

```

**Figure 22: Screenshot for server process during Integrity Establishment Process for file.**

```

LinuxMint [Running] - Oracle VM VirtualBox
Machine View Devices Help
Terminal
File Edit View Search Terminal Help
Number of files found in user directory : 2

1. File ==== f.txt.backup
2. Applied DES Decryption          (MD + N)
3. Added Nonce N'                 (MD + N + N')
4. Applied Hash Algorithm          H(x) = H(MD + N + N')
5. Applied RSA Digital Signature.  E[H(x)^d mod n]
Client Verified
File intact.

2. File ==== s.txt.backup
2. Applied DES Decryption          (MD + N)
3. Added Nonce N'                 (MD + N + N')
4. Applied Hash Algorithm          H(x) = H(MD + N + N')
5. Applied RSA Digital Signature.  E[H(x)^d mod n]
Client Verified
File intact.

```

**Figure 23: Screenshot for server Process during Integrity Monitoring for Files.**

```

LinuxMint [Running] - Oracle VM VirtualBox
Machine View Devices Help
Terminal
File Edit View Search Terminal Help

2. File ==== second.txt
2. Applied Hash on file content. Generated MD.
3. Added Nonce 'N'. Generated (MD + N)
4. Applied DES Encryption. Ek(MD + N).
5. Verifying client: H(x) = D^e mod n
6. Client Verified.
File intact.
+++++
+++++
Performance Analysis
Function File FileSize(Bytes) TimeElapsed(us) TimeEl
integrityEstablishment first.txt 352 1090845
integrityEstablishment second.txt 313 1076547
+++++
+++++
----- Server:Waiting -----
No. of files found : 2

```

**Figure 24: Screenshot of Performance Analysis (Time consumed during Integrity Calculation) for a given File.**

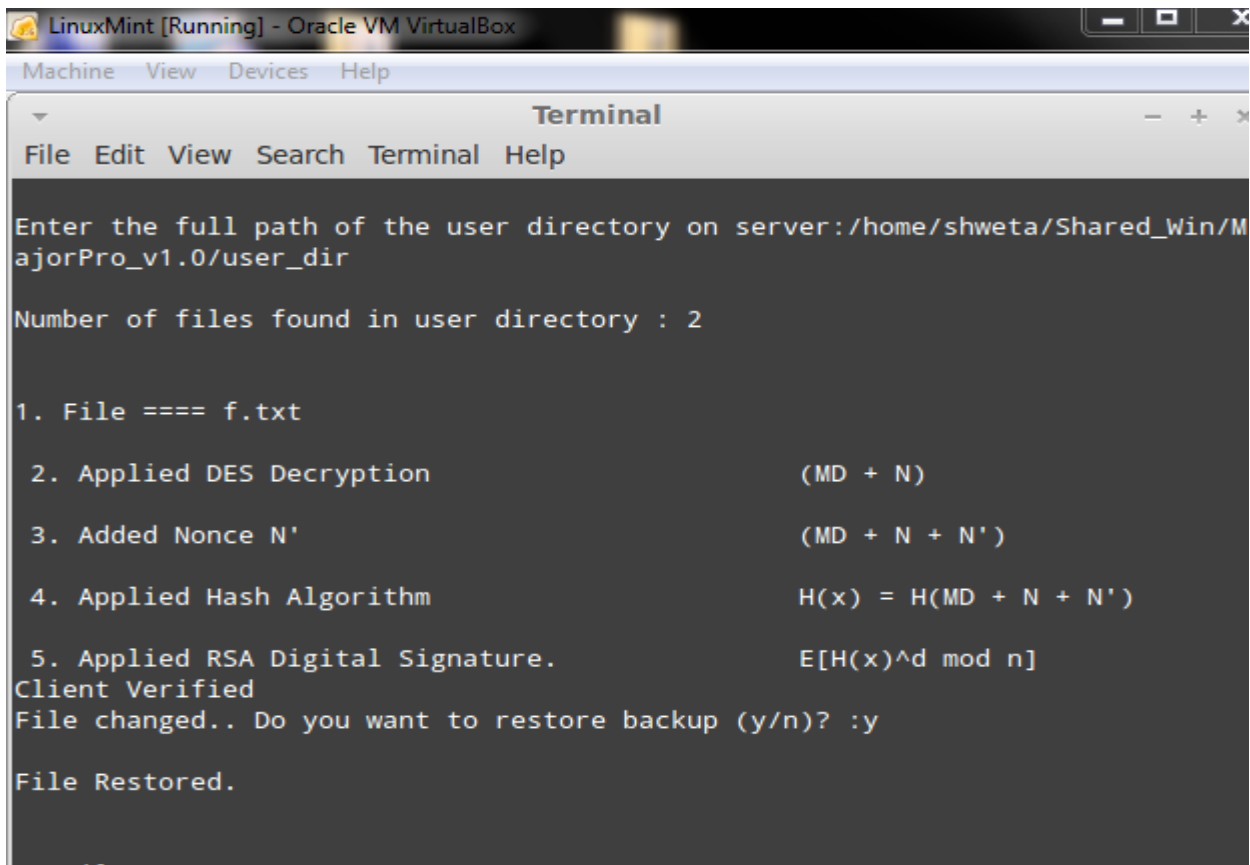
```

LinuxMint [Running] - Oracle VM VirtualBox
Machine View Devices Help
Terminal
File Edit View Search Terminal Help

1. File ==== second.txt
2. Applied Hash on file content. Generated MD.
3. Added Nonce 'N'. Generated (MD + N)
4. Applied DES Encryption. Ek(MD + N).
5. Verifying client: H(x) = D^e mod n
6. Client Verified.
File intact.
+++++
+++++
Performance Analysis
Function File FileSize(Bytes) TimeElapsed(us) TimeEl
integrityMonitoring first.txt 352 1102001
integrityMonitoring second.txt 313 1030882
+++++
+++++

```

**Figure 25: Screenshot of Performance Analysis (Time consumed during Integrity Maintenance) for a given file.**



```
LinuxMint [Running] - Oracle VM VirtualBox
Machine View Devices Help

Terminal
File Edit View Search Terminal Help

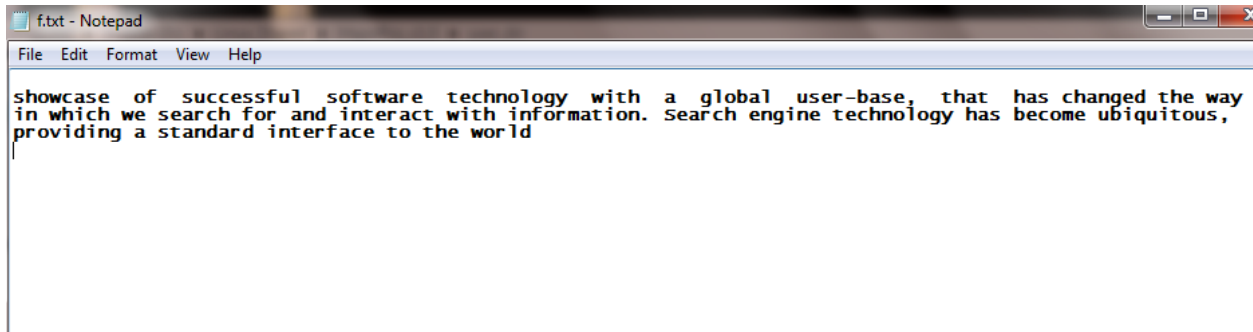
Enter the full path of the user directory on server:/home/shweta/Shared_Win/M
ajorPro_v1.0/user_dir

Number of files found in user directory : 2

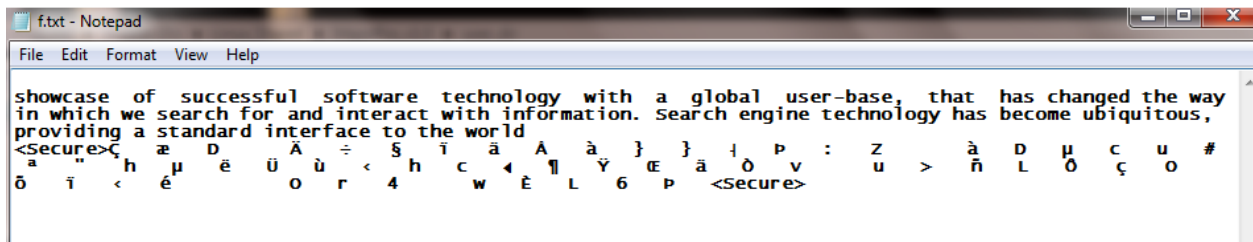
1. File ==== f.txt
2. Applied DES Decryption (MD + N)
3. Added Nonce N' (MD + N + N')
4. Applied Hash Algorithm H(x) = H(MD + N + N')
5. Applied RSA Digital Signature. E[H(x)^d mod n]
Client Verified
File changed.. Do you want to restore backup (y/n)? :y

File Restored.
```

**Figure 26: Screenshot of file restoration functionality on any kind of client information modification within server.**



**Figure 27: Screenshot of Original client's data stored on server.**



**Figure 28: Screenshot of Original Client's data stored after Integrity Calculation (stored within shown tags).**

### 4.3 COMPARISON BASED ON PERFORMANCE AMONG VARIOUS DISCUSSED FILE INTEGRITY MODELS

The following table 6 represents distinction among various file integrity models and newly presented file integrity model-

S.NO	PARAMETER CRITERIA	FILE MONITORING MODEL FOR SECURING FILES[8][9]	INTRUSION DETECTION&IN-KERNEL INTEGRITY CHECKER [12][13]	FLOGGER:AN INTRUSION DETECTION &FILE CENTRIC [15]	NEWLY PRESENTED FILE INTEGRITY MAINTENANCE TOOL
1.	<b>CostEffective Solution</b>	Yes	No	No	Yes
2.	<b>LowMemory Requirements</b>	Yes	No	No	Yes
3.	<b>Low/No Database Requirement</b>	Yes	No	No	Yes
4.	<b>Time Efficient</b>	Yes	No	No	Yes
5.	<b>Client-Server Trust-Level Relationship</b>	No	No	No	Yes
6.	<b>LowServer Overhead</b>	Yes	No	No	Yes
7.	<b>Client Involvement during Integrity Checks</b>	No	No	No	Yes
8.	<b>File Restoration capability after outsider/thwart attacks</b>	Yes	No	No	Yes
9.	<b>No Security &amp; Cachepolicy Requirement</b>	Yes	No	No	Yes
10.	<b>No Physical &amp; Virtual Machines Log based information requirements</b>	Yes	Yes	No	Yes

## CHAPTER 5

### CONCLUSION & FUTURE WORK

For confidential client data, an enhanced security model is required which can provide high level security during information storage under cloud platform. This model has been proposed which can be set-up in a cloud computing environment and is very cost effective and highly secure for client's data. It ensures trust mechanism between both entities during interaction (which is a desirable advantage to promote cloud businesses). This tool can work in form of scheduler/executable to be run on cloud server at customized time/day sequences. It benefits using regular integrity checks and thus supervises the intact integrity of the client's information over the cloud with better client –server communications .Thus, it can be observed that this tool proves to be quite time efficient and cost effective as it performs integrity calculation in around 1000 microseconds for a given 500 byte file which is a very limited requirement as compared to the long delays(in minutes/hours) required for server to perform integrity checks on client's file. Andalso, there is no memory constraint in this tool. This tool works without any database requirement on the server. Thus, satisfies the light weight and secure storage tool criteria for cloud computing environment. In this project, we have proposed two frameworks:-

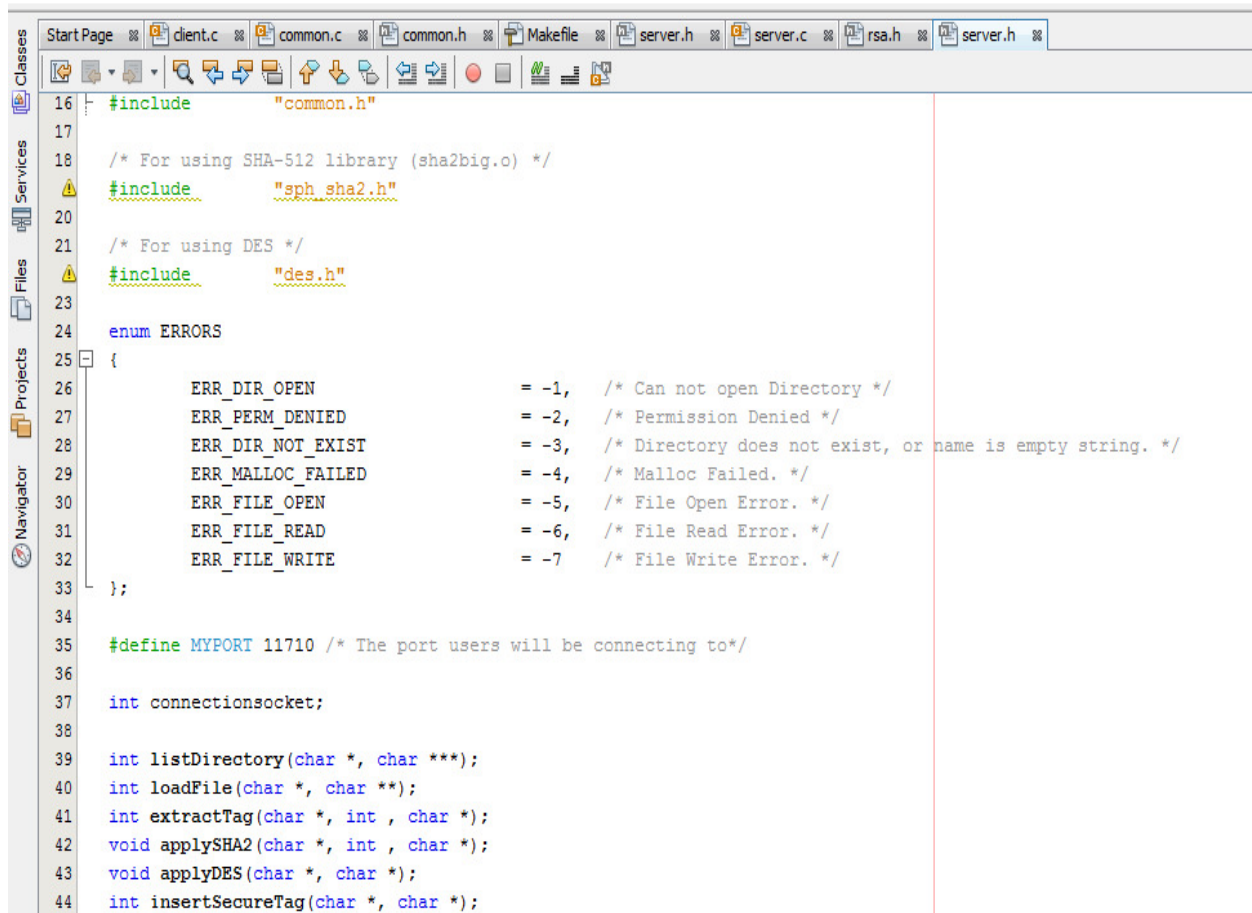
- A Light weight Storage security based concept has been incorporated which includes data Integrity Calculation, Maintenance& Client- Server trust relationship & involvement.
- Another framework has been proposed for highly confidential data in Cloud computing which meets the desired security characteristics, required for secure information transmission over the cloud computing network.

With reference to future work in cloud security, various different security issues already discussed may be addressed and efficient approaches may be suggested to improve the overall cloud computing scenario and enable it to become a better platform for business purposes.

## APPENDIX

### A. SNAPSHOTS OF SOURCE CODE OF FORMER LIGHT WEIGHT FILE INTEGRITY TOOL-

This model has been implemented in windows7 platform using Linux based Oracle VM virtualbox. This tool provides the facility of client- server environment on existing machine only. The following snapshots have been produced using Net Beans IDE during compilation of code. There have been various headers and client/server source code files included in this project as shown below:



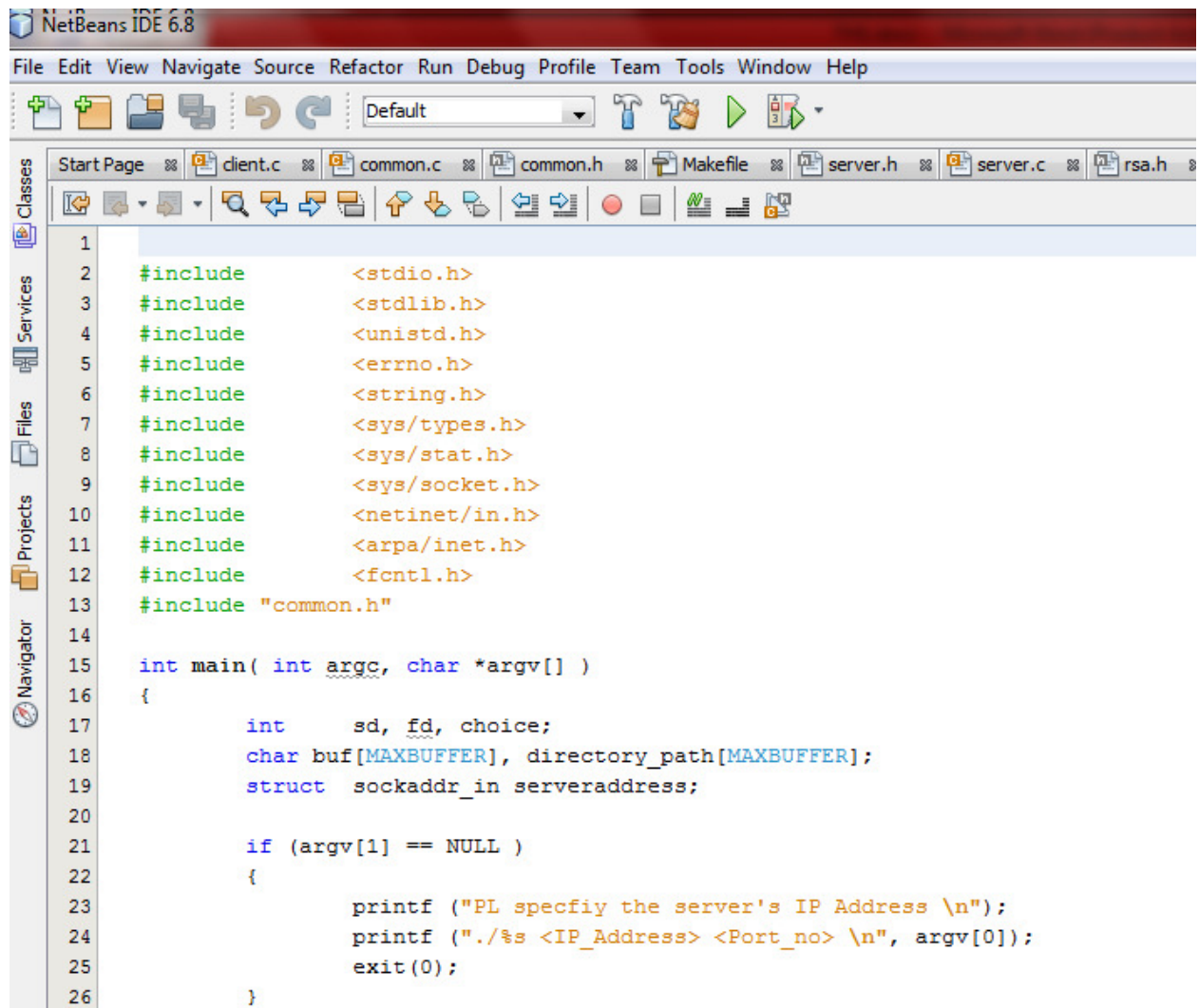
```
16 #include "common.h"
17
18 /* For using SHA-512 library (sha2big.o) */
19 #include "sph_sha2.h"
20
21 /* For using DES */
22 #include "des.h"
23
24 enum ERRORS
25 {
26     ERR_DIR_OPEN = -1, /* Can not open Directory */
27     ERR_PERM_DENIED = -2, /* Permission Denied */
28     ERR_DIR_NOT_EXIST = -3, /* Directory does not exist, or name is empty string. */
29     ERR_MALLOC_FAILED = -4, /* Malloc Failed. */
30     ERR_FILE_OPEN = -5, /* File Open Error. */
31     ERR_FILE_READ = -6, /* File Read Error. */
32     ERR_FILE_WRITE = -7 /* File Write Error. */
33 };
34
35 #define MYPORT 11710 /* The port users will be connecting to*/
36
37 int connectionsocket;
38
39 int listDirectory(char *, char ***);
40 int loadFile(char *, char **);
41 int extractTag(char *, int , char *);
42 void applySHA2(char *, int , char *);
43 void applyDES(char *, char *);
44 int insertSecureTag(char *, char *);
```

Figure 29: Screenshot of server header source file



```
287 *
288 * DESCRIPTION : Establish integrity on user files.
289 *****/
290 void integrityEstablishment(char *pathToUserDir)
291 {
292     char **list = NULL;
293     char fullPath[100];
294     char *fileContent;
295     char tag[65] = {0};
296     char sha2Output[65] = {0};
297     char encryptedText[65] = {0};
298     int fileCount = 0;
299     int i;
300     int retVal;
301     int fileSize;
302
303     fileCount = listDirectory(pathToUserDir, &list);
304     if (0 > fileCount)
305     {
306         socketWrite("No files \n");
307         return;
308     }
309
310     socketWrite("File(s) to be verified under this directory are: \n");
311     for (i = 0; i < fileCount; i++)
312     {
313         socketWrite("%s \n" *(list + i));
```

**Figure 30: Screenshot of server source file**



The screenshot displays the NetBeans IDE 6.8 interface. The title bar reads "NetBeans IDE 6.8". The menu bar includes "File", "Edit", "View", "Navigate", "Source", "Refactor", "Run", "Debug", "Profile", "Team", "Tools", "Window", and "Help". The toolbar contains icons for file operations and a "Default" dropdown menu. The window title bar shows several open files: "Start Page", "client.c", "common.c", "common.h", "Makefile", "server.h", "server.c", and "rsa.h". The left sidebar features a "Navigator" panel with icons for "Classes", "Services", "Files", and "Projects". The main editor area shows the source code for "client.c" with line numbers 1 through 26. The code includes several system and project headers and defines a main function.

```
1
2  #include      <stdio.h>
3  #include      <stdlib.h>
4  #include      <unistd.h>
5  #include      <errno.h>
6  #include      <string.h>
7  #include      <sys/types.h>
8  #include      <sys/stat.h>
9  #include      <sys/socket.h>
10 #include      <netinet/in.h>
11 #include      <arpa/inet.h>
12 #include      <fcntl.h>
13 #include      "common.h"
14
15 int main( int argc, char *argv[] )
16 {
17     int    sd, fd, choice;
18     char  buf[MAXBUFFER], directory_path[MAXBUFFER];
19     struct sockaddr_in serveraddress;
20
21     if (argv[1] == NULL )
22     {
23         printf ("PL specfiy the server's IP Address \n");
24         printf ("./%s <IP_Address> <Port_no> \n", argv[0]);
25         exit(0);
26     }
```

**Figure 31: Screenshot of client source file**

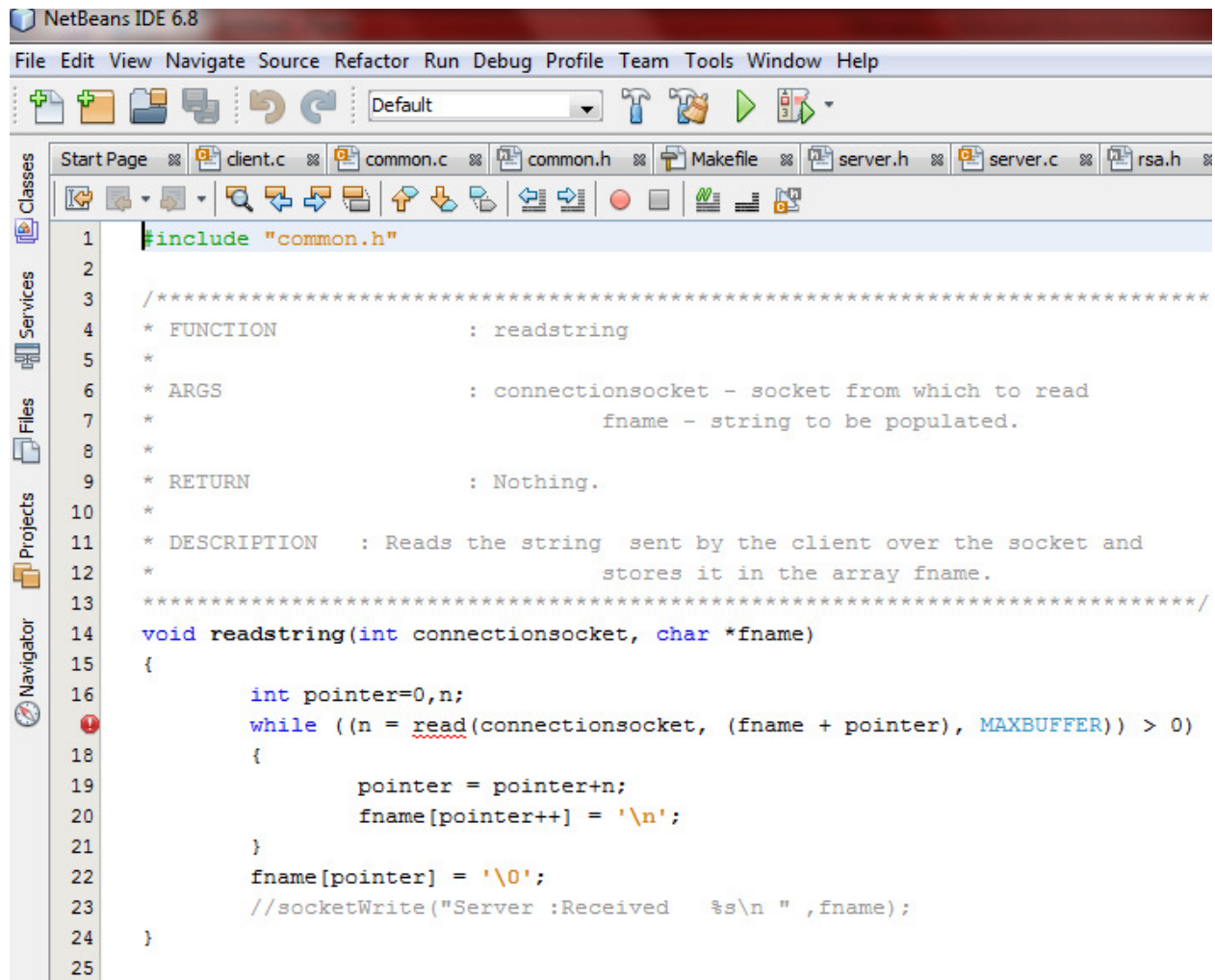
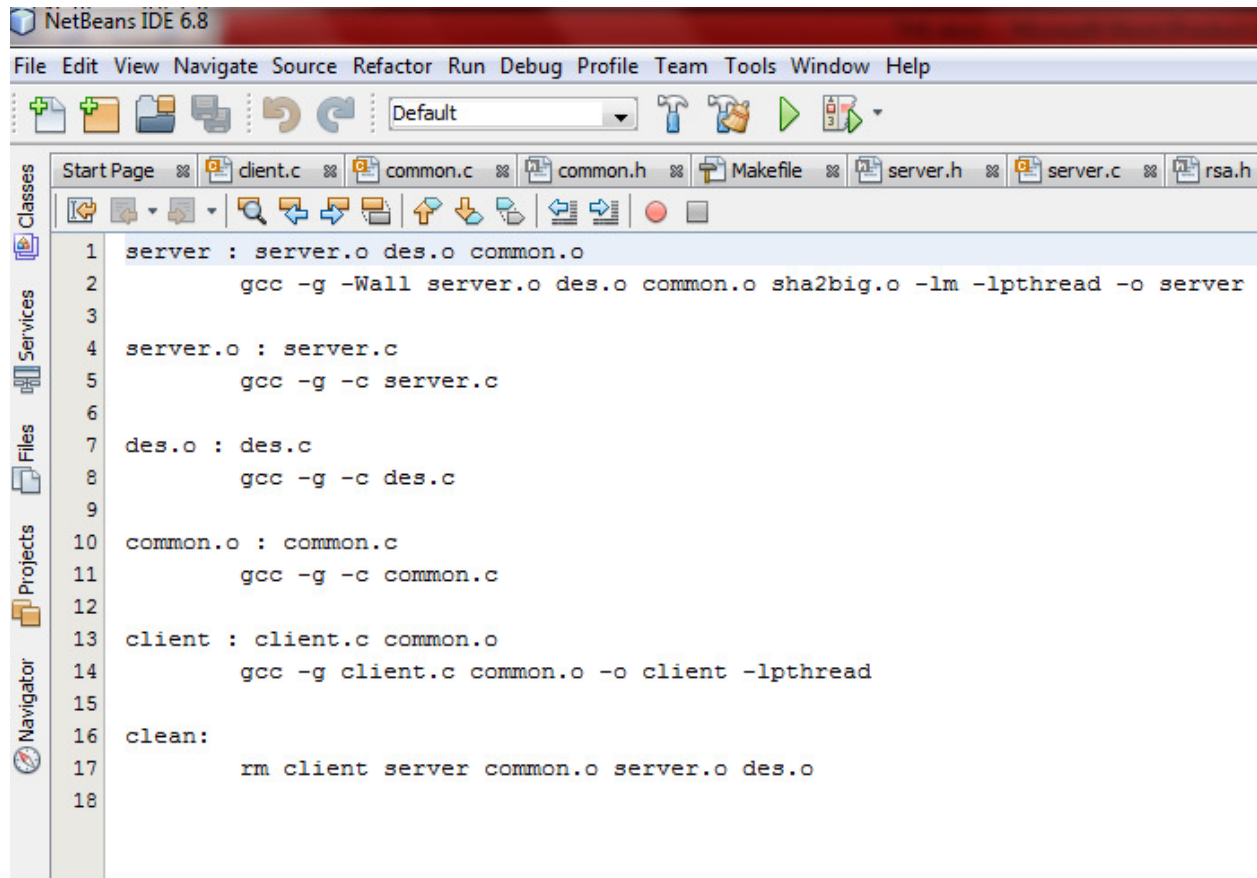
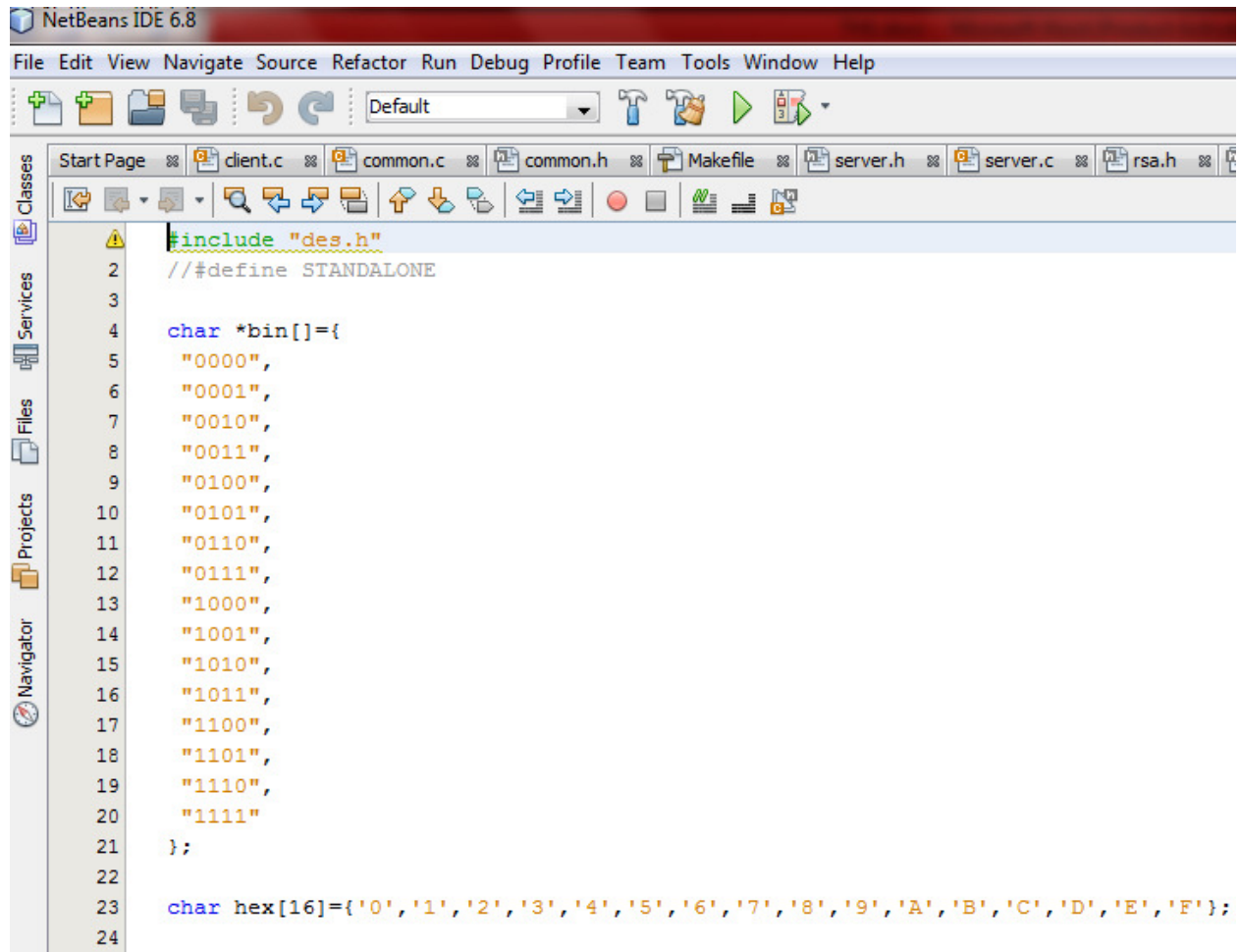


Figure 32: Screenshot of common source file



**Figure 33: Screenshot of Makefile source file**



**Figure 34: Screenshot of des source file**

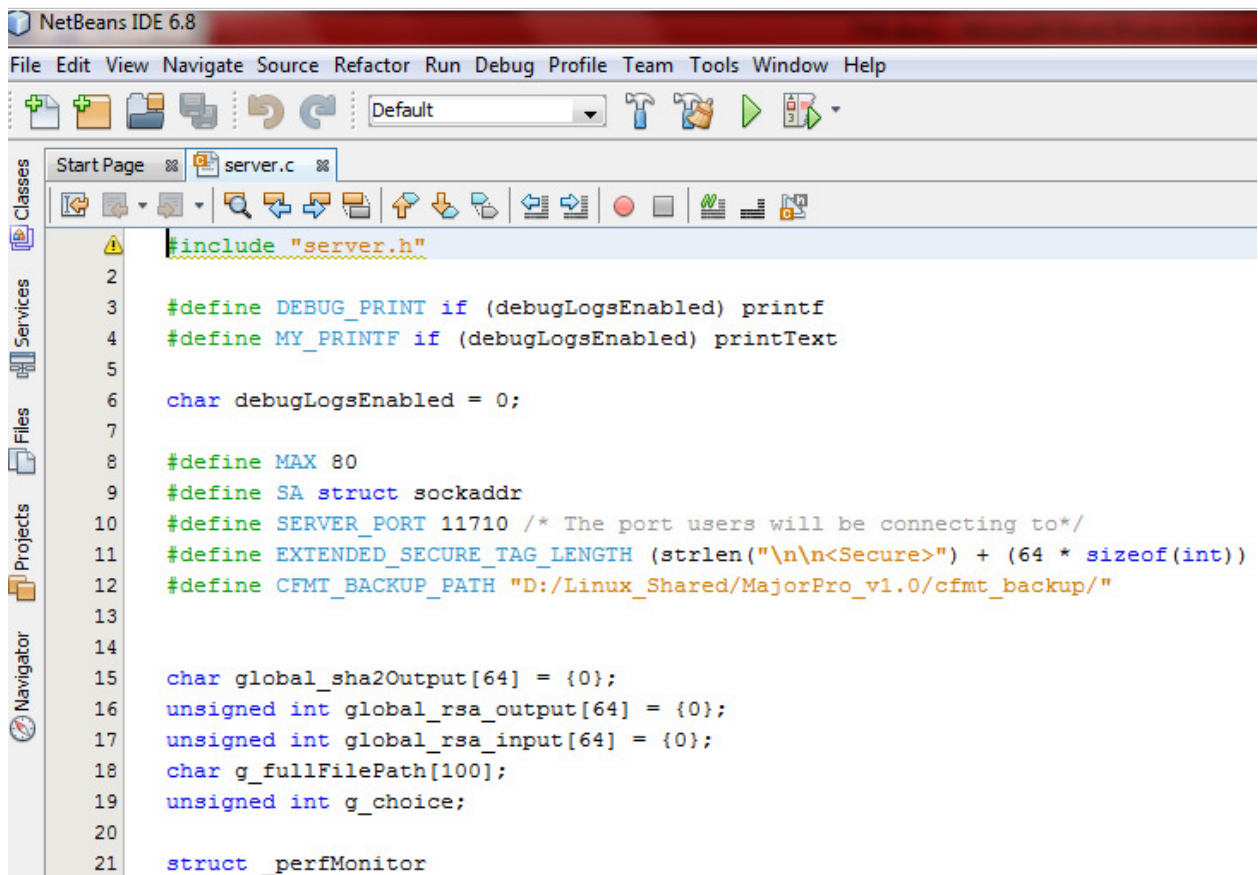
```
1  #include<stdio.h>
2  #include<string.h>
3  #include<malloc.h>
4  #include<stdlib.h>
5  #include<math.h>
6
7  void hex_to_bin(char *,char *);
8  char* bin_to_hex(char *);
9  void permutation(char *,char *);
10 void make_half(char *,char *,char *);
11 void single_shift(char *,char *);
12 void double_shift(char *,char *);
13 void make_key(char *,char *,char *);
14 void permutation_32(char *,char *);
15 void permutation_48(char *,char *);
16 void permutation_64(char *,char *,char *);
17
18 void des_round(char *,char *,char *,char *,char *,char *,char *);
19 void des_round_decry(char *,char *,char *,char *,char *,char *,char *);
20 void copy(char *,char *);
21 void permut_48(char *,char *);
22 void xor(char *,char *,char *);
23 void xor_32(char *,char *,char *);
24 void common_permutation(char *,char *);
25 void hex_to_plain(char *,char *,int);
26 int switch_case(char );
27 void DES_encrypt(char *, char *);
```

Figure 35: Screenshot of des header source file



## B. SNAPSHOTS OF SOURCE CODE OF NEWLY PROPOSED LIGHT WEIGHT FILE INTEGRITY TOOL

This model also has been implemented in windows7 platform using Linux based Oracle VM virtualbox. There are two types of processes which gets generated i.e. client and server processes. The following snapshots have been produced using Net Beans IDE during compilation of code. This also contains the client/server functionalities in source code files along with header files included in this project. These are shown as below:



```
NetBeans IDE 6.8
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help
Default
Start Page server.c
Classes
Services
Files
Projects
Navigator
#include "server.h"
2
3 #define DEBUG_PRINT if (debugLogsEnabled) printf
4 #define MY_PRINTF if (debugLogsEnabled) printText
5
6 char debugLogsEnabled = 0;
7
8 #define MAX 80
9 #define SA struct sockaddr
10 #define SERVER_PORT 11710 /* The port users will be connecting to*/
11 #define EXTENDED_SECURE_TAG_LENGTH (strlen("\n\n<Secure>") + (64 * sizeof(int)))
12 #define CFMT_BACKUP_PATH "D:/Linux_Shared/MajorPro_v1.0/cfmt_backup/"
13
14
15 char global_sha2Output[64] = {0};
16 unsigned int global_rsa_output[64] = {0};
17 unsigned int global_rsa_input[64] = {0};
18 char g_fullFilePath[100];
19 unsigned int g_choice;
20
21 struct _perfMonitor
```

Figure 36: Screenshot of server.c source code file.

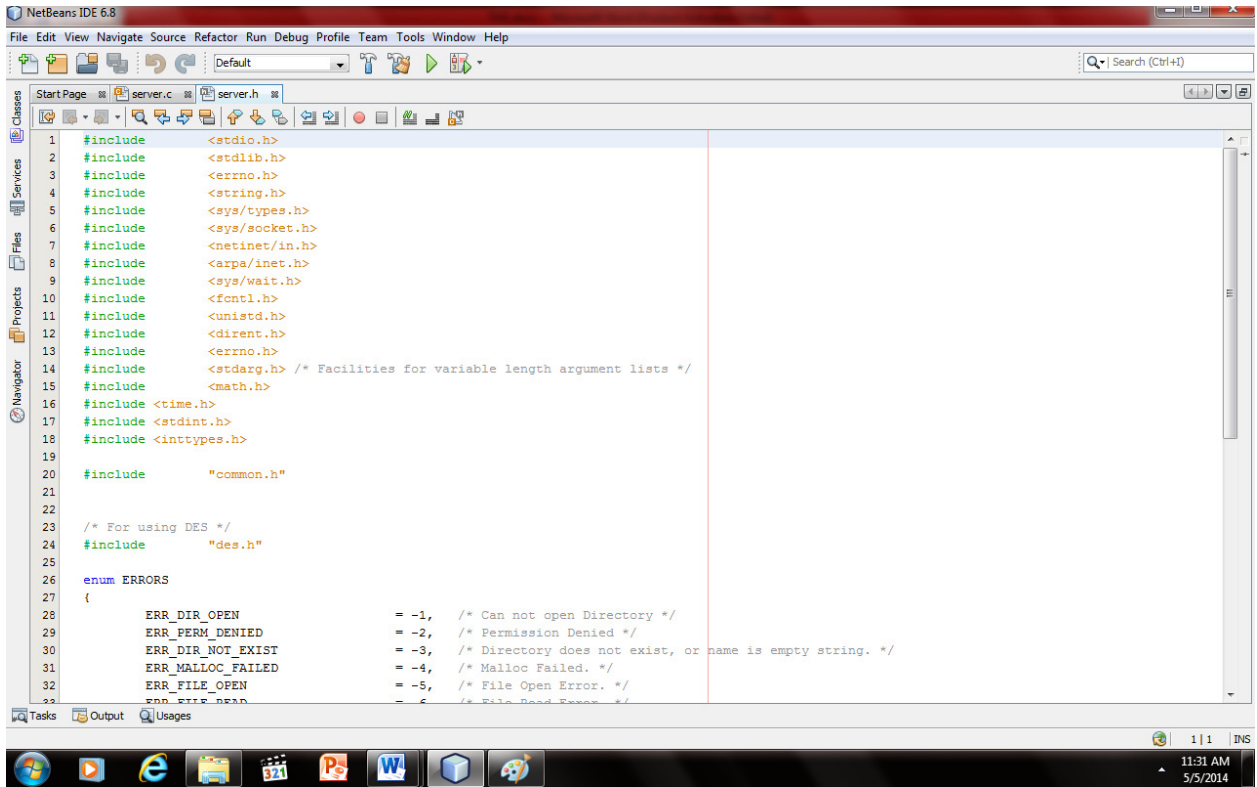
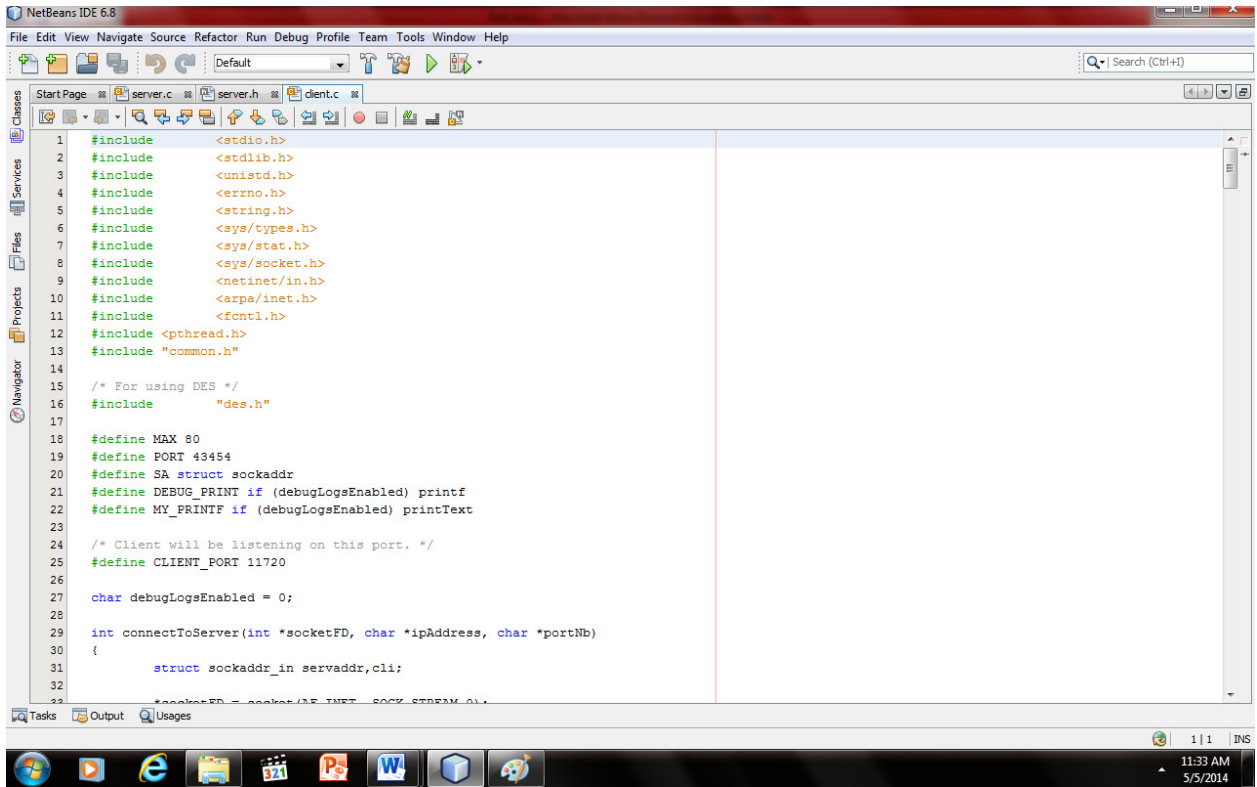
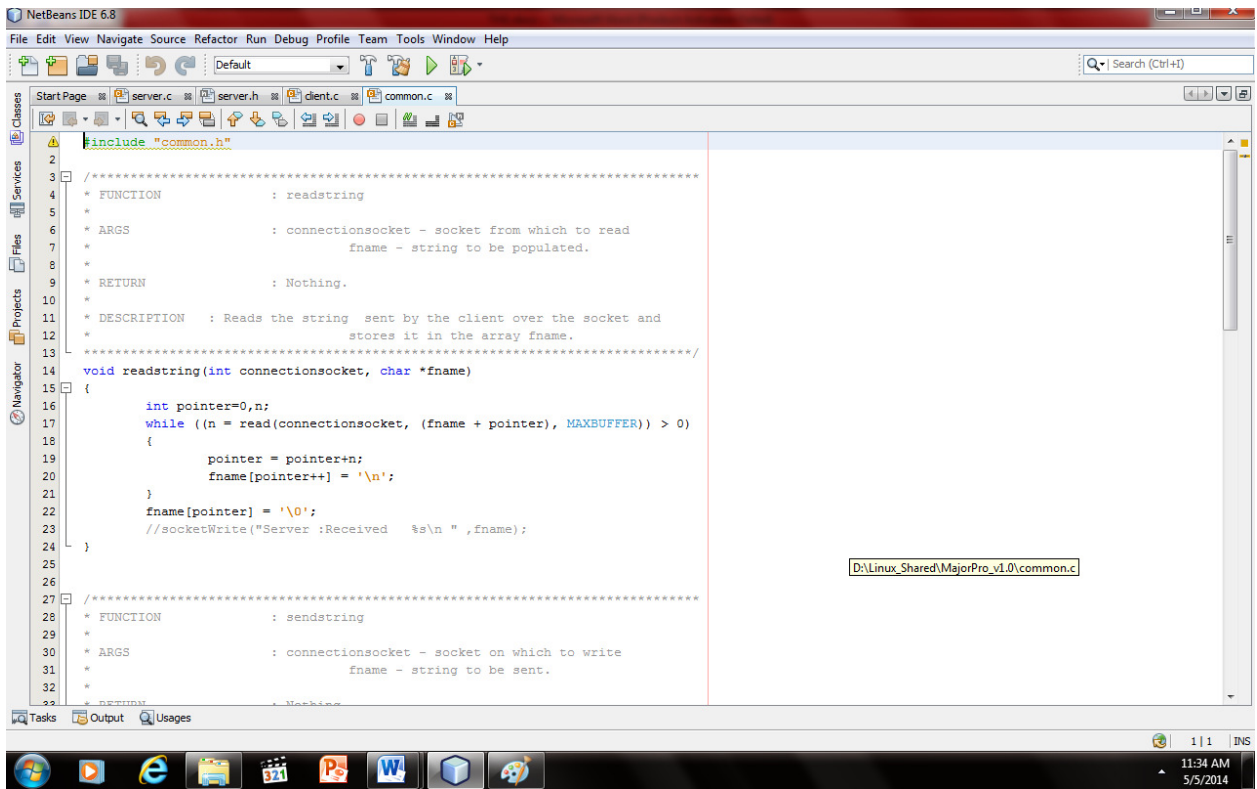


Figure 37: Screenshot of server header source code file.





**Figure 38: Screenshot of client source code file.**



**Figure 39: Screenshot of common source code file.**

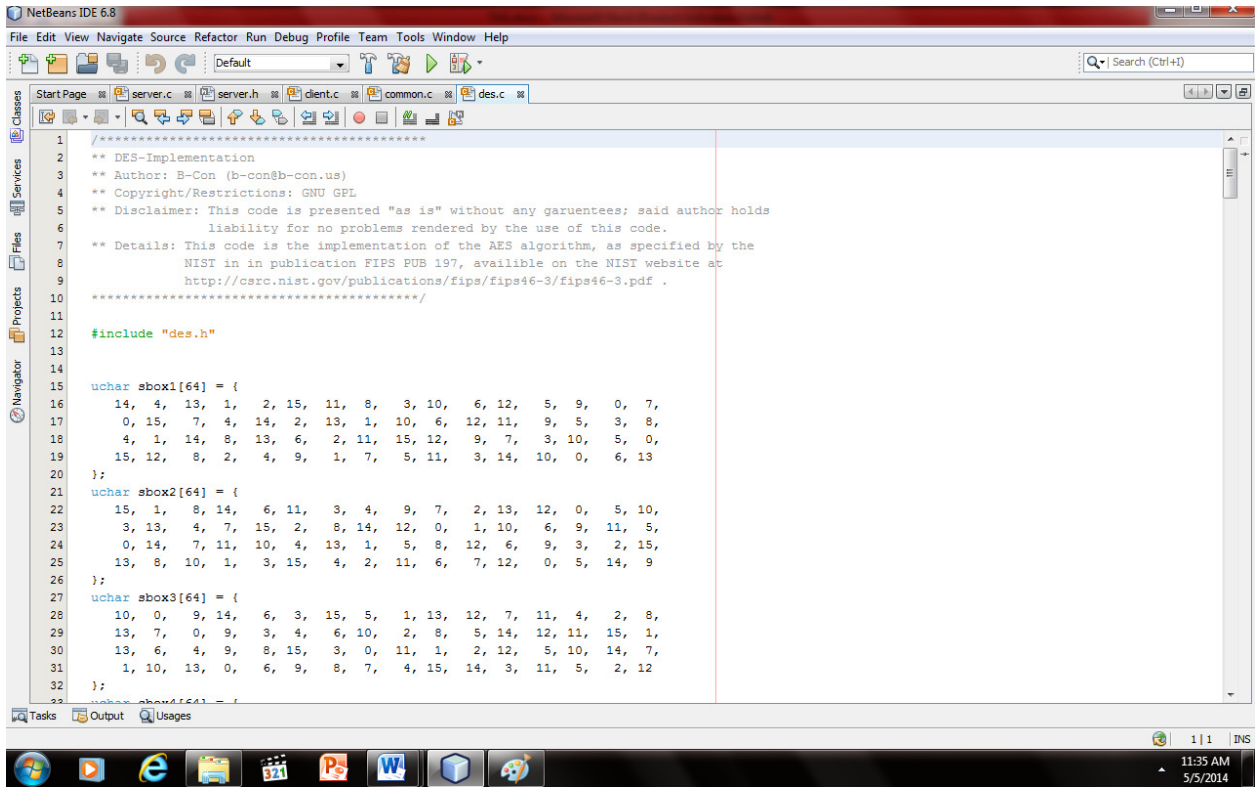


Figure 40: Screenshot of des source code file.

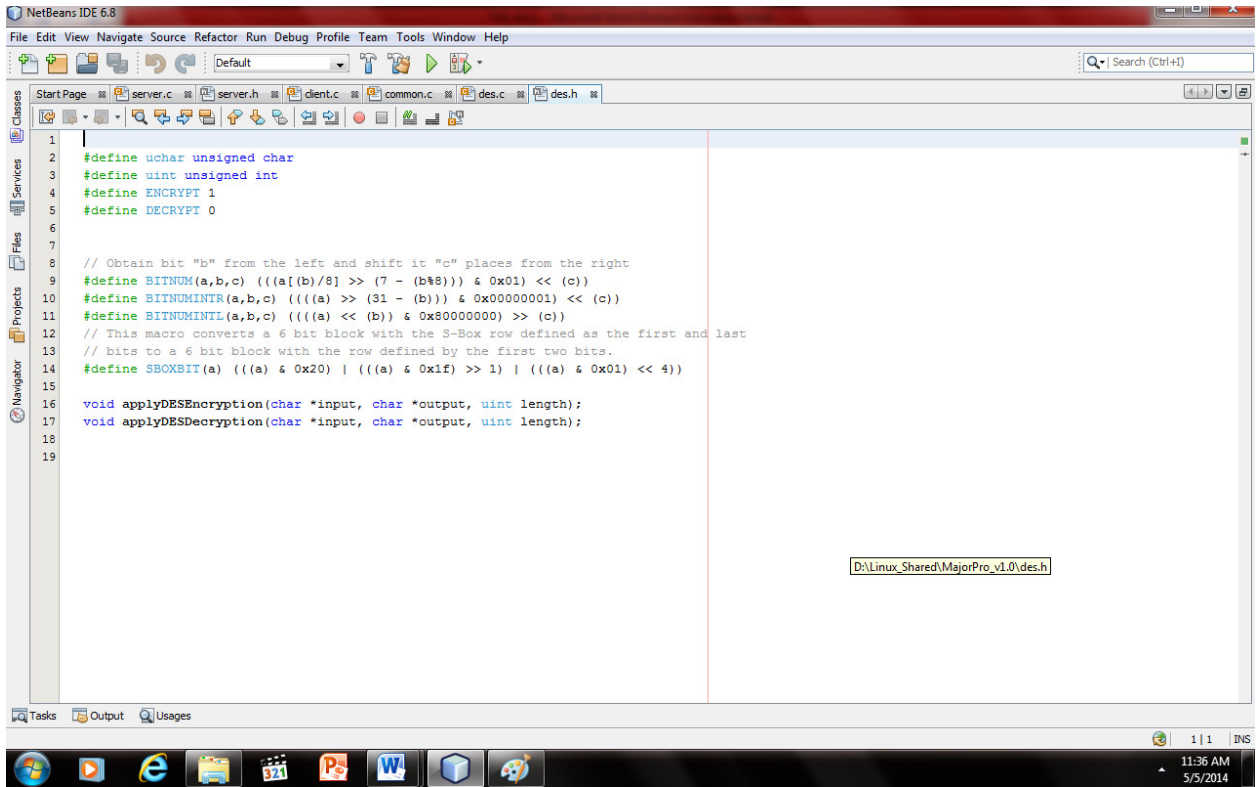


Figure 41: Screenshot of des header source code file.

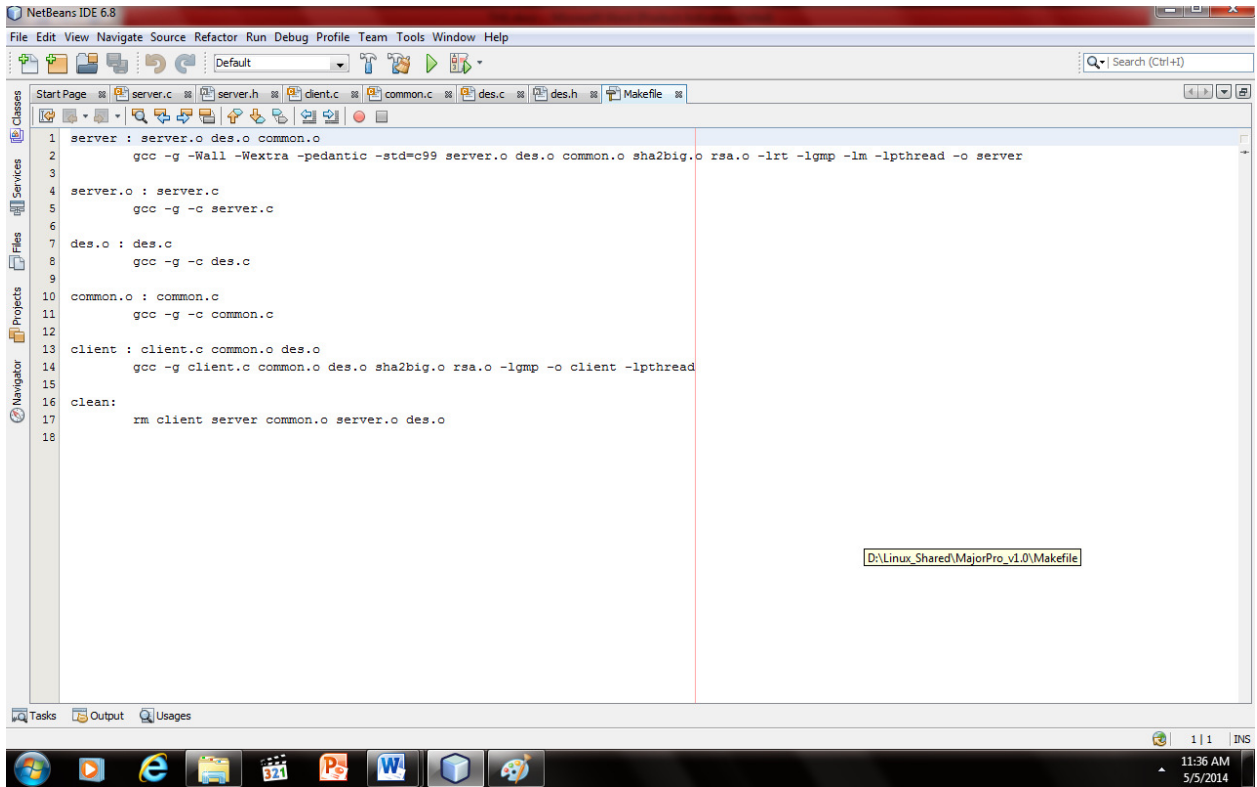
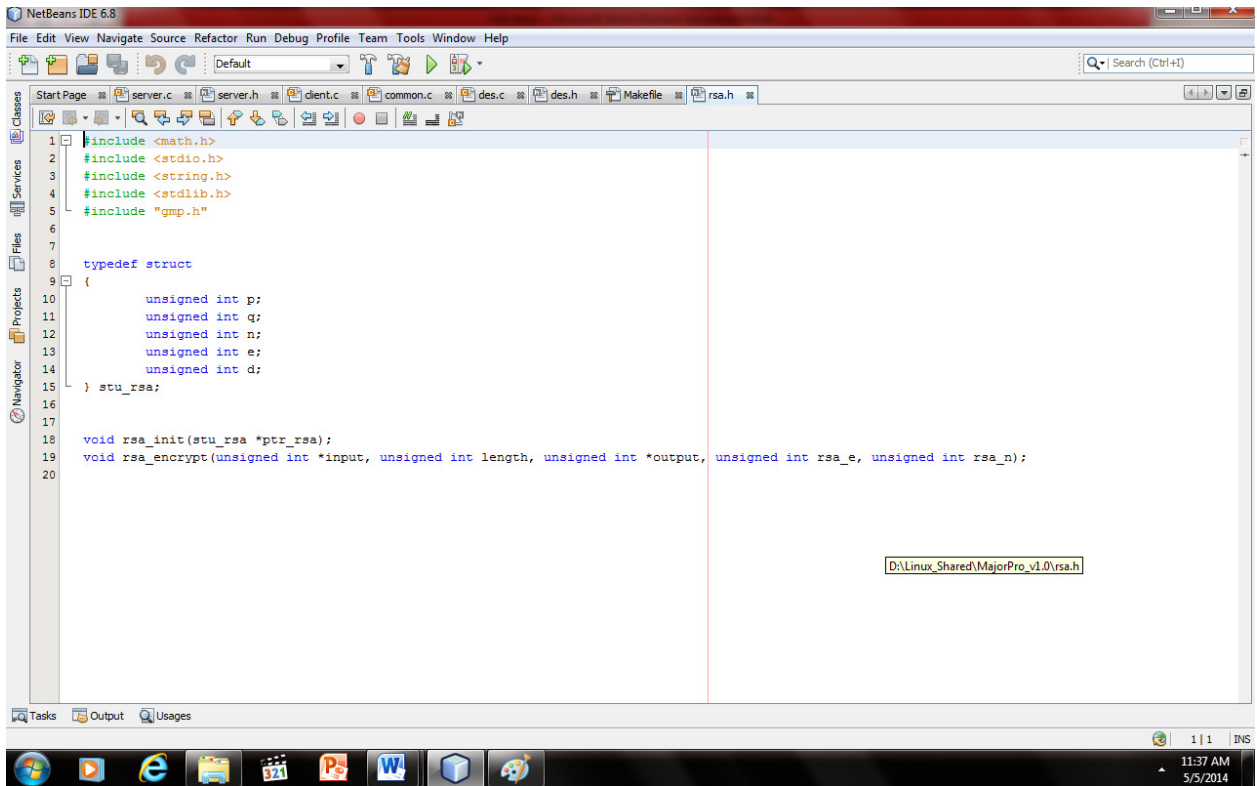


Figure 42: Screenshot of Makefile source code file.



**Figure 43: Screenshot of RSA header source code file.**

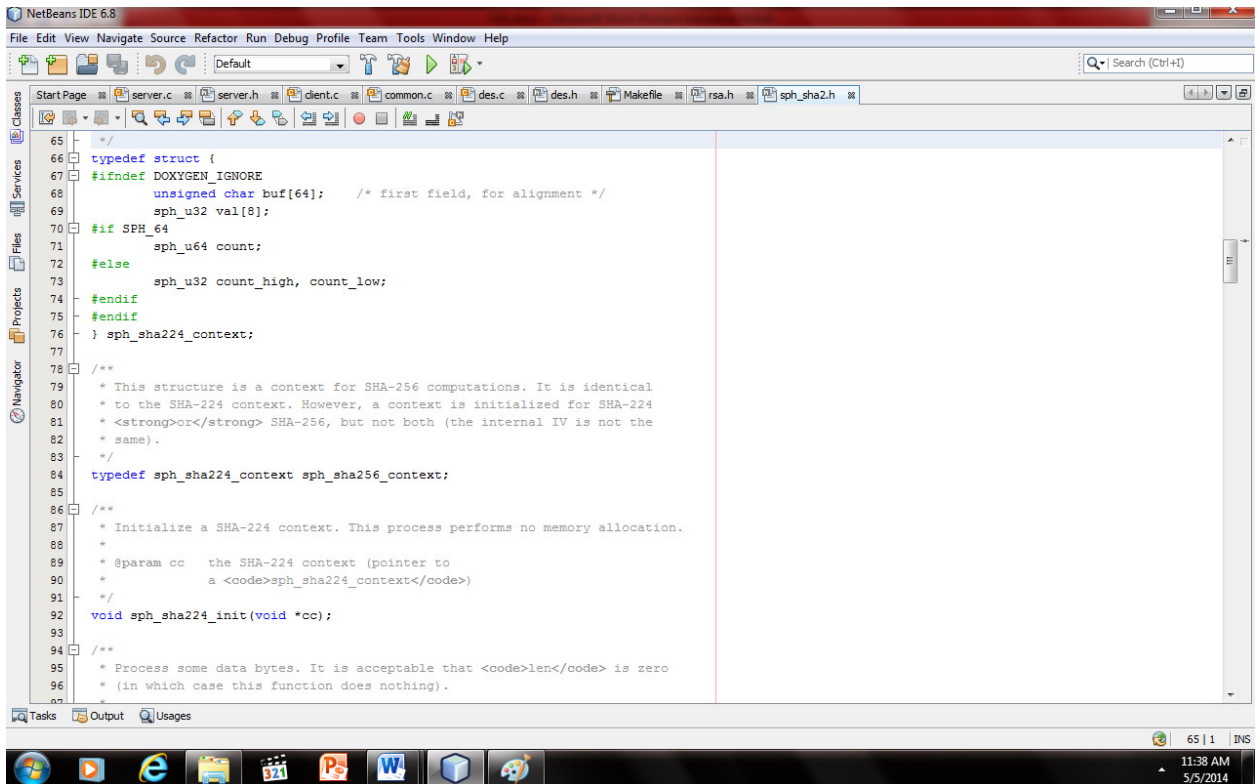


Figure 44: Screenshot of SHA2 header source code file.

## REFERENCES

- [1]**GASolanki**, “Welcome to the Future of Computing: Cloud Computing and Legal Issues”, International Journal of Scientific & Technology Research 2012, vol1, issue 9.
- [2]**JianWang, Yan Zhao, Shuo Jiang, Jiajin Le**,“Providing Privacy Preserving in Cloud Computing”2010, IEEE.
- [3] **Nelson Gonzalez, Charles Miers, Fernando Redigolo, Marcos Simplicio, Tereza Carvalho, Mats Naslund, Makan Pourzandi**, “A quantitative analysis of current security concerns and solutions for cloud computing”, Gonzalez et al. Journal of Cloud Computing: Advances, Systems and Applications 2012.
- [4] **Lijun Mei, W.K.Chan, T.H.Tse**, “A Tale of Clouds: Paradigm comparisons and some thoughts on research issues”, 2008 IEEE Asia-Pacific Services Computing Conference.
- [5] **BalachandraReddyKandukuri,Ramakrishna Paturi V,Dr.AtanuRakshit**, “Cloud security Issues”2009, IEEE.
- [6]**M.Sudha, M.Monika**, “Enhanced Security Framework to Ensure Data Security in Cloud Computing Using Cryptography”, Advances in Computer Science and its Applications 32 Vol. 1, No. 1, March 2012, Copyright © World Science Publisher, United States.  
www.worldsciencepublisher.org.
- [7]**Hongwei Li, YuanshunDai, Ling TianandHaomiao Yang**, “Identity-Based Authentication for Cloud Computing”, Springer-Verlag Berlin Heidelberg 2009.
- [8]**Sanchika Gupta, Anjali Sardana, Padam Kumar**, “A light Weight Centralized File Monitoring Approach for Securing Files in Cloud Environment ”,The 7th International Conference for Internet Technology and Secured Transactions (ICITST-2012) ©IEEE 2012.
- [9]**Sanchika Gupta, Anjali Sardana, Padam Kumar,Ajith Abraham**, “A secure and light weight approach for critical data security in cloud”, 2012 Fourth International Conference on Computational Aspects of Social Networks (CA Sons).
- [10] **Forouzan**, “Cryptography and Network Security”, TMH 2012.
- [11]**A.H. Steven, F. Stephanie and S.Anil**, "Intrusion detection using sequences of system calls," J Comput.Secure.Vol. 6, no. 3, 1998, pp. 151-180.
- [12]**G.P.Adam,D.S.John,G.JohnLinwood,A.N.S. Craig, R.G. Garth and R.G.Gregory**,"Storage-based intrusion detection: watching storage activity for suspicious behavior," Book Storage-based intrusion detection: watching storage activity for suspicious



behavior, Series Storage-based intrusion detection: watching storage activity for suspicious behavior, ed., Editor ed/eds., USENIX Association, 2003.

[13] **S. Patil, A. Kashyap, G. Sivathanu and E.Zadok**, "13FS: An in-kernel integrity checker and intrusion detection file system."

[14] **Q. Nguyen Anh and T. Yoshiyasu**, "A novel approach for a file-system integrity monitor tool of Xen virtual machine," Book A novel approach for a file-system integrity monitor tool of Xen virtual machine, Series A novel approach for a file-system integrity monitor tool of Xen virtual machine, ed., Editor ed./eds., ACM, 2007.

[15] **R.K.L. Ko, P. Jagadpramana and L. Bu Sung**, "Flogger: A File-Centric Logger for Monitoring File Access and Transfers within Cloud Computing Environments," Trust, Security and Privacy in Computing and Communications (TrustCom), 2011 IEEE 10th International Conference on, pp. 765-771.

[16] **H.K. Gene and H.S. Eugene**, "The design and implementation of tripwire: a file system integrity checker," Book The design and implementation of tripwire: a file system integrity checker, Series The design and implementation of tripwire: a file system integrity checker, ed., Editors, ACM, 1994.

[17] **Meiko Jensen, Jorg Schwenk, Nils Gruschka, Luigi Lo Iacono**, "On technical security issues in cloud computing" 2009, IEEE Computer Society.

[18] **Henry Kasim, Terence Hung, Xiaorong Li**, "Data Value Chain as a Service Framework: for Enabling Data Handling, Data Security and Data Analysis in the Cloud", 2012 IEEE 18th International Conference on Parallel and Distributed Systems.

[19] **Tina Francis, S. Vadivel**, "Cloud Computing Security: Concerns, Strategies and Best Practices: Proceedings of 2012 International of Cloud Computing, Technologies, Applications & Management", 2012 IEEE.

[20] **Mr. Prashant Rewagad, Ms. Yogita Pawar**, "Use of Digital Signature with Diffie Hellman Key Exchange and AES Encryption Algorithm to Enhance Data Security in Cloud Computing", 2013 International Conference on Communication Systems and Network Technologies.

[21] **Mrs. G. Nalinipriya ME, (PhD), Mr. R. Aswin Kumar**, "Extensive Medical Data Storage With Prominent Symmetric Algorithms On Cloud", A Protected Framework: 2013 International

Conference on Smart Structures & Systems (JCSSS-20 13), March 28 - 29,2013, Chennai, INDIA.

[22]**Wang Jun-jie,MuSen**, “Security Issues and Countermeasures in Cloud Computing”,2011 IEEE.

[23]**Balachandra Reddy Kandukuri, Ramakrishna Paturi V,Dr. Atanu Rakshits**, “Cloud SecurityIssues”, 2009 IEEE International Conference on Services Computing,2009 IEEE.

[24][http://en.wikipedia.org/wiki/Cloud\\_computing\\_security](http://en.wikipedia.org/wiki/Cloud_computing_security).

[25]<http://www.igi-global.com/article/cloud-computing-security/52037>.