

**DESIGN AND ANALYSIS OF EFFICIENT
ALGORITHMS FOR CLOUD DATA STORAGE AND
SECURITY MANAGEMENT IN CLOUD COMPUTING**

By

PRAKASH GL

SAP ID: 500014745

*A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE
REQUIREMENTS OF THE DEGREE OF*

DOCTOR OF PHILOSOPHY

In

COMPUTER SCIENCE AND ENGINEERING

To



**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES**

DEHRADUN - 248007

MAY, 2018

Under the Supervision of

DR. MANISH PRATEEK

Professor and Director

SoCSE, UPES, Dehradun

DR. INDER SINGH

Assistant Professor(SG)

SoCSE, UPES, Dehradun

I dedicate my Ph.D. Thesis to

My loving Parents, In-Laws, and my Guide Professor

Dr. Manish Prateek, and

Dr. Inder Singh

for their endless support, blessings and guidance.

CERTIFICATE OF CORRECTION

This is to certify that the thesis entitled "**Design and Analysis of Efficient Algorithms for Cloud Data Storage and Security Management in Cloud Computing**" by **Prakash G L, (SAP ID: 500014745)** is being submitted by in fulfillment for the Award of DOCTOR OF PHILOSOPHY in School of Computer Science to the University of Petroleum and Energy Studies. Thesis has been corrected as per the evaluation reports dated 05/12/2018 and all the necessary changes / modifications have been inserted/incorporated in the thesis.

M.P.
15/2/19

Dr. Manish Prateek

Internal Guide

Professor and Director

School of Computer Science,

UPES, Dehradun

I.S.

Dr. Inder Singh

Co-Guide

Assistant Professor(SG)

School of Computer Science,

UPES, Dehradun

M.P.

Dr. Manish Prateek

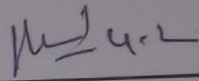
Professor and Dean/Director

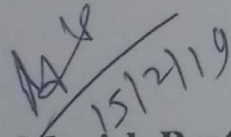
School of Computer Science

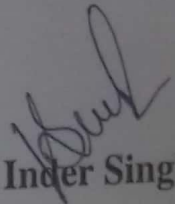
Date: *15/2/19*

THESIS COMPLETION CERTIFICATE

This is to certify that the thesis entitled "Design and Analysis of Efficient Algorithms for Cloud Data Storage and Security Management in Cloud Computing" by Prakash G L, (SAP ID: 500014745) in partial completion of the requirements for the award of the Degree of Doctor of Philosophy in School of Computer Science is an original work carried out by him under our joint supervision and guidance. It is certified that the work has not been submitted anywhere else for the award of any other diploma or degree of this or any other University.

Signature of the Candidate: 
SAP ID: 500014745
Date: 22/01/2019


Dr. Manish Prateek
Internal Guide
Professor and Director
School of Computer Science,
UPES, Dehradun


Dr. Inder Singh
Co-Guide
Assistant Professor(SG)
School of Computer Science,
UPES, Dehradun

**SCHOOL OF COMPUTER SCIENCE AND ENGINEERING
UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
DEHRADUN - 248007**



DECLARATION

I declare that this thesis, which I submit to University of Petroleum and Energy Studies, Dehradun, for examination in consideration of the award of a higher degree Doctor of Philosophy in Computer Science and Engineering is my own personal effort. Where any of the content presented is the result of input or data from a related collaborative research programme this is duly acknowledged in the text such that it is possible to ascertain how much of the work is my own. Furthermore, I took reasonable care to ensure that the work is original, and, to the best of my knowledge, does not breach copyright law, and has not been taken from other sources except where such work has been cited and acknowledged within the text.

Signature of the Candidate:_____

SAP ID:500014745

Date:_____

THESIS COMPLETION CERTIFICATE

This is to certify that the thesis entitled "**Design and Analysis of Efficient Algorithms for Cloud Data Storage and Security Management in Cloud Computing**" by **Prakash G L, (SAP ID: 500014745)** in Partial completion of the requirements for the award of the Degree of Doctor of Philosophy in Computer Science and Engineering is an original work carried out by him under our joint supervision and guidance. It is certified that the work has not been submitted anywhere else for the award of any other diploma or degree of this or any other University.

Signature of the Candidate:_____

SAP ID:500014745

Date:_____

Dr. Manish Prateek

Internal Guide

Professor and Director

SoCSE, UPES, Dehradun

Dr. Inder Singh

Co-Guide

Assistant Professor(SG)

SoCSE, UPES, Dehradun

Acknowledgment

I bow my head humbly to pay heart felt regards to Almighty God for giving me the strengths and blessing in completing this thesis.

There are quite a few people that have helped me in one way or another to the completion of this work. It is with great pleasure, I would like to thank all of you from very deep inside.

Foremost, I would like to express my sincere gratitude to my thesis advisors Prof. Manish Prateek and Dr. Inder Singh, for picking me up as a student at the critical stage of my career and the continuous support of my Ph.D study and research, for his patience, motivation, enthusiasm, and immense knowledge. His guidance helped me in all the time of research and writing of this thesis.

It is absolutely difficult to succeed in the process of finding and developing an idea without the help of a specialist in the domain. I found in my advisors not only the source of wonderful ideas to develop, but also the support that a PhD student needs. I could not have imagined having a better advisor and mentor for my Ph.D study.

Besides my advisors, I would like to thank Chancellor Dr. S. J. Chopra, Vice chancellor Dr. Deependra Kumar Jha and Dean Dr. Kamal Bansal at the University of Petroleum and Energy Studies for their encouragement, suggestions and valuable support for my research work.

I would like to express my special thanks to Dr. J K Pandey, R&D, Director and Dr. Rakhi Ruhel, Program Manager-Ph.D, University of Petroleum and Energy Studies, for his assistance during my research work. I am grateful to the University of Petroleum and Energy Studies, for giving me an opportunity to pursue my research and for providing all facilities in the Department of Studies in School of Computer Science and Engineering.

I would like to thank all the Heads of School of Computer Science and Engineering Departments, doctoral students for their feedback, cooperation, and of course friendship. In addition I would like to express my gratitude to all colleagues in the university.

I would like to express my gratitude towards member of IBM, Bangalore, for their kind co-operation and encouragement which help me in completion of this project.

I would like to thank my teachers, friends and well wishers Dr. Venugopal K R, Dr. Manjula S H, Dr. Venkatagiri Jayaramareddy Byra Reddy, Dr. J Devaraju, Dr. Shivaramu, Dr. Katlakanti Mohan Reddy Mr. V Kesavan, Mr. Ramesh M, and Mr. A N Shankar for their insightful comments and encouragement.

I cannot begin to express my gratitude to my family for all of the love, support, encouragement, and prayers they have sent my way along this journey. I am eternally indebted to my loving parents and in-laws for all the sacrifices they have made on my behalf. I would like to express sincere gratitude to my beloved wife Savitha Heggede who believed in me and provided encouragement during challenging times. Your unconditional love and support in the moments when there was no one to answer my queries has helped me immensely. To Sachin and Chethan, thank you for helping me when I needed the most. To all my friends thank you for your support and constant encouragement. To my daughter Shreya Smaran, she is the inspiration for me to complete this journey and also for the sacrifices made along the way.

Abstract

Cloud computing is one of the important business models in the modern Information Technology. It provides various services (hardware, software) to the users with minimal interaction and low cost. Storage service is one of the most useful services in cloud computing, which move data owners data from local computing system to the cloud. In this paradigm, once the data moves from the local computing system to the cloud, the data owner lost the physical control of the outsourced data on the cloud. So that, storage service creates data security challenges. For space and computation benefits, cloud service provider can remove the rarely accessed data from the cloud storage server. Therefore, the integrity of the outsourced data has to be verified frequently using public or private verification method.

In this thesis, focus on two data security concern such as data confidentiality and remote data integrity on cloud storage system. On one hand, we focus on data confidentiality for cloud applications and services on an untrusted cloud server. For this purpose, we proposed a lightweight block levels symmetric key data encryption and decryption algorithm with key rotation technique. To protect unauthorized access to the data sharing an honest but curious server and a malicious user adversary threats is considered in our proposed system.

On the other hand, we focus on the remote data integrity concern. In cloud storage, the data owners can store their data, applications, and services on a remote server without the burden of local infrastructure and maintenance. So that, data owners no longer physical control of outsourced data. Therefore, the data protection at rest, in transit and process is a challenging task in cloud storage system. For Secured cloud data audit, we have introduced a Third Party Auditor(TPA), which reduces the burden of the data owner. Moreover, this process of auditing will not bring any new security risk on data privacy. In order to ensure the data integrity and reduce the data owners computational resources, in this work we have proposed a remote data integrity auditing methods such as identity-based and linear authentication protocols and Elliptical Curve Digital Signature(ECDS) methods.

In the identity and linear authentication method, which utilize the bilinear operation and decisional Diffie-Hellman algorithm to verify the integrity of outsourced data without retrieving the original file. Its concerns about the proof of data possession issue. We have considered a security level, public verification and performance aspects for remote data

verification without keeping the data locally. In this method, we introduce the block level data verification using third-party auditor to fulfill the security requirement of the data owners. To analyze the performance of the system, first, we define the single data owner on multiple servers and then multiple data owners on a single server for public data verification.

In order to reduce the storage overhead and computation overheads with same security level as RSA public cryptography, we propose a novel method of data verification technique using elliptic curve digital signature method. Besides, these methods not only verify the integrity of data, but also detect the invalid data block during the verification process.

Contents

Declaration	ii
Thesis Completion Certificate	iii
Acknowledgment	iv
Abstract	vi
List of Figures	xi
List of Tables	xii
1 INTRODUCTION	1
1.1 Introduction	1
1.2 Cloud System Architecture	2
1.2.1 Research challenges	4
1.3 Cloud Data Security Challenges	7
1.3.1 Security issues	8
1.3.2 Applications	11
1.4 Solutions to Data Security Challenges	12
1.4.1 Data Confidentiality	12
1.4.2 Data Integrity Auditing	13
1.5 Organization of the Report	16
1.6 Summary	16
2 LITERATURE SURVEY	17
2.1 Introduction	17
2.2 Data Confidentiality	17
2.3 Remote Data Verification using Protocol	24
2.4 Data audit using Digital Signature	33
2.5 Summary	37

3	BACKGROUND	38
3.1	Bilinear Pairing	38
3.2	Symmetric Key Cryptography	38
3.3	Message Authentication Code(MAC)	40
3.3.1	Restricted MAC method	41
3.4	Homomorphic Linear Authenticator method	41
3.4.1	Initial File Setup Phase	42
3.4.2	Data audit phase	44
3.5	ECDSA Digital Signature	45
3.5.1	Elliptic Curves	46
3.5.2	Mathematical Background	47
3.5.3	ECDSA-Algorithm	49
3.6	Summary	51
4	STATEMENT OF THE PROBLEM	52
4.1	System Model	53
4.2	Security model	55
4.3	Objectives	55
4.4	Contribution	56
4.5	Summary	57
5	PROPOSED DATA AUDITING SCHEMES	58
5.1	Introduction	58
5.2	Data Encryption using Key Rotation	59
5.2.1	System Model and Setup	61
5.2.2	Definitions	63
5.2.3	Algorithms	66
5.3	Data Audit using Protocol	68
5.3.1	System model	69
5.3.2	Basic Auditing Scheme	70
5.3.3	Identity-based Data Auditing Method	73
5.3.4	Linear Authenticator based Data Auditing	79
5.4	Data Audit using Digital Signatures	86
5.4.1	Initial File Setup	86

5.4.2	Data verification	90
5.4.3	Batch auditing	92
5.5	Summary	93
6	SIMULATION RESULTS AND ANALYSIS	94
6.1	Simulation setup	94
6.2	Data Encryption and Decryption	95
6.3	Remote Data Audit using Protocol(RDAP)	97
6.3.1	Communication Cost:	97
6.3.2	Computation Cost:	98
6.4	RDADS Simulation Results	101
6.4.1	Communication Cost	101
6.4.2	Computation Cost	101
6.5	Summary	108
7	CONCLUSIONS	109

List of Figures

1.1	Cloud Data Storage System Model	3
1.2	Cloud Data Security Challenges	8
3.1	Block diagram of a symmetric key cryptography	39
3.2	MAC block diagram	40
3.3	Basic Block Diagram of Data Auditing using HLA's	43
3.4	Digital signature block diagram	45
3.5	Elliptic curve: $(y^2 = x^3 + x + 1 \text{ mod } 23)$	49
3.6	Elliptic curve point addition on Finite field F_{23}	50
4.1	Proposed cloud System Model	53
5.1	Block Diagram of Data Encryption and Decryption in Cloud System	60
5.2	Cloud Data Storage System Model	61
5.3	Data Auditing System Model	71
5.4	Basic Data Auditing System	73
5.5	Batch Auditing setup and Private Verification	78
5.6	Data Auditing using Public Verification	78
5.7	Batch Auditing Model	80
5.8	System model	87
5.9	Metadata Representation	90
6.1	CA Shifter and CA inverter comparison	96
6.2	Key Motor Encryption and Decryption Time comparison	96
6.3	Signature generation cost for different blocks	102
6.4	Initial File setup and Upload time(Sec)	103
6.5	Data block verification cost	104
6.6	Incorrect verification blocks time	105
6.7	CSP and TPA 256KB Data Blocks Audit Time(Sec)	106
6.8	CSP and TPA 50KB Data Blocks Audit Time(Sec)	107
6.9	RDAP vs RDADS Signature generation time for 256KB blocks	107
6.10	RDAP vs RDADS 256KB data blocks verification time	108

List of Tables

1.1	Different types of security threats	9
1.2	Existing solution for cloud data confidentiality	13
1.3	Data Auditing Methods	15
2.1	Data Auditing Methods	34
2.2	Data Auditing Methods	35
3.1	Domain parameters	47
5.1	Structure of Block Status Table	62
5.2	Notations	63
5.3	Notations	72
6.1	Simulation setup	94
6.2	File text and Cipher text	95
6.3	Tag generation time for different file sizes	98
6.4	Tag generation time for different data block sizes	99
6.5	Data verification cost with a different block size	99
6.6	Data verification cost with different batch size	100

Chapter -1

INTRODUCTION

1.1 Introduction

Cloud computing is a new computing paradigm in Information Technology. The cloud computing is defined by National Institute of Standards and Technology (NIST) as enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications and services) which can be rapidly provisioned and released with minimal management effort or service provider interaction [1]. In this model, both hardware and software resources are delivered over the internet with the interaction between users and cloud service providers. There are three service models namely software as a service (SaaS), platform as a service (PaaS), and infrastructure as a service (IaaS) which are implemented over private, public, hybrid or community deployment model. Despite several advantages, cloud computing has various issues and challenges, which need to be solved. Data privacy and security is one of the significant research areas in cloud computing.

Cloud computing is one of the on-demand high-quality service delivery models with a lower cost. Amazon Web Service (AWS) is the most commonly used cloud service provider. They provide various services such as; Amazon Elastic Compute Cloud (EC2) an IaaS service, Amazon Elastic Beanstalk a PaaS service for hosting applications, Amazon Elastic Block Storage (EBS) and Amazon Simple Storage Ser-

vice (S3) for storage, AWS Identity and Access Management (IAM) service for secure control access to AWS.

1.2 Cloud System Architecture

The cloud data storage system model for secure data access sequences is explained in Figure 1.1. The block diagram of cloud computing system architecture contains four functional blocks for data storage such as a data owner, Cloud Service Provider (CSP), authorized users, and Trusted Third Party [2]. They are used for accessing data from data centers in public cloud servers.

The functions of these functional blocks are as follows;

*Data owner:*The data owner can be any organization for generating outsourcing data to store in the data center of public cloud model for the external use on the demand of the authorized users based on pay per usage.

Cloud Service Provider: Manage the cloud servers and data centers in the public cloud and provide the storage infrastructure to the data owner for storage of outsourced data in the data center on the payment based on the requested storage capacity. It coordinates the trusted third party to verify the authorized users and to retrieve the data from the cloud server to make them available for the authorized user on demand. Depending on the type of cloud used, the cloud service providers responsibilities could include providing infrastructure, physical security of the premises, operating system and network security. Sharing of cloud resources such as providing infrastructure, operating system, application and network security is controlled by the cloud service provider depending on the cloud deployment model.

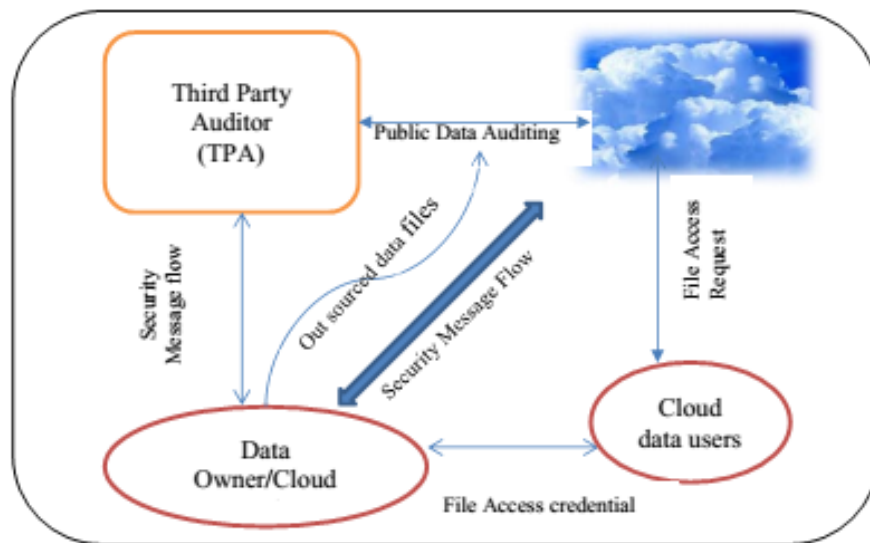


Figure 1.1: Cloud Data Storage System Model

On the other hand, the actively processing cloud data is controlled by cloud users depending on the cloud service model used in their application. An organization classifies the information according to the sensitivity to its loss or disclosure. The level of information sensitivity classification is defined by the data owner based on the security control. Storage as a service is provided by the cloud service provider such as Google Drive, iCloud, Dropbox, etc., to data users with lower cost than traditional storage service. The cloud service providers not only store the owner's data in the cloud but also share it with authorized users. Once the data leaves the data owner premises, there is no control of outsourced data to the data owner.

Users: the set of authorized users to access the remote data stored in cloud server through trusted third party and cloud service provider. All the users are the clients of the data owner.

Trusted Third Party: An entity which is trusted by all other entities of the system such as CSP, data owner, and users. In many distributed computing systems the data owners rely on TTP for performing his operation securely and efficiently. The function

of TTP is to send the audit challenge message to CSP and receives the corresponding response message as a proof of data verification. The role of TTP is important in the public data verification scheme for data security and privacy in the cloud storage system. If the TTP acts dishonestly with other entities of the system, then there is no data security guarantee for the cloud data storage system.

1.2.1 Research challenges

The cloud computing is one of the important research areas in the field of information technology. Some of the research challenges in the field of cloud computing are listed below. These research challenges mainly focus on cloud deployment and service models.

- Service Level Agreement(SLA)
- Cloud Data Management
- Security and privacy
- Interoperability
- Energy Resource Management
- Multi-tenancy
- Reliability and Availability of service
- Data recovery and backup
- Server consolidation

Service Level Agreement(SLA): In cloud computing cloud resources are administered by service level agreements. The cloud service providers SLAs evaluation is a challenging task. The cloud service providers create SLAs for cloud users to access various cloud resources. The SLAs are defined while considering various parameters such as; data protection, price structure, outage, etc. Before signing a contract with CSP for a specific service, cloud user must evaluate the various SLAs parameters. The specification of SLAs will reflect the user needs.

*Cloud Data Management:*In the past few years the cloud data has grown at an alarming pace. This data can be organized in several forms such as structured, unstructured, semi-structured, static and dynamic data. Management of the data is an important research area in cloud computing. The CSPs do not have any access to the physical security of the data, but they are dependent on the infrastructure provider for data security. The cloud infrastructure provider must provide the security for data confidentiality and availability. The confidentiality of the data is achieved by using cryptographic techniques, whereas the availability of the data is achieved by using secured remote data auditing methods.

Security and privacy: The security and privacy is more important in the development of cloud computing to protect the resources from the unauthorized users. In cloud computing, data security and privacy becomes the important issue, because the data is located at remote places across the globe.

Interoperability: An ability to work with multiple systems while exchanging information between the systems. In public cloud deployment model, many networks are not designed to interact with each other. Due to lack of integrations, it is very

difficult to combine these networks in the cloud. To overcome this challenge a standard inter-operable platforms and data portability must be developed, which helps CSPs.

Energy Resource Management: Energy saving in the data center has an important economic incentive for data center operations and also a significant contribution to the environment. In cloud data centers the cost of cooling and power requirement is always higher than the operational expenditure. Designing an energy efficient data center is one of the research areas in cloud computing.

Multi-tenancy: It is one of the important technologies in cloud computing to allow one instance of the application to serve many customers by sharing resources over the Internet. To access the shared resources such as servers, database, etc., is essential as it affects the response time and the user performance. Proper security and isolation of resources in multi-tenancy is a challenging task in cloud computing infrastructure layer.

Reliability and Availability of service: *Reliability* is the ability of a hardware or software resources to consistently perform according to its specifications. The challenge of reliability comes in cloud computing when resources are delivered as on-demand service. To deliver resources under any network conditions, reliability is a challenging task in cloud computing.

Availability is the ratio of time a resource is functional to the total time required to function. It can be expressed in terms of average or total uptime/downtime of a resource (hardware/software) for a given period. In a traditional computing system, the availability of these has been limited to users deployed and maintained, whereas in the cloud these resources are shifted to data-center. It is very important for the cloud

service providers to offer the available resources to the users.

Data Recovery and Backup: Data security is significant in cloud computing as the data resides at remote storage servers. The cyber-attacks are increasing on cloud data so that the responsibility of decision-makers is important to strengthen its data security and avoid these threats. The data backup and recovery solutions are important in cloud computing. Designing a secure data recovery and backup approach by saving crucial data is one of the research areas in cloud computing.

Server consolidation:: Server consolidation is a technique to maximize resource utilization of computer server resources in order to reduce the capital expenditure and minimize energy consumption in cloud computing. Server consolidating provides plenty of benefits using server virtualization in cloud computing environment and which is directly impacts the performance of the system.

1.3 Cloud Data Security Challenges

The benefits of cloud computing include managing and utilizing cloud services like lower fixed costs, higher flexibility, automatic software updates, increased collaboration, and the freedom to work from anywhere.

According to the Cloud Security Spotlight Report, more than 90 percent of organizations are concerned about public cloud security because the cloud has its data security issues. The importance of the security in a cloud system is shown in Figure 1.2. Data security and privacy are one of the most important issues in cloud security challenges.

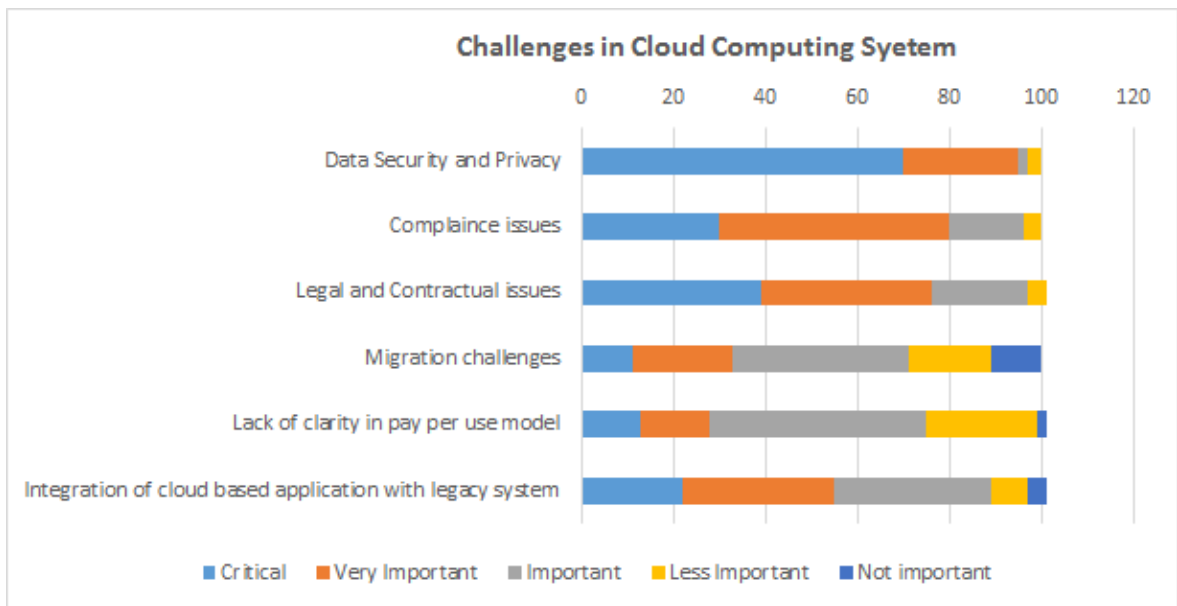


Figure 1.2: Cloud Data Security Challenges

1.3.1 Security issues

In a cloud deployment model, the user loses control over the physical security of the resources. In a public cloud, computing resources are shared with other organizations or users. In a shared pool outside the premises, we don't have any knowledge or control of where the The resources are being used. Outsourcing our data in a shared environment in the cloud storage may put our data at risk.

The various threats and its suggested prevention techniques [3] on network, host and application layer in cloud computing paradigm is summarized in Table 1.1.

Based on the survey conducted by Cloud Security Alliance (CSA) the following security issues are identified within cloud computing.

Data encryption: It is a process of converting data from plaintext to ciphertext before it is sent to cloud storage. In cloud computing, cloud service providers offer this services to provide security of outsourced data. For example, Office 365 Message

Table 1.1: Different types of security threats

Security attack	Possible techniques for prevention
Spoofting attack	Authentication, Protect secrets, Don't store sensitive information as plain text
Tampering with data Authorization	Data hashing and signing, Message authentication codes, Digital signatures
Refuse to accept	Digital signatures, Time-stamps, Audit trails
Information disclosure	strong authorization, secured encryption, don't not store secrets (for example, passwords) in plain-text.
Denial of service	Resource and bandwidth throttling techniques, Quality of services, Authorization and authentication.
Elevation of privilege	use least privileged service accounts to run processes and access resources.
Session Hijacking(man in the middle attack)	encrypted session, encrypted communication channels.
Unauthorized Access	Configure secure Web permissions, Assign permission to files and folders.
Auditing and Logging	Audit and log activity on the Web server and database server, and on the application server, Do not use shared accounts
Poor key generation or key management	Use built-in encryption routines, store the key in a restricted location, Use strong random key generation functions, Expire keys regularly
Weak Encryption	Use the proven cryptographic services, use standard algorithms

Encryption built in service encrypt all the messages in the cloud platform. Designing a secured remote data storage method is one of the important research areas in cloud computing.

Data breach: It is an unauthorized way of accessing sensitive, confidential or protected data in a network. Data breaches may involve the hacker accessing the confidential personal information such as; personal health information (PHI), personally

identifiable information (PII), or intellectual property, etc.

Access control: It is an Identity and authentication security technique that can be used to ensure the authorized users to access the resources in a computing environment. Access control can be a physical or logical access control . The physical access control limits access to the various physical entity such as physical IT resources, whereas the logical access control limits to access the stored data and files.

Insecure Interfaces and APIs: Insecure Interfaces and Application Programming Interfaces(APIs): In cloud computing, cloud service providers expose a set of user interfaces and APIs to cloud users to use cloud services. The availability and security of the cloud services are dependent on the APIs. The user interfaces and APIs are the most exposed part of the network. These assets will be the source of the heavy attack in the cloud.

System vulnerabilities: It is exploitable bugs in programs which helps the attacker to infiltrate a system to steal data and take complete control of the system. The system vulnerabilities in the operating system put the security of services and data at risk. It creates new attack in a distributed system while access to shared memory and resources in the cloud.

Cloud account hijacking: It is a process of hijacking cloud users account through malicious or unauthorized activity. With stolen information, attackers can access cloud computing services and compromise the confidentiality, integrity, and availability of those services.

Data Loss: It is a result of any process or event that, data is corrupted or made

unavailable to user or application . Data loss can occur on data both at rest and transit due to the various reasons such as; Data corruption, deleted or/and stolen by an attacker or any network intervention attack, physical damage of storage devices and Virus infection. Data loss can be prevented by implementing data backup solutions and security mechanisms on data storage assets.

*Denial of service attacks (DoS):*It is an attack which originates by requesting resources from the cloud server that server cannot respond to requests. It prevents users to access their data or applications. By forcing the targeted cloud service to consume an excessive amount of system resources such as; processor power, memory, disk space, or network bandwidth and attackers can cause a system slowdown.

1.3.2 Applications

Cloud services are provided to users at different levels such as; applications, platform or infrastructure and more number of cloud services are currently available to the users. Using these services developers can develop a wide variety of applications and delivered to the end users. Some of the examples of cloud applications [[4, 5, 6] are as follows;

Software-as-a-Service (SaaS): It is a software distribution model, which deliver the applications to the end users over the Internet. Examples of SaaS applications are; Salesforce.com, DropBox, Microsoft Office 365, Google Apps, Amazon Web Services etc.

Platform-as-a-service (PaaS): It delivers a hardware or software platform for developers to carry out the specific task. Examples of PaaS applications are; Google

App Engine, AWS Elastic Beanstalk, Windows Azure, etc.

Infrastructure-as-a-service (IaaS): It provides maximum controlled virtualized computing resources over the internet to the end users. Examples for IaaS are; Amazon EC2, Google Compute Engine (GCE), and IBM Cloudburst, etc.,.

1.4 Solutions to Data Security Challenges

To provide the security to the outsourced data is a challenging task in cloud computing. The most important parameters for data security is confidentiality, integrity, and availability of cloud data storage service. Before utilizing the cloud data by authorized user, data integrity should be verified using any verification method.

1.4.1 Data Confidentiality

Protecting data in the cloud, authentication and integrity[7], access control, encryption [8, 9, 10, 11], integrity checking and data masking [12] are some of the data protection techniques.

Cryptography is one of the efficient methods for data security in cloud computing which includes the design and implementation of an efficient encryption and decryption algorithms. In symmetric cryptography, before outsourcing data to a cloud server, the data is encrypted into cipher text using a secret key and later user decrypts using the same shared secret key.

In cloud computing, data owners increasingly outsource their sensitive data in encrypted form to public cloud for more flexibility and economic savings [13, 14]. To protect data in transit to and from the cloud as well as data stored in the cloud, efficient

data encryption and decryption algorithms are used for security.

There are several existing cloud data confidentiality techniques are presented in the Table 1.2

Table 1.2: Existing solution for cloud data confidentiality

Researcher	Function	Technique	Service types
Jin Li et al., [15]	Ramp secret sharing scheme on distributed system	File and Block level de-duplication	data confidentiality without encryption
Cong Wang et al., [13]	Ranked keyword search	one to many ordered preserving symmetric encryption	secure data access
Lan zhou et al., [16]	Cryptographic roll based access control	It uses trust and roll model for data owners and users	secure data access
Rongmao chen et al., [17]	Dual server public key encryption with keyword search	It uses linear and homomorphic smooth projection hash function	data security
Jie xu et al., [18]	Verifiable delegation circuit cipher text policy attribute based hybrid encryption	It uses boolean circuit with single output	access control

1.4.2 Data Integrity Auditing

The traditional method of data verification methods proposed in RSA [19], MD5 [20] is not efficient in cloud computing model, because the amount of data stored in the cloud is of large scale which consumes more computational resources for data

verification. To verify correctness of the outsourced data in cloud computing, recently there are many public auditing mechanisms proposed in [21, 22, 23, 24] and [25] without downloading the entire data from the cloud server.

For integrity and privacy of the outsourced data on the cloud, several existing remote data auditing protocols are listed in Table 1.3. Table 1.3 compares the existing data auditing methods with services, functions, and techniques. In all these methods follows the stateless verification, unbounded use of queries, public verification dynamic auditing and batch auditing requirements in the design process. The design process consists of an initial file setup phase and data verification phase. All these existing data auditing methods give more attention on the data verification phase, rather than the initial setup phase so far.

From Table 1.3, we can also observe that many of the existing solutions cannot support the privacy of the data during the audit phase. To reduce the computational overhead of initial file setup phase and to reduce the burden of data owners, we introduce the sample data block auditing with finite group operations.

Although Cloud computing provides various services with minimal overheads, which have many challenges on the performance of the system. Security of the data at rest is one of the big challenges in cloud computing. The security of the cloud data can provide in three ways; such as confidentiality, integrity, and availability . Data integrity is one of the most important parameters for data security in cloud storage. The integrity of the cloud data cannot be guaranteed due to the following reasons;

- Due to losses in the control of the outsourced data the traditional data integrity checking methods cannot be applied directly.

Table 1.3: Data Auditing Methods

Researcher	Function	Technique	Service
A Juels et al. [26]	Data Proof of Retrievability	Pre-computed file hash values have stored on cloud server	Data auditing
C Gentry et al.[27]	Encrypted data blocks	Calculated metadata will be stored locally and cloud server	Block level data auditing
C. Wang et al. [28]	Block Authentication with random masking	Local and Public auditing using TPA	Block level data auditing
G. Ateniese et al. [29]	BLS signature and Merkle Hash Tree	Dynamic data updates	data verification
B. Wang et al. [30]	Group operation using Bilinear map	Batch auditing in multi-cloud	Privacy preserving
Boyang Wang et al. [31]	Homomorphic authentication using proxy	Block level outsourced data auditing	Data integrity checking
L.Chen et al. [32]	Uses Message Authentication Code	Auditing of shared data	privacy preserving

- Due to hardware failure and software failure, the cloud service provider may hide the data corruption for his storage service reputation point of view.
- To save the data maintenance cost, the cloud service provider can delete the rarely accessed data from the storage or can change to off-line method of storage.

1.5 Organization of the Report

The thesis is organized into the following chapters. The second chapter presents a brief review of literature for cloud data security parameters confidentiality and the integrity of the cloud data using protocol and digital signatures techniques. The background for designing data encryption and data auditing techniques is explained in the chapter three.

The system model and objective of the research work is explained in the chapter four. The chapter five discuss the proposed data encryption algorithm using using key rotation technique, remote data audit using protocol and public key digital signature methods. In chapter six, we presented the simulation results of the proposed data auditing methods. Finally, a brief summary, contributions is presented in the chapter seven.

1.6 Summary

In this chapter, we explain a basic architecture of data storage system model in cloud computing paradigm. Various research area and data security issues and challenges are listed on cloud system. Some of the traditional data privacy and security techniques, contribution of the proposed work and organization of the report is presented. Next chapter, present a various data security issues of the existing solutions proposed by the researcher.

Chapter -2

LITERATURE SURVEY

2.1 Introduction

Literature survey is the most important step in software development process. Before developing the tool, it is necessary to determine the time factor, economy and company strength.

In the recent years, many data security issues are discussed by researcher in the field of cloud data and applications storage services [33, 34, 35]. The goal of this chapter it to identify the significance of the data security in the field of cloud computing, then identify specific methods and techniques for the proposed research work in the field of data security.

This chapter discusses the recent cloud data security and privacy issues in cloud storage service such as data confidentiality, secure data auditing using protocol and data integrity verification using digital signature schemes are proposed by the various researches till date and the research gap for proposed research work.

2.2 Data Confidentiality

In cloud computing now a day's adoption and diffusion of data sharing is one of the most demands in the distributed storage service. In this storage service, data security is the challenging issue in cloud storage system. Recently there are many solutions are proposed to address the cloud data security and the research gaps are discussed in

the following section.

To ensure the data is only accessible by the authorized users and for end-to-end secure data transfer requires the efficient encryption and or decryption algorithm to the whole dataset before performing any further operations. Further, it is very difficult to see the upcoming applications for the future without the proper use of algorithms. Also, several data security softwares need to be developed for the privacy of the organization's data.

There are issues related to the confidentiality, correctness, query and access permission of outsourced data because cloud is managed by un-trusted third party [33], [36]. Owners are always suspected about security of the data. So we consider this as basic idea for proposed concept. The collusion attack and distributed DOS attacks are serious issues due to the malicious user in the system and DDOS Attack occurs when multiple big data systems flood the resources of the targeted big data systems. The secure data transfer can be introduce the identity based secure encryption and re-encryption.

Jing-Jang Hwang et al. [37], has proposed a business model for cloud computing for data security using data encryption and decryption algorithms. In this method, cloud service provider is responsible for data storage and data encryption/decryption tasks, which takes more computational overhead for process of data in cloud server. The main disadvantage of this method is, there is no control of data for data owner i.e, data owner has completely trusted with cloud service provider and he has more computational overhead.

Junzuo et al. [38]], proposed an Attribute Based Encryption (ABE) and verifiable data decryption method to provide data security in cloud based system. They have designed the data decryption algorithm based on the user requested attributes of the out sourced encrypted data. One of the main efficiency drawbacks of this method is, cloud service provider has more computational and storage overhead for verification of user attributes with the outsourced encrypted data. While introducing third party auditor we can reduce the storage, computation, and communication overheads of the cloud server, which improves the efficiency of the cloud data storage.

Chun-I Fan et al., in [14] proposed an Attribute-based encryption (ABE) to protect the privacy of the outsourced data on cloud using set of data attributes. However, it supports privacy of the data on the cloud, it takes more storage and communication overheads. Since the data, updating may occur frequently in cloud computing, data management will become an important issue in cloud.

In [39] Baodong Qin et al., proposed a verifiable and non-verifiable Attribute-based encryption (ABE) technique for cloud data privacy and security using public key cryptography. In cloud, the volume of data size grows larger and larger, so that public key cryptographic operations are not efficient for large data management in cloud and also the cloud service provider can derive the outsourced data from the attributes set. So that, we can apply symmetric key cryptographic operation to improve the data security.

Junbeom Hur in [40] proposed a cryptographic based Cipher-text policy attribute-based encryption (CP-ABE) solution for data security on distributed cloud storage system. In this solution, the data owner defines the set of policies over the data before

send to the cloud. This method has the following drawbacks, such as; i). the key generator is an untrusted entity so that, he can generate the private key from the public key to access the outsourced user data. ii). It follows the public key crypto-systems so that it is not suitable for large data. iii). It depends on the attributes set so that cloud server can derive the data from the predefined attribute sets of the outsourced data.

In [41] Cipher-text Policy Attribute Based Encryption (CP-ABE) scheme was proposed to encrypt data using encryption policy, which is specified in the access structure and it is associated with cipher-text In this scheme a private key is generated by data owner using set of his attributes. A user can decrypt the data only if the attributes in the private key satisfy the encryption policy specified in the cipher-text

Data access control and confidential in cloud storage system Jie Xu et al., [42] present a circuit cipher-text-policy attribute-based hybrid encryption with verifiable delegation scheme on outsourced data. In this scheme, due to resource constrained low computational end devices in the cloud, data owner delegate the workload of data owner to cloud servers which reduces the computation cost at data owner. Due to workload delegation the cloud server can cheat the data owners for the purpose of cost saving and his benefits.

Sahai and Waters [43], proposed an identity and attribute based encryption algorithm to provide confidentiality of the data at transit, during process and at rest on untrusted cloud server. In this scheme a data owner identified a set of attributes which acts as a secret key for the encryption and also it defines the access structure to encrypt data for confidentiality and share it to authorized users. Lai et al. [44] proposed a attribute based encryption with verifiable outsourced decryption algorithms using

commitment for the correctness of the original cipher text. In this method data owner generates a commitment message without any identity, thus the untrusted server can then forge a commitment message, therefore the ciphertext relating to this commitment message is at risk.

In [26, 45], proposed a fuzzy identity based encryption scheme. In this scheme a group of attributes identify the identity of the users. The data can decrypts only the user who has attributes that matches in the cipher-text A Key Policy Attribute Based Encryption (KP-ABE) was proposed in [46] with some attribute matching constraints than the attribute based encryption. In this scheme, to identify the identity of the user a tree based access structure linked to private key. For the decryption, the attributes in the private key must matches with attributes specified in the cipher-text The drawback of this scheme in that, if the data owner re-encrypts the data, the new private key has to be shared to all the data access users.

Cong Wang et al., [13], proposed a ranked keyword search over the outsourced cloud data for secure data access. In this method , the data owner send the data file along with list of keywords in the data and it frequency to the cloud server. The cloud server can derive the outsourced data using shared keywords and its frequency. So that, there is no privacy of the outsourced data from the untrusted cloud server. Besides, this the cloud service provider can also delete the rarely accessed data from the cloud storage for his storage space benefits.

Fatemi Moghaddam et al., in [47] discussed the performance of six different symmetric key RSA data encryption algorithms in cloud computing environment. They have proposed two separate cloud servers; one for data server and other for key

cloud server and the data encryption and decryption process at the client side. The main drawback of this method is maintaining two separate servers for data security in cloud, which creates more storage and computation overheads. To ensure that the data is only accessible for only authorized users and for end-to-end secure data transfer requires the efficient encryption and/or decryption algorithm to the whole dataset before performing any further operations. Further, it is very difficult to see the upcoming applications for the future without the proper use of algorithms. Also, several data security software need to be developed for the privacy of the organizations data.

Mazhar Ali et al., [48] proposed a secure data sharing scheme in cloud system that provides data confidentiality, secure data sharing and forward and backward data access control. The cost of data decryption is more as compared to the data encryption and maintaining multiple keys at third party server is not secure.

Secure data sharing in the public cloud is important issue in cloud computing. Xu et al., [49] addressed the issues of secured data sharing within the group and they proposed a certificate-less proxy re-encryption (CL-PRE) technique for secure data sharing in cloud storage system. In this technique the data owner is encrypted data using symmetric key algorithm and the symmetric key is encrypted using public key algorithm. The encrypted data and the key are uploaded to the cloud server, then the cloud server re-encrypt the encrypted key using public key. This re-encryption is based on the complex bilinear pairing operations. The computational cost of pairing operation is more costlier than all the standard operations in the finite fields.

Seo et al., [10] proposed a mediated certificate-less encryption technique for data

sharing the data on public cloud without using bilinear pairing operations, which reduces the computational overhead. In this approach, the key pairs is generated by cloud server and distributes the public key to all the authorized data owners. The key management and partial decryption are done by the cloud server, user handling in easier but this is not suitable for the data security point of view because of untrusted cloud server. Besides, the cloud server handles the key management and decryption operation, the computational overhead increases.

To share sensitive information in cloud Khan et al., [50] are proposed a secured data sharing method using trusted third party as a low computational resource constrained server, which manages the key generation and data access to the cloud server. The proposed method [50] utilize the El-Gamal cryptosystem and complex bilinear pairing operations, which increases the computational complexity of the data sharing at cloud server.

Chen and Tzeng et al.[51] , proposed a shared key derivation scheme for securing data sharing among a data owners in a group. This scheme uses binary tree computation operation for derivation of keys. The computational cost of key derivation in this scheme more than the re-keying scheme. Therefore this method is not suitable for cloud and it was not secure due to the collision attack.

In [52], Sana et el., presented a symmetric cryptographic lightweight encryption and decryption algorithms to encrypt data files in the cloud computing system. This scheme has key management issue and also the proposed solution is not flexible and secure when a user leave the group in order to prevent him from accessing the data.

2.3 Remote Data Verification using Protocol

In public cloud, remote data integrity checking is an important security issue. Since the clients massive data is outside of their control, the clients data may be corrupted by the malicious cloud server regardless of intentionally or unintentionally. In this section we present the recent remote data integrity verification scheme using protocol proposed by the researchers.

Ari Juels and Burton S. Kaliski Jr. [53], proposed the proofs of retrievability (PORs) scheme verifies the integrity of an archive or back-up data file on cloud server. The PORs has designed to handled large file using cryptographic hash function. In POR protocols the cost of input/output data transmission, memory access and storage requirements are independent to the length of the data during the data verification process. To verify small portion of the data file it takes more communication overhead due to download the entire file. Secondly, the auditor and verifier can derive the data file from the corresponding meta-data of the file. It is designed for only static data file and which take more computational and transmission cost to verify the few blocks of data file.

To solve data integrity issue for the data security in cloud, recently several data auditing techniques are proposed [54, 55, 26]. These data auditing techniques allows data owner to check the cloud service provider stores data correctly. The PDP schemes reserve the integrity of outsourced data, but POR is not only preserve the integrity of the data and it recovers the partially corrupted data using error correcting codes. In this scheme the data integrity is verify by sampling method, so that both the method takes more storage and computational overheads.

Later on H. Shacham and B. Waters in [56] propose the improved version of POR data auditing techniques such as private and public data verification. But in these schemes the private verification is only for data owner and also it is not support for batch auditing.

Subsequently, several PDP version of data auditing schemes [57, 58] are proposed to address the data privacy and security issues in cloud. In these method, the data auditing is performed without retrieving the enter data from the Cloud servers, which is called a public data auditing scheme. In public auditing schemes cloud servers requires more computational and storage overheads for remote data auditing process.

In [59] proposed a delegable PDP scheme for remote data verification in this method the data owner generates the delegation key for a verifier and stores on cloud server for data verification. But it does not supports for encrypted data so that there is no privacy guarantee for outsourced data.

Afterwards, Z. Mo et al., [60] presented a proxy PDP model in which the data owner delegates the data-auditing task to a proxy server by sending the meta-data of the outsourced data. Due to insecure data auditing operations on unbelievable cloud server a designated verifier PDP schemes [61, 62, 63] are presented. In this method, the delegated verification is independent from cloud server, which solve the data privacy problem, but it suffers from signature forgery attacks.

Afterwards, Tsu Yang Wu et al., [64] presented in non-repudiable PDP with designated verifier scheme to reduce the communication over between the verifier and cloud server and also address the forgery signature attacks. The cloud servers can

derive outsourced data from the metadata during auditing task.

The provable data possession (PDP) model introduced in [29], [42] [65], that allows a client that has stored data at an untrusted server to validate the integrity of data. This model generates probabilistic proofs of possession by sampling random sets of blocks from the server, which drastically reduces I/O costs. The client maintains a constant amount of metadata to verify the proof. The challenge/response protocol transmits a small, constant amount of data, which minimizes network communication. Thus, the PDP model for remote data checking supports large data sets in widely-distributed storage systems. The proposed provably-secure PDP schemes [66] that are more efficient than previous solutions, even when compared with schemes that achieve weaker guarantees. In particular, the overhead at the server is low (or even constant), as opposed to linear in the size of the data. The main drawback of these schemes is owner data will be stored in untrusted cloud servers.

Giuseppe Ateniese et al. [29], addressed the problem of storage and communication cost to verify the outsourced data without retrieving the original data file from the cloud server using provable data possession (PDP) model which minimizes the input/output cost. The data verification operation is performed on unencrypted data blocks so that there verifier can misuse data block for his benefits. In case to verify the data blocks more frequently more number of input/output requests and more computational cost is required to cloud server.

To provide the security for outsourced data and to reduce the computation and storage cost between auditor and cloud server, Cong Wang et al. in [67], proposed the Privacy-Preserving Public Auditing scheme using Trusted Third Party (TPA) and

homomorphic linear authenticator and random masking techniques during the auditing process, which reduces expensive computation cost at cloud server and provide the security of the outsourced data using masking techniques.

Wenju Lu et al., has proposed a data privacy for the outsourced personal and graphical data using homomorphic encryption and feature/index randomization techniques [68]. The homomorphic technique has analyzed the performance in terms of search, data privacy and computational overheads for smaller size of data set. In this method, the untrusted server can leak the owner data by analyzing meta data sent from the data owner at the time of outsourcing and which is applicable only for text data. The feature/index randomization technique [68] is the improved version of homomorphic technique and which applied for both text and graphical data. This scheme greatly increases the computation, communication and storage overheads of cloud server for mapping of query graph and outsourced graph for larger data size.

In other related work, Ning et al., [69] proposed the method of checking the integrity and security of the remotely stored confidential graph structured encrypted data across distributed server. To reduce the storage, computational overheads of the outsourced meta data for privacy of outsourced data, sub-graph filtering and verification method are utilized. In this method, the meta data of the data graphs and query graphs are represent the binary bit vectors. To provide privacy for data these data vectors are encrypted using the invertible inverse matrix. This method, having three issues related to server computational overhead and data privacy. Firstly, for large number of users, matching all the requested images from the receiver, the computational overhead of the server increases exponentially. Secondly, the server is not trusted, so that there is

no guarantee of data privacy in server. Finally, it is not suitable for larger size data graphs i.e., number of graph vertices should be less than 64. To address these issue , some of the server workload can assign to the intermediate trusted third party which helps to verify and retrieve the requested user data optimally.

Huaqun Wang et al. [70] proposed a solution using, delegate au thorization access and Remote data integrity checking in public cloud using Identity based Public key cryptography (IPKC) and Provable data possession (PDP). In IPKC model a key is provided to the client so that he can access his data in the cloud and in PDP model, the checker can check the remote data integrity without retrieving or downloading the whole data. The main drawbacks of this method are that, Public-key cryptography may be vulnerable to impersonation, even if users private key is not available. Motivated by the application needs, the concrete data integrity checking protocol is provably secure and efficient by using the formal security proof and efficiency analysis. On the other hand, the proposed protocol can also realize private remote data integrity checking, delegated remote data integrity checking and public remote data integrity checking based on the original clients authorization.

In PKI (public key infrastructure), provable data possession protocol needs public key certificate distribution and management. It will incur considerable overheads since the verifier will check the certificate when it checks the remote data integrity. In addition to the heavy certificate verification, the system also suffers from the other complicated certificates management such as certificates generation, delivery, revocation, renewals, etc. In cloud computing, most verifiers only have low computation capacity.

Identity-Based Distributed Provable Data Possession in Multi-Cloud Storage (ID-DPDP) for remote data integrity checking without downloading the whole data and the certificate management is proposed in [71].

The first ID-DPDP protocol is provably secure under the assumption that the CDH (computational Diffie Hellman) problem is hard. In public cloud, remote data integrity checking is an important security problem. Since the clients massive data is outside of their control, the clients data may be corrupted by the malicious cloud server regardless of intentionally or unintentionally.

In public cloud environment, the client moves its data to public cloud server (PCS) and can not control its remote data. Thus, information security is an important problem in public cloud storage, such as data confidentiality, integrity, and availability.

A Proxy Provable Data Possession in Public Clouds Solutions are proposed in [72], [62], [73] using bilinear pairing technique, an efficient proxy provable data possession (PPDP) protocol is designed, which solves the problem of remote data possession checking. This protocol is a valid lightweight remote data integrity probabilistic checking model but it cannot be applied into the field of dynamic data in their pioneering work.

To ensure the data integrity of a file consisting of a finite ordered set of data blocks in cloud server several solutions are defined by Qian Wang et al, in [74]. The first and straight forward solution to ensure the data integrity is, the data owner pre-compute the MACs for the entire file with a set of secret keys, before our sourcing data to cloud server. During auditing process, for each time the data owner reveals the secret

key to the cloud server and ask for new MAC for verification. In this method the number of verification is restricted to the number of secret keys. Once the keys are exhausted, the data owner has to retrieve the entire file from the cloud server to compute the new MACs for the remaining blocks. This method takes the huge number of communication overhead for verification of entire file, which affects the system efficiency.

The another solution to overcome the drawback of previous method, generate the signatures for every block instead of MACs to obtain the public auditability. This solution can provide probabilistic assurance of data correctness and public auditability, which again results in large communication overhead and affects the system efficiency. The above solutions support only static data and none of them can deal with the dynamic data updates. Qian Wang et al, in [74] designed an efficient solution to support the public audit-ability without retrieving the data blocks from server. The design of dynamic data operations is a challenging task for cloud storage system. They proposed a RSA signature authenticator for verification with data dynamic support. To support the efficient handling for multiple auditing task, they extend the technique of bilinear aggregate signature and they introduce a third party auditor to perform the multiple auditing task simultaneously.

In the recent resource sharing paradigm in distributed system such as cloud computing, the most challenging task in data sharing system is defining of access policies and dynamic data updating. In [75], Junbeom Hur, explains the cryptographic based solution for data sharing using cipher-text policy attribute-based encryption(CP-ABF) to improve the security of the data. In this method the data owners define the

access policies on the data to be distributed. The major drawback of this method is the unauthorized users can access the key to decrypt the encrypted data.

In cloud computing, both data and applications are controlled by the data owner and cloud service provider. To access the cloud data and applications as a cloud service more securely a data security model has been defined Mohamed, E.M. in [76]. In this security model, it provides a single default gateway as a platform to secure user data across public cloud applications. The default gateway encrypts only sensitive data using encryption algorithm, before sending in to the cloud server. In this method the data is accessed by only authorized users but the cloud service provider can grant the access permission for unauthorized users while cheating to the data owner. Therefore in this method degrades security because of proper key management is not implemented in the system.

To increase the revenue and degree of connectivity from cloud computing model while accessing and updating data from data center to the cloud user, Dubey et al. in [77] developed a system using RSA and MD5 algorithms for avoiding unauthorized users to access data from cloud server. The main drawback of this method is the cloud service provider is also an equal control of data as data owner and the computation load for cloud service provider is proportional to the degree of connectivity so that the performance of the system can degrade.

In [29] Ateniese et al., proposed a public auditing model as Provable Data Possession (PDP) for auditing data on untrusted storage entity. They introduced homomorphic linear authenticators to audit the selected outsourced data blocks of the outsourced file. This method may leak the user information to the auditor during auditing process

and which may leads to helps to derive the user information, therefore it does not provide the security for the outsourced data.

In [78] Juels et al., introduced a Proof of Retrievability (PoR) model for verification and retrieval of data from remote data storage service using error-correcting codes. This method has the following drawbacks: i) It has fixed data auditing challenges, ii) Suffers from public and delegate verification.

In [79] Yujue et al., addressed identity-based data outsourcing for distributed users. In this method of audit, the data users have to authorize the dedicated proxy before storing data on cloud server and which is more controlled way of outsourcing data on cloud server. To verify the integrity of outsourced data is more expensive.

In [30, 80, 31, 81] proposed a data auditing protocol on the cloud server to support the batch auditing on multi-servers. In these methods, individuals used data tags to the owner and these cannot help to combine multi-owner tags to conduct batch auditing. To combine these individual tags, third party auditor is introduced which takes additional computation and communication cost. Due to these overhead this method reduces the efficiency of the auditing system.

In [82], [83] proposed the data privacy protocol for auditing data in cloud storage server by using the bilinear privacy operations to verify the correctness of the response message. The drawback of this method is that, for multi-cloud auditing to segregate the data blocks of the multiple users, the auditor takes more computational task and which is a low end user entity in the cloud storage system. This method suffers from yet another drawback while using unencrypted used information for auditing process,

thereby empowering the auditor to derive user data.

2.4 Data audit using Digital Signature

Several techniques has been proposed for public data auditing with third party auditor(TPA) recent public data auditing schemes are presented in the following section.

There are various cryptographic techniques are proposed for data auditing methods using message authentication [84], homomorphic linear authenticators[85] and Boneh-Lynn-Shacham (BLS) based homomorphic methods [82]. The comparison among the several existing remote data auditing techniques is summarized in the Table 2.1 and Table 2.2. In these table we listed the cryptographic operations involved in the data auditing phase, advantages and issues of the existing techniques to audit data on cloud.

In [88] Jian Liu et al., proposed a techniques to check integrity of outsourced data and to recover corrupted data uasing a TPA (Third Party Auditing) for public auditing. The TPA checks data integrity on behalf of owner of data. Due to uploading files in an encrypted format, TPA converts this data into safe place. Provided with a semi-trusted proxy to recover a data against corruption, user can recover failed data using proxy server. There are some disadvantages in this method such as; it does not support dynamic auditing, data is not stored in encrypted format, there is heavy storage on the server, the performance evaluation of that RDC is expensive for both clients and servers.

In [90] Yan Zhu et al. proposed a cooperative Provable data possession (cPDP) for integrity checking of outsourced cloud data on distributed cloud storage [91] [71]

Table 2.1: Data Auditing Methods

Researcher	Data Audit technique	Cryptographic operations	Advantages	Issues
G. Ateniese et al., [29]	Provable Data Possession	Hash function to reduce the size of the proof message	It supports encrypted data, -small portion of the file data is required for verification of the entire file	-it supports static data only -it is probabilistic in nature -CSP can derive the outsourced data from the metadata
A.Juels et al., [53]	Proofs of Retrievability	Error correcting code is used to recover the partially corrupted file	It is able to recover the file, if the file is corrupted	It supports static data only -It requires more storage space for error correcting code
G, Ateniese et al., [86]	Scalable Provable Data Possession	-MAC code is used for authentication, SHA-2 hash function, Pseudo-random function, Pseudo-random permutation	It uses the symmetric key algorithm, which is better than public key cryptography	It supports only private verification, fixed batch size
Kevin D. Bowers et al., [87]	High-Availability and Integrity Layer	MAC code is used for authentication, Reed-Solomon codes, Error-correcting code	It supports distributed storage, Size of proof message is small	-It supports only static data, more computation and storage overheads
TAN Shuang et al., [80]	Remote data integrity checking	Hash functions and Bilinear pairing	-supports dynamic operations -low communication overhead -support batch auditing	-CSP can forge a signature of the data file. -CSP can derive data from the metadata of the file.

Table 2.2: Data Auditing Methods

Researcher	Data Audit technique	Cryptographic operations	Advantages	Issues
Jian et al., [88]	Regenerating code based auditing	-BLS signatures, GF(p), Hash function and Bilinear pairing	-supports dynamic operations -supports recovery of corrupted data blocks	-It supports only for plain text -More storage overhead -data can derive from the metadata
Jia wei et al., [25]	Public integrity auditing for dynamic data sharing	-One way hash function, Bilinear pairing	-supports both public and private data verification -low communication overhead due to signature aggregation	-It supports only text file, -CSP can derive data from the metadata
Jingwei [89]	SecCloud	Merkle Hash tree, Hash function, Bilinear pairing	-supports both encrypted and plain text, -support distributed file system	-more storage overhead due to large number of hash codes, -Fixed block size
Kan Yang et al., [82]	Dynamic data auditing protocol	Hash functions, Bilinear pairing	-supports multi-user and multi-cloud system -supports only text file	-CSP takes more computation overhead for separation of challenging message on multi-cloud

[70]. In this method of data auditing for smaller block sizes of data which takes more number of bilinear operations. Due to this complex operations to verify the remote data which take more computation and communication cost. Secondly, to maintain the metadata of the outsourced data on cloud storage server the index size of the metadata table is also grows linearly due to more number of block data blocks signature. Finally the signature generation includes complex computation operations like exponentiation and multiplication operations which increases the signature generation cost of the data blocks.

In [70] Huaqun Wang et al., proposed to perform the remote data integrity checking, it incur considerable overhead since the verifier will check the certificate when it checks the remote data integrity. In the Identity based Public key cryptography method a key is provided to the client so that he can access his data in the cloud and the provable data possession [62] [92] [93] model, the verifier can check the remote data integrity without retrieving or downloading the whole data. Public-key cryptography may be vulnerable to impersonation, even if user's private key is not available.

Yong Yu et al. [94] introduced a identity-based Remote data integrity checking (RDIC) for storing owner's data on untrusted cloud server. This method of auditing utilizes the key-homomorphic cryptographic primitive operation to verifies the out-sourced data blocks. The drawback of this method has more computational overhead at the resource constrained verifier device compared to the public verifier end.

Verification of outsourced data on cloud storage Yuan Zhang et al., [95] proposed public verification protocol using message authentication code tags of the data blocks. This method is suffers from the detection of malicious auditor so that which can

missuses the outsourced user data on a untrusted storage server.

In [79] Yujue Wang et al., proposed a workload distribution scheme such as Online/Offline Provable Data Possession data auditing on cloud server. In his method the light weights file processing computations are assigned to the offline mode and heavy weight computation to the online mode. The main drawback of this method is the segregation of light and heavy workload has a challenging task.

2.5 Summary

This chapter focuses on, the recent cloud data security and privacy issues in cloud storage service, such as data confidentiality and secure data auditing and integrity verification are proposed by the various researches till date and the research gap for proposed research work. In the next chapter we explain the background for designing the proposed data auditing methods.

Chapter -3

BACKGROUND

In this chapter, we present the basic background fundamental cryptographic operations and functional requirements for our proposed secure data auditing methods on the cloud.

3.1 Bilinear Pairing

Let G , and G_T are the multiplicative cyclic group of order p and a bilinear map $e(u, v)$ is a function which takes elements from group G , and outputs an element of other group G_T , that means $e : G \times G \rightarrow G_T$ satisfies the following conditions.

Bilinear: for all elements $u, v, c \in G$ and for all a, b integers. $e(u^a, v^b) = e(u, v)^{ab}$ and $e(cu, v) = e(c, v).e(u, v)$

Non-degenerate: there exist a generators u, v of G , such that $e(u, v) \neq 1$ in G_T .

Computability: for all elements u, v from group G there exist an efficient algorithm to compute $e(u, v)$.

Decisional Diffie-Hellman Problem: for a given generator g from the group G and (g^a, g^b, g') for all integers a and b decide the equation is true or false; $g^{ab} \stackrel{?}{=} g'$.

3.2 Symmetric Key Cryptography

Cryptography is an art of coding information into secrets by converting intelligible text into unintelligible text and vice-versa. It is mainly used to provide a security

and privacy of data with respect to data confidentiality, data integrity, and data origin authentication in the domain of the military, diplomatic and governmental secret services, etc.,.

Based on the relation between the pair of keys involved in the message encryption and decryption, different types of cryptographic schemes (symmetric and asymmetric) are derived in the cryptography.

In cloud computing, the cryptography employs encryption techniques to secure sensitive data stored on remote servers in the cloud. It allows authorized users to securely access shares data and cloud services protected with encryption. In the cloud, data owner has no physical control over the remotely stored data and services, so that, to ensure the privacy of data at rest, motion, and process data has to be stored cryptographically. The general block diagram of the symmetric cryptographic algorithm as shown in Figure 3.1. The symmetric encryption algorithm is represented as function $c = E(p, k)$ and decryption algorithm is represented as $p = D(c, k)$, where c is the cipher-text, p is the plain text and k is the symmetric key.

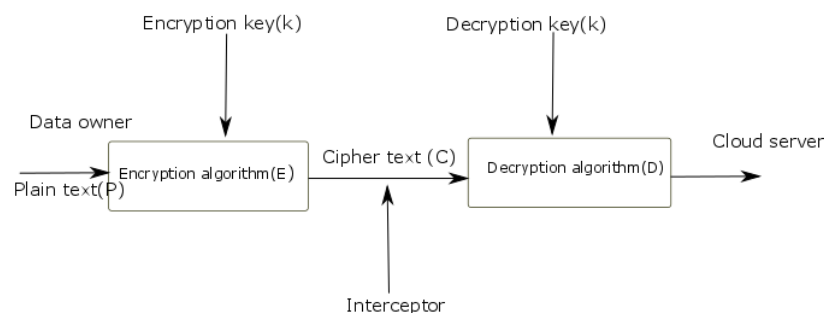


Figure 3.1: Block diagram of a symmetric key cryptography

3.3 Message Authentication Code(MAC)

A message authentication code (MAC) is used to maintain integrity, identity, and authentication of the data owner. The MAC codes are generated from hash functions, which include hash value and the message. The block diagram of data integrity verification using MAC as shown in Figure 3.2.

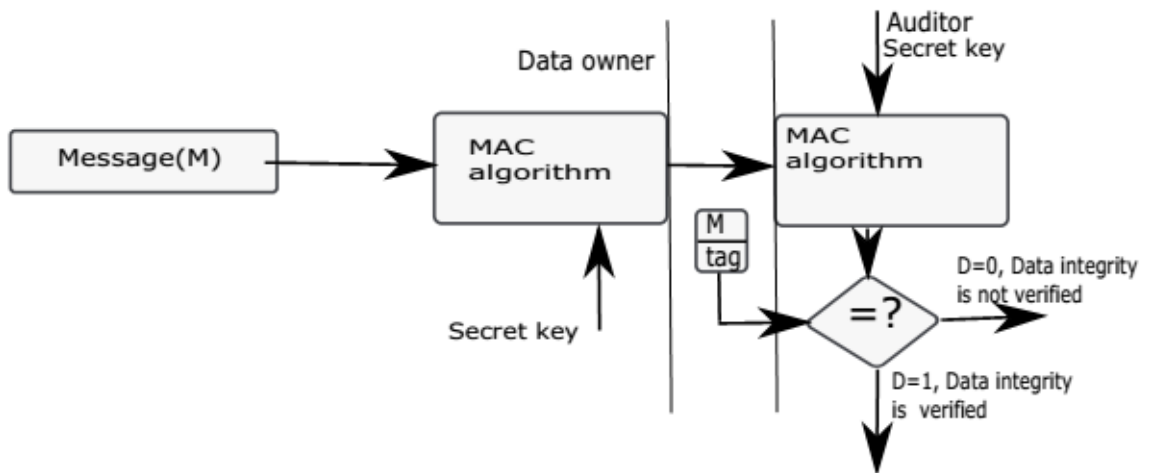


Figure 3.2: MAC block diagram

Before outsourcing data to a cloud server, the owner splits the data file into data blocks and generates the MAC for each block using a secret key (k), then send data blocks and MAC's to the server and corresponding MAC secret key to the auditor. In cloud computing, it is very simple to use to verify the integrity and authentication of the outsourced data. For verification, auditor retrieves the data blocks from the server and computes the verification message using the symmetric key k . Although a MAC preserves the data integrity, it lost the data privacy. And also intruders can attack the data or share it with others. The basic data auditing process used in cloud computing as follows;

1. Generate the secret key (k).

2. Data owner prepare the MAC codes (tag) for the data blocks using the private key. $tag = MAC_k(M)$
3. the data owner sends the data blocks, MACs and the corresponding secret keys to the server.
4. Auditor sends the data auditing request to the server
5. server prepare the corresponding response message $D(0 \text{ or } 1)$ and send to the auditor. $D = MAC_k(M, tag)$
6. if auditor receives 1, then the message is correct, otherwise, the message is altered.

3.3.1 Restricted MAC method

To overcome the drawbacks of the MAC method and privacy of the outsourced data, owner restrict the number of verification for equality checking. In this method, data owner generates a set of MAC secrete keys $\{sk_i\}$ for entire file F i.e $MAC_{sk_i}\{F\}$ where, $i = 1$ to S . After preparing meta-data for the file $\{MAC$ and $sk\}$, the owner outsource file to the server and publish meta-data to the auditor. For each audit, the auditor request file's MAC to verify data on the server using a MAC secret key sk .

3.4 Homomorphic Linear Authenticator method

To improve the proof of data possession/Resolvability of outsourced data Ateniese et al., [29] proposed a homomorphic linear authenticate (HLA) code scheme, which is one of the efficient approaches in cloud storage model.

Homomorphic authentication (HA) is a block less data verification, which means auditor verifies the data without processing data and meta-data of the block. In this method, the data owner generates the tag for each data block or file and stored at server. Then, the server can generate the proof by adding a linear combination of tag values. The HA perform three task in the data auditing process such as; aggregate signatures, generate homomorphic signature and verification.

The HA consists of four algorithms such as; key generation, meta-data generation, prove and verify.

1. The *key-generation* algorithm is executed by the data owner in the initial set up stage for proof of storage. It takes input as a security parameter and outputs public key and private key.
2. The *meta-data* generation takes a file and secret key as input and generates the encoded file and state information.
3. The *prove* algorithm sends the challenge message along with the public key and encoded file then it generates a response message for the requested challenge.
4. The *verify* algorithm verifies the data and generates 1 or 0 if data are correct or no:

With these algorithms, the secret key is not needed during verification of data block. The general block diagram of data auditing process as shown in Figure 3.3.

3.4.1 Initial File Setup Phase

The initial file setup for remote data auditing scheme consists of *key generation* and *meta-data generation* stage. The design process of these stages as follows.

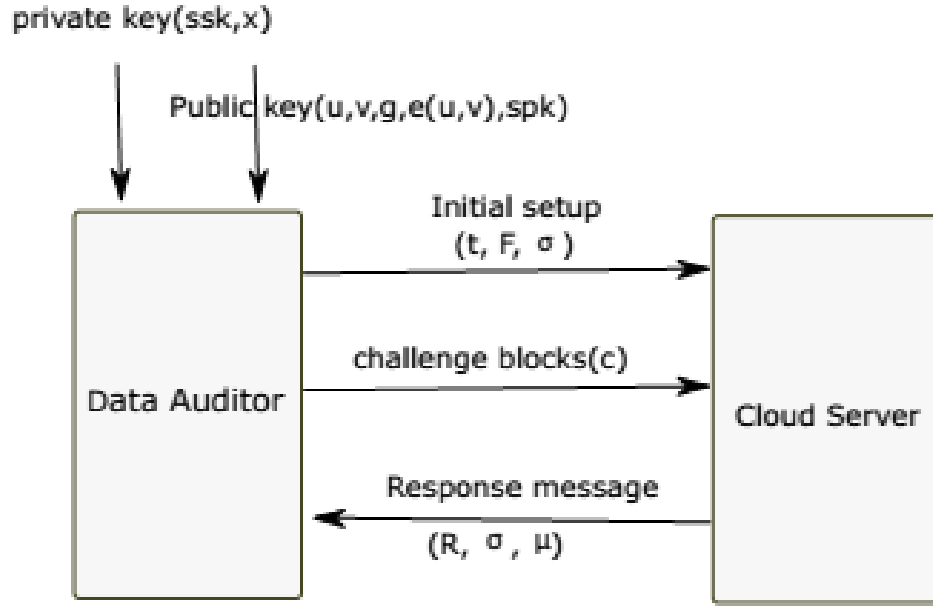


Figure 3.3: Basic Block Diagram of Data Auditing using HLA's

key generator: Initially the data owner chooses a random signing key(ssk, spk), two random integer numbers x and u . Then the user computes $v = g^x$ and generates the private and public parameters as $sk = (x, ssk)$ and $pk = (u, v, g, e(u, v), spk)$ respectively.

Meta-data generation: For a given file (F), the data owner generates the authenticator as follows.

$$F = \begin{bmatrix} m_{11} & \cdots & m_{1s} \\ m_{21} & \cdots & m_{2s} \\ \vdots & \ddots & \vdots \\ m_{n1} & \cdots & m_{ns} \end{bmatrix} \text{ where } m_{ij} \text{ is } i^{th} \text{ block and } j^{th} \text{ sector file block}$$

$$F' = \begin{bmatrix} m'_{11} & \cdots & m'_{1s} \\ m'_{21} & \cdots & m'_{2s} \\ \vdots & \ddots & \vdots \\ m'_{n1} & \cdots & m'_{ns} \end{bmatrix} \text{ where } m'_{ij} = h(m_{ij}) \text{ } F' \text{ is hash values of the file blocks}$$

$$\sigma = \begin{bmatrix} \sigma'_{11} & \cdots & \sigma'_{1s} \\ \sigma'_{21} & \cdots & \sigma'_{2s} \\ \vdots & \ddots & \vdots \\ \sigma'_{n1} & \cdots & \sigma'_{ns} \end{bmatrix} \text{ where } \sigma'_{ij} = (h(\text{filename}|i).u^{m'_{ij}})^x \text{ mod } p$$

Data owner stores the file F and its meta-data (σ, t) on a cloud server.

3.4.2 Data audit phase

In the data audit phase, the following sequence of operation is performed between auditor and cloud server.

1. the data owner or third party auditor first verifies the file name using file tag (t) , which is stored on the cloud server.
2. After verification of the file name, the auditor sends challenge message $c = \{V_i, i\}$ to the cloud server.
3. Once the server receives the challenged blocks, it prepares the response message (R, μ, σ) , where $\mu' = \sum_c V_i m_i \text{ mod } p$, $\mu = r + \Upsilon \mu' \text{ mod } p$, $R = e(u, v)^r$ and $\Upsilon = h(R)$, $\sigma = \prod_{i \in c} \prod_{j \in s} (\sigma'_{ij})^{V_i}$ and sends it to the auditor.
4. After receiving the response message, the auditor verifies the message using the following equation.

$$R = e(\sigma^Y, g) \stackrel{?}{=} e((\Pi_c h(filename|i)^{V_i})^Y . u^\mu, v) \quad (3.1)$$

3.5 ECDSA Digital Signature

The traditional authentication schemes use symmetric algorithms, that requires secret keys. The management and protection of secret keys in the symmetric scheme can be a challenging task. To overcome this task a asymmetric key cryptography was introduced.

A Digital signature is a public-key cryptographic technique that binds a personal identity to the digital data. It is a cryptographic value that is calculated from the data and a private key known by the signer. The identity of the data can be verified independently by the receiver and third-party auditor.

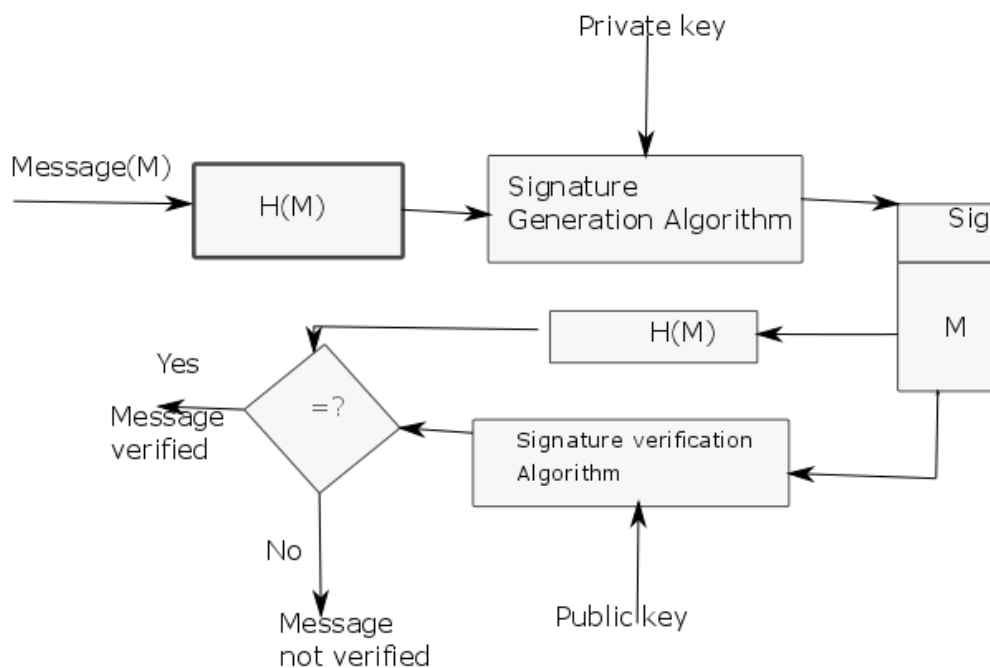


Figure 3.4: Digital signature block diagram

The general block diagram of the public-key digital signature scheme is shown in Figure 3.4. To generate a signature of any message the digital signature scheme includes two stages such as; signature generation stage and signature verification stage.

A data owner (signer) uses the signature generation algorithm to generate a digital signature of data and the data verifier uses the signature verification algorithm to verify the authenticity of the signature. Each data owner has a private and public key pair. The data owner generates the digital signature using private key and the verifier verify the digital signature using public key. The private key is a secret key and it is known by the data owner. In both the signature generation and verification algorithms, the signed data is converted in to fixed length by using secure hash function(SHA-512). The digital signature and the original message are send to the verifier. The verifier using the public key generated by the data owner to verify the validity of the signature received from the data owner.

Cryptography has symmetric and asymmetric authentication scheme to verify the validity of the message. The symmetric scheme depends on the secret key shared between sender and verifier but asymmetric scheme, digital signature generation depends on the private key and the verification depends on the public key. Elliptic Curve Digital Signature Algorithm (ECDSA) is one of the flexible data authentication methods in the asymmetric cryptographic system.

3.5.1 Elliptic Curves

In Elliptic curves cryptographic digital signature algorithms are use two types of a curve such as pseudo-random curves defined over prime fields $GF(p)$ and binary fields $GF(2^m)$. The domain parameters for ECDSA are (p, a, b, G, n, h) , where

- p is the field size;
- a and b are two field elements which represent the curve coefficients;
- G is a base point of prime order on the curve (i.e., $G = (x_G, y_G)$),
- n is the order of the point G
- h is the cofactor (which is equal to the order of the curve divided by n).

For example, Table 3.1 shows the domain parameters of curve $P-192$, which is a pseudo-random curve over a prime field.

Table 3.1: Domain parameters

Parameter Name	Value
Prime Modulus (p)	62771017353866807638357894232076664 16083908700390324961279
Prime Order n	62771017353866807638357894231760590 13767194773182842284081
Coefficient a	-3
Coefficient b	64210519 E59C80E7 0FA7E9AB 72243049 FEB8DEEC C146B9B1
x coordinate of Base Point $G(x, y)$	188DA80E B03090F6 7CBF20EB 43A18800 F4FF0AFD 82FF1012
y coordinate of Base Point $G(x, y)$	07192B95 FFC8DA78 631011ED 6B24CDD5 73F977A1 1E794811

3.5.2 Mathematical Background

The elliptic curve cryptography involves various points operations over the curve points. The points addition, multiplication and inverse operations over $GF(p)$ are needed for key generations, signature generation and verification in ECDSA scheme.

The various point operation on an elliptic curve over $GF(p)$ as follows;

Points addition: If a and b are in the finite field over prime $F_p(a, b)$, then $a + b = r$, where r is the remainder when $a + b$ is divided by p and $0 \leq r \leq (p - 1)$ known as addition modulo p .

If $P(x_1, y_2)$ and $Q(x_2, y_2)$ are two points in finite field F_p , and $O(x, y)$ is the point at infinity over the curve. Then the addition of curve points as follows.

1. If $P \neq Q$ then $P + Q = R(x_3, y_3)$

where $x_3 = s^2 - x_1 - x_2$ and $y_3 = s(x_1 - x_3) - y_1$, $s = \frac{(y_2 - y_1)}{(x_2 - x_1)}$

2. If P is (x_1, y_1) and $-P$ is $(x_1, -y_1)$ then $P - P = O$

3. $P + O = O + P = P$

Points Multiplication: If $a, b \in F_p$ then $a.b = s$, where s is the remainder when $a.b$ is divided by p and $0 \leq s \leq p - 1$ known as multiplication modulo p .

Let $P = (x_1, y_1)$, where $P \neq -P$. Then $2P = (x_3, y_3)$ where, $x_3 = s^2 - 2x_1$ and $y_3 = s(x_1 - x_3) - y_1$ and $s = \frac{(3x_1^2 + a)}{2y_1}$

Let consider an example of Elliptic curve ($y^2 = x^3 + x + 1 \pmod{23}$) over finite field $F_{23}(1, 1)$, the curve point distribution as shown in Figure 3.5 and the addition of curve points over the finite field $F_{23}(1, 1)$ as shown in Figure 3.6

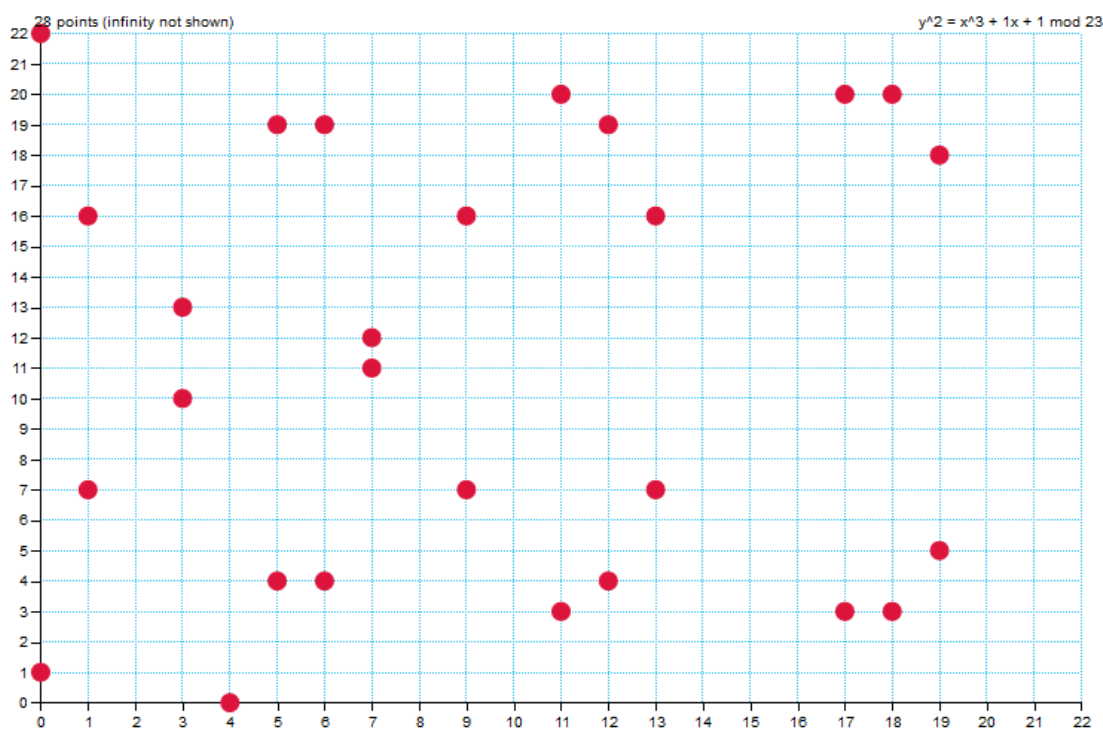


Figure 3.5: Elliptic curve: $(y^2 = x^3 + x + 1 \pmod{23})$

3.5.3 ECDSA-Algorithm

The Elliptic Curve Digital Signature Algorithm (ECDSA) is an asymmetric key cryptographic algorithm in which the data owner uses the private key to create a signature of the message and the verifier uses the public key to verify the authenticator. The designing method of ECDSA algorithms is explained in the following section.

Key Pair Generation: To generate a signature of a message the signer needs to generate the private key using curve base point $g(x, y)$. The public key is derived from the private key and the curve domain parameters. The private key must keep secret at signer side and the public key is openly accessible by the verifier. The key pair generation process as follows;

1. select a random number d from the points $[1 \text{ to } n]$

+	=	(0,1)	(0,22)	(1,7)	(1,16)	(3,10)	(3,13)	(4,0)	(5,4)	(5,19)	(6,4)	(6,19)	(7,11)	(7,12)	(9,7)	(9,16)	(11,3)	(11,20)	(12,4)	(12,19)	(13,7)	(13,16)	(17,3)	(17,20)	(18,3)	(18,20)	(19,5)	(19,18)
=	=	(0,1)	(0,22)	(1,7)	(1,16)	(3,10)	(3,13)	(4,0)	(5,4)	(5,19)	(6,4)	(6,19)	(7,11)	(7,12)	(9,7)	(9,16)	(11,3)	(11,20)	(12,4)	(12,19)	(13,7)	(13,16)	(17,3)	(17,20)	(18,3)	(18,20)	(19,5)	(19,18)
(0,1)	(0,1)	(6,19)	=	(12,19)	(17,20)	(6,4)	(13,16)	(9,7)	(11,20)	(19,18)	(0,22)	(3,13)	(11,3)	(18,20)	(17,3)	(4,0)	(5,19)	(7,12)	(1,16)	(19,5)	(3,10)	(18,3)	(1,7)	(9,16)	(7,11)	(13,7)	(5,4)	(12,4)
(0,22)	(0,22)	=	(6,4)	(17,3)	(12,4)	(13,7)	(6,19)	(9,16)	(19,5)	(11,3)	(3,10)	(0,1)	(18,3)	(11,20)	(4,0)	(17,20)	(7,11)	(5,4)	(19,18)	(1,7)	(18,20)	(3,13)	(9,7)	(1,16)	(13,16)	(7,12)	(12,19)	(5,19)
(1,7)	(1,7)	(12,19)	(17,3)	(7,11)	=	(4,0)	(5,4)	(3,13)	(19,18)	(3,10)	(9,7)	(19,5)	(18,20)	(1,16)	(13,16)	(6,19)	(13,7)	(12,4)	(0,22)	(11,3)	(9,16)	(11,20)	(18,3)	(0,1)	(7,12)	(17,20)	(5,19)	(6,4)
(1,16)	(1,16)	(17,20)	(12,4)	=	(7,12)	(5,19)	(4,0)	(3,10)	(3,13)	(19,5)	(19,18)	(9,16)	(1,7)	(18,3)	(6,4)	(13,7)	(12,19)	(13,16)	(11,20)	(0,1)	(11,3)	(9,7)	(0,22)	(18,20)	(17,3)	(7,11)	(6,19)	(5,4)
(3,10)	(3,10)	(6,4)	(13,7)	(4,0)	(5,19)	(7,12)	=	(1,16)	(1,7)	(18,3)	(18,20)	(0,22)	(3,13)	(19,5)	(17,20)	(12,4)	(13,16)	(12,19)	(11,3)	(9,7)	(11,20)	(0,1)	(9,16)	(19,18)	(6,19)	(5,4)	(17,3)	(7,11)
(3,13)	(3,13)	(13,16)	(6,19)	(5,4)	(4,0)	=	(7,11)	(1,7)	(18,20)	(1,16)	(0,1)	(18,3)	(19,18)	(3,10)	(12,19)	(17,3)	(12,4)	(13,7)	(9,16)	(11,20)	(0,22)	(11,3)	(19,5)	(9,7)	(5,19)	(6,4)	(7,12)	(17,20)
(4,0)	(4,0)	(9,7)	(9,16)	(3,10)	(1,16)	(1,7)	=	(7,11)	(7,12)	(17,20)	(17,3)	(5,4)	(5,19)	(0,1)	(0,22)	(11,20)	(11,3)	(13,7)	(13,16)	(12,4)	(12,19)	(6,19)	(6,4)	(19,5)	(19,18)	(18,3)	(18,20)	(18,3)
(5,4)	(5,4)	(11,20)	(19,5)	(19,18)	(3,13)	(4,7)	(18,20)	(7,11)	(17,20)	=	(12,19)	(7,12)	(6,4)	(4,0)	(11,3)	(18,3)	(0,22)	(9,16)	(6,19)	(12,4)	(17,3)	(13,7)	(5,19)	(13,16)	(3,10)	(9,7)	(1,16)	(0,1)
(5,19)	(5,19)	(19,18)	(11,3)	(3,10)	(19,5)	(18,3)	(1,16)	(7,12)	=	(17,3)	(7,11)	(12,4)	(4,0)	(6,19)	(18,20)	(11,20)	(9,7)	(0,1)	(12,19)	(6,4)	(13,16)	(17,20)	(13,7)	(5,4)	(9,16)	(3,13)	(0,22)	(1,7)
(6,4)	(6,4)	(0,22)	(3,10)	(9,7)	(19,18)	(18,20)	(0,1)	(17,20)	(12,19)	(7,11)	(13,7)	=	(13,16)	(5,4)	(9,16)	(1,16)	(18,3)	(19,5)	(5,19)	(17,3)	(7,12)	(6,19)	(4,0)	(12,4)	(3,13)	(11,20)	(1,7)	(11,3)
(6,19)	(6,19)	(3,13)	(0,1)	(19,5)	(9,16)	(0,22)	(18,3)	(17,3)	(7,12)	(12,4)	=	(13,16)	(5,19)	(13,7)	(1,7)	(9,7)	(19,18)	(18,20)	(17,20)	(5,4)	(6,4)	(7,11)	(12,19)	(4,0)	(11,3)	(3,10)	(11,20)	(1,16)
(7,11)	(7,11)	(11,3)	(18,3)	(18,20)	(1,7)	(3,13)	(19,18)	(5,4)	(6,4)	(4,0)	(13,16)	(5,19)	(17,20)	=	(11,20)	(19,5)	(9,16)	(0,22)	(17,3)	(13,7)	(6,19)	(12,4)	(7,12)	(12,19)	(1,16)	(0,1)	(3,10)	(9,7)
(7,12)	(7,12)	(18,20)	(11,20)	(1,16)	(18,3)	(19,5)	(3,10)	(5,19)	(4,0)	(6,19)	(5,4)	(13,7)	=	(17,3)	(19,18)	(11,3)	(0,1)	(9,7)	(13,16)	(17,20)	(12,19)	(6,4)	(12,4)	(7,11)	(0,22)	(1,7)	(9,16)	(3,13)
(9,7)	(9,7)	(17,3)	(4,0)	(13,16)	(6,4)	(17,20)	(12,19)	(0,1)	(11,3)	(18,20)	(9,16)	(1,7)	(11,20)	(19,18)	(6,19)	=	(7,12)	(5,19)	(3,10)	(18,3)	(1,16)	(19,5)	(3,13)	(0,22)	(5,4)	(12,4)	(7,11)	(13,7)
(9,16)	(9,16)	(4,0)	(17,20)	(6,19)	(13,7)	(12,4)	(17,3)	(0,22)	(18,3)	(11,20)	(1,16)	(9,7)	(19,5)	(11,3)	=	(6,4)	(5,4)	(7,11)	(18,20)	(3,13)	(19,18)	(1,7)	(0,1)	(3,10)	(12,19)	(5,19)	(13,16)	(7,12)
(11,3)	(11,3)	(5,19)	(7,11)	(13,7)	(12,19)	(13,16)	(12,4)	(11,20)	(0,22)	(9,7)	(18,3)	(19,18)	(9,16)	(0,1)	(7,12)	(5,4)	(4,0)	=	(1,7)	(3,10)	(3,13)	(1,16)	(18,20)	(19,5)	(17,20)	(6,19)	(6,4)	(17,3)
(11,20)	(11,20)	(7,12)	(5,4)	(12,4)	(13,16)	(12,19)	(13,7)	(11,3)	(9,16)	(0,1)	(19,5)	(18,20)	(0,22)	(9,7)	(5,19)	(7,11)	=	(4,0)	(3,13)	(1,16)	(1,7)	(3,10)	(19,18)	(18,3)	(6,4)	(17,3)	(17,20)	(6,19)
(12,4)	(12,4)	(1,16)	(19,18)	(0,22)	(11,20)	(11,3)	(9,16)	(13,7)	(6,19)	(12,19)	(5,19)	(17,20)	(17,3)	(13,16)	(3,10)	(18,20)	(1,7)	(3,13)	(5,4)	=	(7,11)	(4,0)	(6,4)	(7,12)	(9,7)	(18,3)	(0,1)	(19,5)
(12,19)	(12,19)	(19,5)	(1,7)	(11,3)	(0,1)	(9,7)	(11,20)	(13,16)	(12,4)	(6,4)	(17,3)	(5,4)	(13,7)	(17,20)	(18,3)	(3,13)	(3,10)	(1,16)	=	(17,3)	(4,0)	(7,12)	(7,11)	(6,19)	(18,20)	(9,16)	(19,18)	(0,22)
(13,7)	(13,7)	(3,10)	(18,20)	(9,16)	(11,3)	(11,20)	(0,22)	(12,4)	(17,3)	(13,16)	(7,12)	(6,4)	(6,19)	(12,19)	(1,16)	(19,18)	(3,13)	(1,7)	(7,11)	(4,0)	(5,4)	=	(17,20)	(5,19)	(0,1)	(19,5)	(9,7)	(18,3)
(13,16)	(13,16)	(18,3)	(3,13)	(11,20)	(9,7)	(0,1)	(11,3)	(12,19)	(13,7)	(17,20)	(6,19)	(7,11)	(12,4)	(6,4)	(19,5)	(1,7)	(1,16)	(3,10)	(4,0)	(7,12)	=	(5,19)	(5,4)	(17,3)	(19,18)	(0,22)	(18,20)	(9,16)
(17,3)	(17,3)	(1,7)	(9,7)	(18,3)	(0,22)	(9,16)	(19,5)	(6,19)	(5,19)	(13,7)	(4,0)	(12,19)	(7,12)	(12,4)	(3,13)	(0,1)	(18,20)	(19,18)	(6,4)	(7,11)	(17,20)	(5,4)	(13,16)	=	(11,20)	(1,16)	(11,3)	(3,10)
(17,20)	(17,20)	(9,16)	(1,16)	(0,1)	(18,20)	(19,18)	(9,7)	(6,4)	(13,16)	(5,4)	(12,4)	(4,0)	(12,19)	(7,11)	(0,22)	(3,10)	(19,5)	(18,3)	(7,12)	(6,19)	(5,19)	(17,3)	=	(13,7)	(1,7)	(11,3)	(3,13)	(11,20)
(18,3)	(18,3)	(7,11)	(13,16)	(7,12)	(17,3)	(6,19)	(5,19)	(19,5)	(3,10)	(9,16)	(3,13)	(11,3)	(1,16)	(0,22)	(5,4)	(12,19)	(17,20)	(6,4)	(9,7)	(18,20)	(0,1)	(19,18)	(11,20)	(1,7)	(12,4)	=	(13,7)	(4,0)
(18,20)	(18,20)	(13,7)	(7,12)	(17,20)	(7,11)	(5,4)	(6,4)	(19,18)	(9,7)	(3,13)	(11,20)	(3,10)	(0,1)	(1,7)	(12,4)	(5,19)	(6,19)	(17,3)	(18,3)	(9,16)	(19,5)	(0,22)	(1,16)	(11,3)	=	(12,19)	(4,0)	(13,16)
(19,5)	(19,5)	(5,4)	(12,19)	(5,19)	(6,19)	(17,3)	(7,12)	(18,3)	(1,16)	(0,22)	(1,7)	(11,20)	(3,10)	(9,16)	(7,11)	(13,16)	(6,4)	(17,20)	(0,1)	(19,18)	(9,7)	(18,20)	(11,3)	(3,13)	(13,7)	(4,0)	(12,4)	=
(19,18)	(19,18)	(12,4)	(5,19)	(6,4)	(5,4)	(7,11)	(17,20)	(18,20)	(0,1)	(1,7)	(11,3)	(1,16)	(9,7)	(3,13)	(13,7)	(7,12)	(17,3)	(6,19)	(19,5)	(0,22)	(18,3)	(9,16)	(3,10)	(11,20)	(4,0)	(13,16)	=	(12,19)

Figure 3.6: Elliptic curve point addition on Finite field F_{23}

2. compute $c(x, y) = d * g(x, y)$ by adding point $g(x, y)$ itself d times.
3. Return the public key as c and private key as d

Signature Generation: To generate a signature of a message (M) using private key and domain parameters as follows;

1. Find the hash value of the message M i.e $z = H(M)$.
2. Initialize the signatures $r = s = 0$
3. While r and s are not equal to 0 do the following operations
 - Select a random integer number k between 1 and $n - 1$.
 - $P(x, y) = k * g(x, y)$
 - $r = x \% n$
 - $S = ((z + r * d) * k^{-1}) \% n$

4. End while
5. Return r and s

Signature Verification: The signature verification is the important step in the signature computation at the verifier. It verifies the message authenticity using public key, signatures and the domain parameters. The signature verification steps as follows.

1. find the hash value of message M , i.e, $z = H(M)$.
2. compute x_2 and y_2
 - $w = s^{-1} \bmod n$
 - $u_1 = (h(M) * w) \bmod n$
 - $u_2 = (r * w) \bmod n$
 - $(x_2, y_2) = (u_1 * g(x, y) + u_2 * c(x, y)) \bmod n$
3. if x_2 is equal to r then verification is successful, otherwise, verification is unsuccessful using the private key.

3.6 Summary

In this chapter, we present a general introduction to symmetric and public key cryptography. As a specific variant of public cryptography, we describe the MAC, Homomorphic Linear authentication and elliptic curve digital signature schemes. In the next chapter, we define the statement of the problem with the objectives consider in the proposed system model.

Chapter -4

STATEMENT OF THE PROBLEM

With the rapid development of the Information Technology, cloud storage service (such as AWS Simple Storage Service, Block Storage Service [96], Google Drive[97], DropBox [98], etc.,) is one of the most significant services in cloud computing in our daily life. It enables data owners to store the data in the remote cloud storage system without the burden of local infrastructure, maintenance and is shared over the internet making it economically more viable. The data owner outsources the data to the remote storage server, the physical control of the data will be lost. So that the data confidentiality and integrity challenges have a significant influence on the data security and privacy of cloud storage systems.

One major data security issue is how to ensure the confidentiality and integrity of the outsourced data on cloud storage server. For example, due to hardware or software failures, external or internal attacks, the cloud server may lose the owners data. However, because of the reputation of the service the cloud service provider can hide the administrative errors to the data owners.

In this chapter, we define the system model, security model, objectives and contribution of the proposed cloud storage data auditing scheme with the third-party auditor in cloud computing.

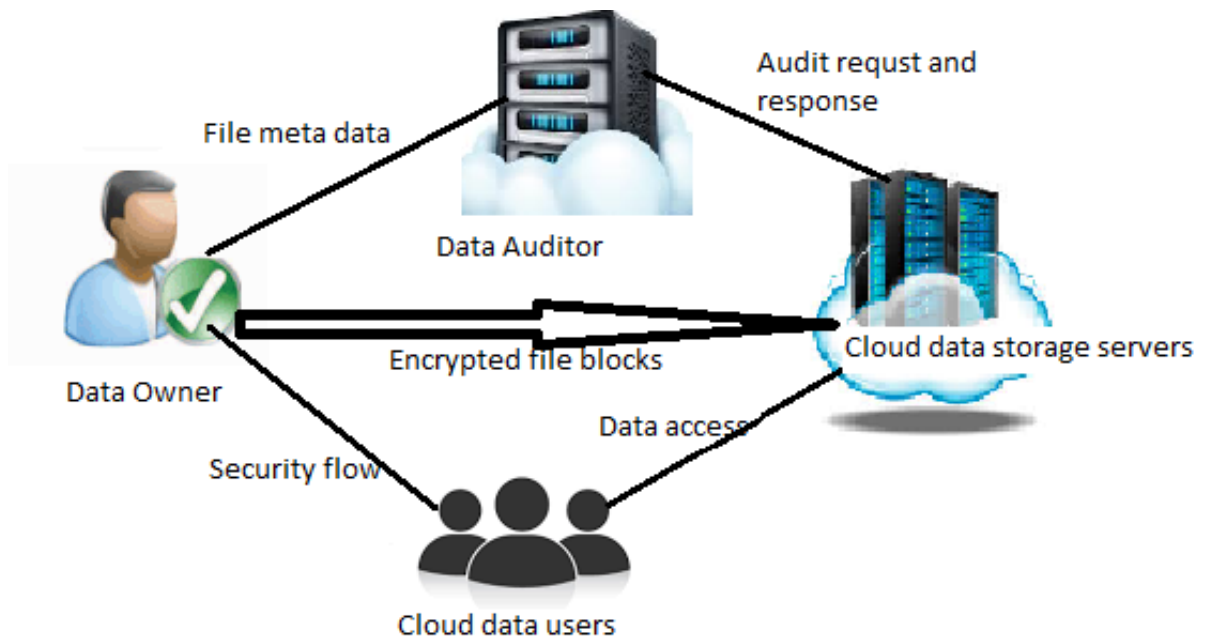


Figure 4.1: Proposed cloud System Model

4.1 System Model

Consider a cloud storage system in Figure 4.1 consists of three computing entities such as; data owner, public verifier (data auditor), cloud service provider, and data users. Cloud storage system permits the data owners to store, retrieve and share data with users. The detailed functions of these entities are as follows.

Data owner: The data owner can be any organization or an individual user who outsources the data to be stored in the data center. A unique identifier identifies each data owner.

Data Auditor/verifier: It is an entity who is trusted by all other entities of the system such as cloud service provider (CSP) and data owner. The functions of the verifier is to verify whether the requested user is authorized or not and to check the correctness of outsourced data using the Decisional Diffie-Hellman method.

Cloud servers: It is a set of servers, which is managed by the cloud service provider to provide the storage service, which has massive compute and storage facility. It coordinates with the trusted third party to verify the authorized users and to retrieve the data from the cloud server to make them available for the authorized user on demand.

Data Users: An authorized user, who can access the outsourced data based on the request.

The initial file setup and the auditing process has followed the following steps.

1. The data owner splits the file (F) into n blocks of size s i.e $F_i = \{m_{ij}\}$ where $i = 1$ to n and $j = 1$ to s . Each block is encrypted using encryption algorithm with a specific key and upload each block to the CSP. The data owner also sends the meta-data to the CSP.
2. The data owner then uploads the private key as well as the meta information of the uploaded file to the public verifier
3. The public auditor sends the data audit request message to the CSP regarding a specific file using its ID.
4. The CSP sends the file tag for the requested file ID. Once the tag is verified, the public verifier shall query file blocks to the CSP. The CSP shall send the digital signatures of the requested blocks. Once the tag and all the signatures are verified, the public verifier shall confirm the validity of the file in CSP.
5. When the file data blocks are verified by public verifier then it sends the status of the auditing to the data owner.

4.2 Security model

In the proposed cloud data storage system model, the Third Party Auditor (TPA) and Cloud Service Providers (CSP) are semi-trusted entities, which means they are honest but are curious about the received data. Since they are semi-trusted entities they are prone to data attacks. The following attacks are considered in the proposed data auditing scheme.

1. *Replace Attack*: The cloud server can replace any data block and its signature with other valid data block and its signature due to some hardware, software or internal attacks on data.
2. *Response Attack*: The semi-trusted server may generate the response message for the requested data blocks from the previous audits without using the actual owner's data.
3. *Data Attack*: The server and auditor can derive the user's data using meta-data information in the frequently auditing task.
4. *Signature Forge Attack*: The cloud server can forge the metadata of data block and replace with another valid block.

4.3 Objectives

In the proposed method, the following objectives are achieved for auditing outsourced data on cloud storage server. In our proposed method we achieved the following objectives for auditing data on cloud storage.

1. *Privacy of the data:* To design a secured data auditing scheme, in which the verifier cannot derive owner's data from its meta-data.
2. *Flexible data auditing:* To design a private or public data verification method, which can be applied based on the priority of outsourced data.
3. *Block level data operation:* To design a secure block level encrypted data operations.
4. *Lightweight overhead:* To optimize the storage, computation and communication overhead to perform the data auditing on a cloud server.

4.4 Contribution

The contribution of our proposed secure data auditing methods for cloud data storage system focuses on the following aspects.

We propose the lightweight data encryption and decryption algorithm using key rotation technique to provide privacy and security of remote data on an untrusted cloud server. For remote data integrity checking we have presented identity-based and linear authentication based data routing protocol on the cloud.

Finally, to reduce the computation and communication overheads we also propose the data auditing method using ECDSA digital signature method. Besides, remote data verification it also detects the corrupted data blocks during the verification. Our proposed method can realize both private and public data verification.

4.5 Summary

This chapter presents the proposed cloud data storage system model, different types of data security attacks, list of objectives and the contribution of the proposed work. In the next chapter, we discuss the detailed design methodology and algorithms for remote data auditing techniques.

Chapter -5

PROPOSED DATA AUDITING SCHEMES

5.1 Introduction

In modern computing technology, the cloud-computing paradigm is an important technology used to provide various remote services such as computing, storage, memory and other services with low computing cost as compared with many traditional approaches. There are various cloud service providers available in recent days including Amazon Web Services, Microsoft Azure, Google Cloud Platform and IBM Cloud that provide storage as a service. Storage as a Cloud service is one of the important features of cloud computing used to share user's data across the network. The cloud servers examine the outsourced data very frequently because the data can be lost or corrupted due to hardware failure, software failure or from the assailants [99] [100].

Maintaining the integrity of the outsourced data in a cloud server is an important issue in cloud computing [74]. In order to maintain the reputation of the cloud service, the cloud service providers set access restrictions on the services it provides to the users. To avoid loss profit of the service and to maintain the quality of the cloud service, verification of the integrity of outsourced data becomes mandatory before data utilization.

To verify the integrity of outsourced data, various traditional approaches such as RSA, hash functions, MAC, digital signature [19] [101] [102] are proposed. These existing approaches are retrieved the entire data from the servers to verify the correctness

of the outsourced data, so that the auditor can derive the user data from this information and it takes more computation and communication cost, which can degrade the efficiency of the system. Therefore, the traditional integrity checking approaches are not suitable for cloud computing to utilize the resources optimally [103]. In general, the size of the data is very large for downloading the server and verification of data integrity would demand availability of more resources.

This chapter discusses the detailed design methodology for secure data auditing techniques using identity-based [104], linear authenticator-based protocol [105] and RDADS scheme [106] using ECDSA digital signature algorithm.

5.2 Data Encryption using Key Rotation

Cloud computing provides on-demand resource access from a shared pool of computing resources such as; hardware and software for efficient manage. By outsourcing the user data to the public cloud environment, which decreases the control of data for data owner. To maintain the control of data in rest or data in motion within networks, offers more advantages for data security.

Protecting data in the cloud, authentication and integrity, access control, encryption, integrity checking and data masking are some of the data protection techniques. Cryptography is one of the efficient methods for data security in cloud computing. Which includes the design and implementation of an efficient encryption and decryption algorithms. In symmetric cryptography, before outsourcing data to the cloud server is encrypted into cypher text using a secret key and later user decrypted using the same shared secret key.

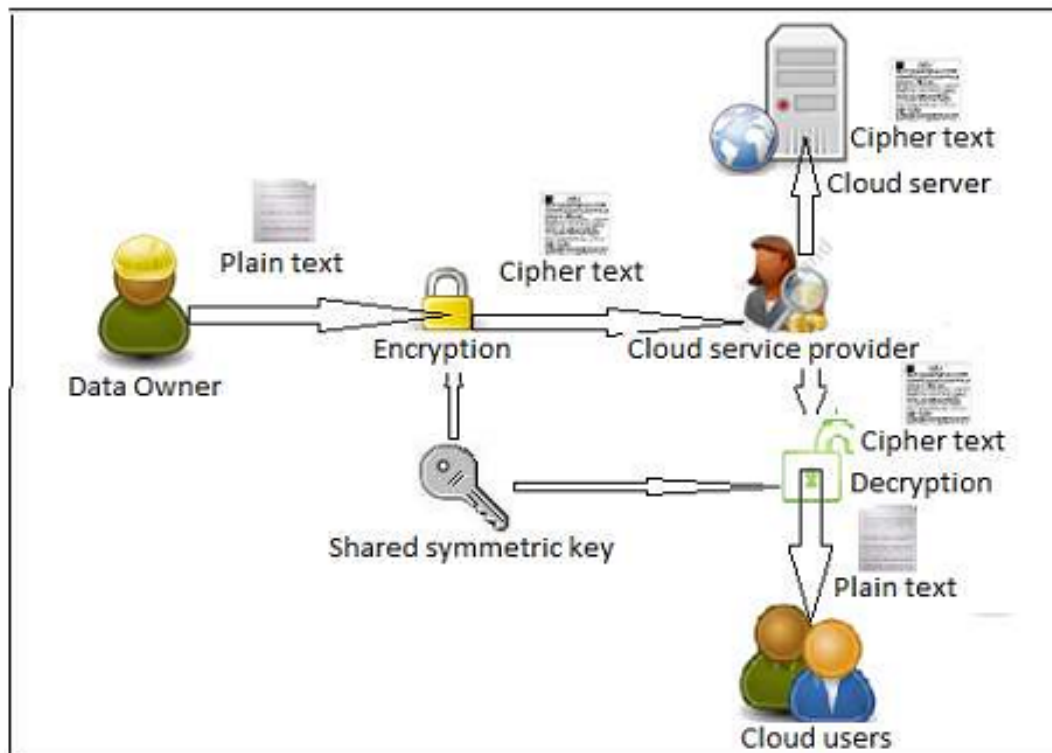


Figure 5.1: Block Diagram of Data Encryption and Decryption in Cloud System

Encryption is one of the ways to protect data at rest in a cloud server. There are four ways to encrypt the data at rest, such as; full disk level, directory level, file level and application level. The most critical part for the implementation of any of these methods is key management for data encryption and decryption. The common way to protect data in motion is to utilize encryption with authentication, which safely passes data to or from the cloud server [107].

In cloud computing, data owners increasingly outsource their sensitive data in encrypted form from local systems to public cloud for more flexibility and economic savings [108]. To protect data in transit to and from the cloud as well as data stored in the cloud, efficient data encryption and decryption algorithms are used for security. The block diagram of symmetric key encryption and decryption data storage as shown in Figure 5.1. It involves the use of a single secret key for

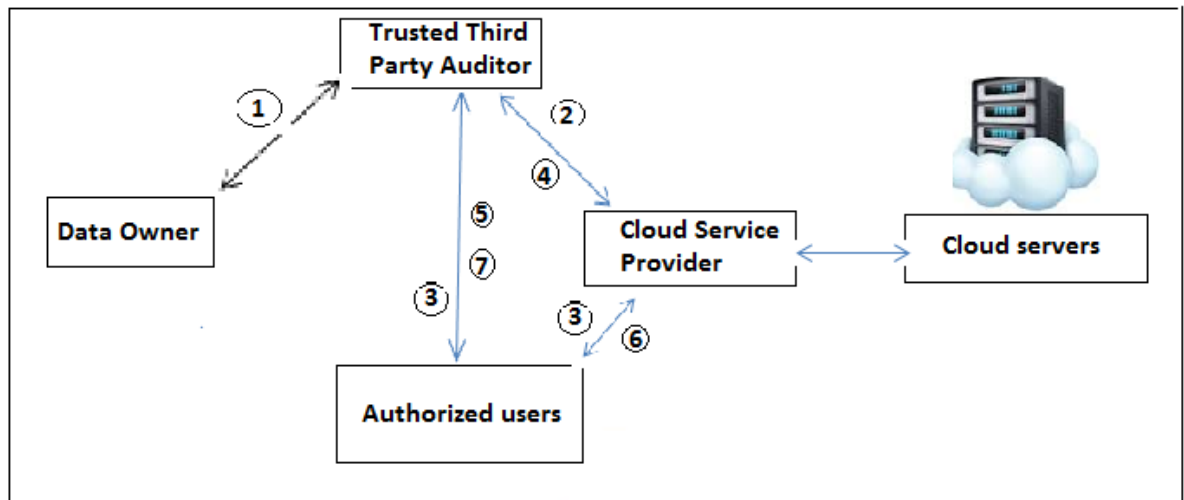


Figure 5.2: Cloud Data Storage System Model

both encryption and decryption. Data owner split the file into smaller blocks and encrypts all the blocks using symmetric secret key before sending into the cloud service provider. Then the cloud service provider stores all the encrypted blocks of the source file in a cloud server [109].

5.2.1 System Model and Setup

The cloud data storage system model for secure data access sequences is explained in Figure 5.2. The following sequence numbers are represented for data storage and access operations in cloud server.

1. The data owner splits the source file into blocks of 128 characters and encrypt all the blocks using efficient encryption algorithm and prepare the Block Status Table(BST) for encrypted blocks, then send the encrypted file, key, BST to the Trusted Third Party(TTP) auditor.
2. The TTP calculate the combined hash values for BST(TH) and encrypted file (FH), then send only encrypted file and BST to the cloud server for storage.

3. The authorized user sends the data access request to both TTP and cloud server.
4. TTP verifies the authorized user, if the user is verified then, it sends the authorization signal to the cloud server.
5. TTP send the hash values of BST (TH) and encrypted file (FH) to the requested user.
6. Cloud server sends the BST and encrypted file to the user.
7. User calculates the hash values of BST and encrypted file received from the cloud server then verifies with hash values received from the TTP. If both values are verified then the user gets a data decryption key and decrypt the data blocks.

Block Status Table(BST): The Block Status Table(BST) is a small data structure used to access the outsourced encrypted file from the cloud service provider. It consists of two column such as SN_j and BN_j , where SN_j is the sequence number of physical storage of data block j in the file and BN_j is the data block number. Initially the data owner stores table entries as $SN_j = BN_j = j$. For insertion of data blocks, the BST is implemented using a linked list. The structure of BST for insertion of data blocks as shown in Table 5.1.

Sequence number	Block number
1	1
2	2
3	3
4	4

Table 5.1: Structure of Block Status Table

Notations: The various symbols are used in this paper for the encryption and decryption algorithms is listed in the Table 5.2.

Table 5.2: Notations

Symbol	Meaning
τ_b	File Chunk Size /Block size of block b
χ	Encryption Key
F	Data Owner's file targeted for Encryption
b	File chunk/block.
E_m	Encoding Map for every Character
b_c	Binary Equivalent of Chanter c
\overline{CA}	Circular Vector of Characters.
C_{ch}	Cipher Text for a character for ch
φ_F	Size of a file F

5.2.2 Definitions

The definition of terminology used in this scheme as follows;

File Chunk Size: The security is provided at the block level. The file is divided into blocks and confidentiality is ensured on every block and finally on file. The block size is fixed for the experimental purpose.

Data Owner File: The file or the content that data owner is looking for confidentiality. And the file is a set of blocks and file size depends on the block size and defined as below in equation.

$$F = \{b_1, b_2, b_3, \dots, b_n\} \quad (5.1)$$

Similarly, every block and an encryption key is a set of characters as defined below,

$$b = \{c_1, c_2, c_3, \dots, |b|\} \quad (5.2)$$

$$\chi = \{k_1, k_2, k_3, \dots, |\chi|\} \quad (5.3)$$

The file size is defined as the summation of its component block size.

$$\varphi_F = \sum_{i=0}^{|F|} |\tau_{b_i}| \quad (5.4)$$

Encoding Map (Em): The encoding Map/Encoding table is a map between a character to every other random character in ASCII range. The ASCII value of a character is split into digits and the characters from every digit position is summed up in ASCII range to a find new character as follows, Let rc be a random character for c and ac is a set of digits forming an ASCII value of Character c ,

$$ac = \{d_i | \forall d_i \in Z+, 0 \leq i \leq |ac|\} \quad (5.5)$$

The random character ASCII value is defined as;

$$arc = \sum_{i=0}^{|F|} b_i \% 256 \quad (5.6)$$

Circular Array(CA): The circular array is used in both encryption and decryption process. The circular array is used for shift operations on both characters and on a key character. The binary equivalent of a character is stored on the array and hence the values are either 0 or 1. The circular array has the value obtained as a result of the signed right shift operation. The shift operation is performed to disguise the information by changing its bits position and it is defined by the following equation.

$$CA = \{v_i | 0 \leq i \leq |CA|, v_i \in \{0, 1\}\} \quad (5.7)$$

Key Chooser(KC): The key chooser is a vital component which defines the criteria for selecting a key character for disguising the block character of a file. The key

character is selected in such a way that, if 1st chunk character is selected for encryption then the first character of the key is considered for encryption, if a selected block character comes outside the range of key size, modulus of block character position to key size is performed to fetch a key. Two key characters are selected for every block character if i^{th} character of a block i.e. c_i is chosen for encryption then its corresponding key character at position i is selected as follows.

$$\chi_i = \chi_{(i \% |\chi|)} \quad (5.8)$$

Similarly, second key character χ_j is selected from the third position away from χ_i as;

$$\chi_j = \chi_{((i+2) \% |\chi|)} \quad (5.9)$$

CA Inverter (CAI): The CA Inverter inverts/complements the circular array for a high degree of security. The criteria on which the complements happens based on the resultant number obtained after processing the adjacent key characters. The ASCII values of adjacent key characters are added if the resultant is even then CA is inverted. Let CA be the circular array having the binary equivalent of block character c_i and let χ_i be the chosen key character, then Inverted CA is defined as below

$$ICA = \neg CA, \text{ if } \chi_i + \chi_{i-1} \% 2 = 0 \quad (5.10)$$

CA Shifter (CAS): The CA shifter shifts/rotates the circular array. The stepper movement for the circular array is based on the summation of two key characters. If the summation is a factor of 5 then the circular array is moved by 2 else it is moved by the remainder obtained from the division of summation by 5 as below,

$$SCA = CA \gg (2 | (\chi_i + \chi_j) \% 5) \quad (5.11)$$

Where χ_i, χ_j is the chosen key characters, SCA is the shifted Circular for i^{th} block character c_i and CA is a circular array storing a binary value of c_i .

Encryption Engine/Cipher Engine: Encryption Engine is a black box which takes block character to produce cipher character. The Encryption Engine is composed of above three components (*Key Chooser, CA Inverter, and CA Shifter*) in that order as below, Let c_i be the i^{th} block character and CE be the cipher engine

$$CE = KC \cup CAI \cup CAS \quad (5.12)$$

Decryption Engine/Decipher Engine(DE): The Decryption Engine is composed of same components as Encryption Engine but these components are applied in reverse order as below in equation 5.13.

$$DE = CAS \cup CAI \cup KC \quad (5.13)$$

5.2.3 Algorithms

The data owner encrypts the file before sending it to the Cloud Service Provider (CSP). The encryption algorithm has several steps and composed of key Chooser, Circular Array Inverter and Circular Array shifter. The encryption algorithm will disguise the information at highest factor by applying series of rotations on every block character and the key is rotated for every character. From this, it is ensured that the same key is not used for encrypting every character and hence this algorithm is called as Key Motor encryption algorithm.

In the encryption process, the file is divided into blocks and confidentiality is em-

Alg. 1 Key Motor Encryption Algorithm

Input: F : File for Encryption, χ :Encryption Key

Output: CF : Cipher text file CF

Algorithm steps :

- 1 Split the characters of the file/string into blocks.
- 2 Get the mapped value from E_m .
- 3 Get the binary equivalent of the Current Character.
- 4 Remove the first character from the binary value and store it into First Character and consider rest of binary value for shift operations (this is done to avoid having the case the resultant value 00001 or 01111 after shift operations, this would result in forgetting a bit as 0001 can also represent as 1)
- 5 Store binary value into a CA for bit operations.
- 6 The key chooser component selects a key character from i^{th} position, such a way that if chunk character is selected for encryption then select the first character of the key, so $i = \text{selected chunk character} > \text{size of key } |\chi|$ get the modulus of chunk character position.
- 7 Add the selected i^{th} key character integer value and its previous (i^{th}) character integer value instance. If 1^{st} key character is selected then the previous character would be the 16^{th} character (k stored into a circular array or double way linked list for this operations)
- 8 The CA Inverter component complements Circular array (ICA) if the summed result is as per the equation 5.10
- 9 Add the selected i^{th} key character and $(i + 2)^{th}$ key character.
- 10 The CA shifter shifts the Circular to find SCA as per the equation 5.11.
- 11 Add the stripped off first Character to the resultant binary value of Circular Array obtained the previous step.
- 12 Get the character equivalent of the binary value which is the cipher character.
- 13 Repeat steps 2 through 12 for all the chunks with different Key (by shifting the key character using Circular Array a double way linked list).

phasized on every character level of a block. The binary equivalent of block character is stored in a circular array and number of moves the circular array is rotated is decided

by the CA Shifter. Since stepper movement of CA is different for the different character it's hard/impossible to determine the actual value of CA as explained in the algorithm 1.

The key portion of the algorithm is the CA inverter and CA shifter which is performed on every block character and finally an entire block. If File has N blocks and if every block has N characters then CAI and CAS is performed by N^2 operations. Therefore, this algorithm has complexity of $O(N^2)$.

The decryption process happens exactly opposite to encryption which finds a block character from cipher text as per equation 5.13. The decryption process in the algorithm 2 suggests that CA shifter is performed first then Key chooser component is used to select two keys and they are added before inverting CA. Since CA already contains complemented value and complement of CA now yields original encoded value. The Encoding Map Em is searched to get its original character. The Algorithm has same complexity as encryption algorithm.

When the user wants to access data from cloud server, the data verification procedure is explained in the algorithm 3.

5.3 Data Audit using Protocol

In this section we explain the design method and algorithms of identity based and linear authentication based public data verification schemes.

Alg. 2 Key Motor Decryption Algorithm

Input: CF : Cipher text file for Decryption, χ : Decryption Key

Output: Plain text of a file F

Algorithm steps :

- 1 Split the characters of the file/string into blocks.
 - 2 Get the binary equivalent of the current cipher character.
 - 3 Remove the first character from the binary value and store it into a first Character variable and consider the rest of binary value for shift operations.
 - 4 Store binary value into a \overline{CA} .
 - 5 Select a key character from i^{th} position, such a way that if 1^{st} chunk character is selected for encryption then select the first character of the key, so $i = 1$.
if selected chunk character $>$ size of $(|\chi|)$ then, get the modulus of chunk character position.
 - 6 Add the stripped off first Character to the resultant binary value of \overline{CA} after bit operations.
 - 7 Add the selected χ_i and χ_{i+2} key character.
 - 8 The CA shifter shifts the Circular to find SCA as per the equation 5.11.
 - 9 Add the selected i^{th} key character int value and its previous $(i - 1)^{th}$ character integer value, for instance if 1^{st} key character is selected then the previous character would be the 16^{th} character (key stored into circular array or double way linked list for this operations)
 - 10 Get the character equivalent of the binary value
 - 11 Get the mapped value from E_m , which is the decrypted value of the character.
-

5.3.1 System model

The system model considered in the proposed work as shown in Figure 5.3. It consists of five components such as; *key generator*, *cloud servers*, *verifier*, *cloud users*, and *combiner*.

Key generator: It is an entity, which receives the identity of the user(ID) and generates the secret key(sk_{id}) for the user(ID) using a computational Diffie-Hellman (CDH)

Alg. 3 Data Access Procedure

Algorithm steps :

- 1 An authorized user sends a request to both the CSP and TTP auditor.
 - 2 CSP sends the encrypted file and BST to the requested user.
 - 3 TTP sends the FH_{ttp} , TH_{ttp} and key to the user.
 - 4 user computes the TH_{user} using TH_{csp} and FH_{user} using data blocks. Then compared with stored TH_{ttp} and FH_{ttp} respectively for integrity check.
 - 5 If both BST and file computed hash values are matches, then user decrypt the file using shared secret key
-

method.

Cloud users: It is an individual user or an organization which outsource the data to multi-cloud servers for maintenance and management of shared data.

Verifier: It is an entity, either the data user or third party auditor to check the correctness of outsourced data using the Decisional Diffi-Hellman method.

Cloud servers: It is a set of servers, which managed by the cloud service provider to provide the storage service, which has massive compute and storage facility.

combiner: It is an independent and trusted entity. Which distribute the requests to the servers and aggregate the responses from the servers.

Notations: The various symbols are used in this scheme is listed in Table 5.3 as follows.

5.3.2 Basic Auditing Scheme

The basic data integrity verification scheme consists of three stages such as; key generation, meta data generation and data audit. The detail of these stages are shown

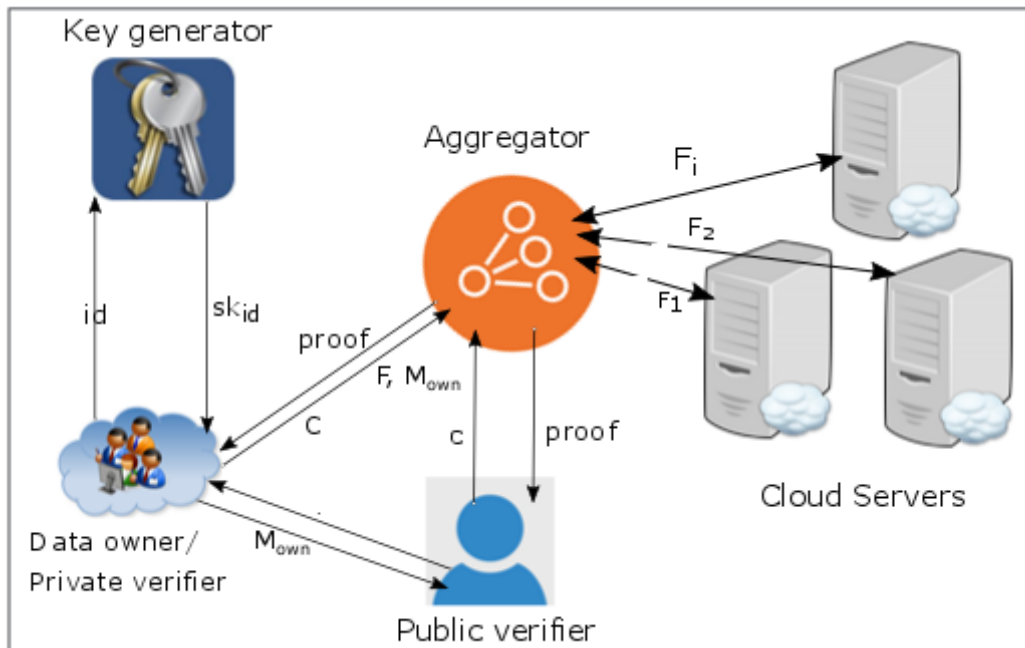


Figure 5.3: Data Auditing System Model

in Figure 5.4.

Key Generation: It is an entity, which takes the input as security parameter(k) and the data owner identifier (id) and generates the public parameters, secret key, public key, and owners private key(pk_id).

Meta-data Generation: To generate the meta-data for a given file (F_i) of the owner (id), the meta-data generator takes input as owners private key (pk_id) and the file (F_i) as input and generates the signature of file blocks interns of *block – tag* pair.

Data audit: The data auditing process consists of five steps of the request, response, and verification among the verifier, combiner and cloud servers.

1. Verifier sends the challenge request to the combiner for verification of the selected number of data blocks stored on cloud servers.
2. The combiner searches the requested data blocks meta-data from the meta-data table and then distribute the request to the corresponding cloud server.

Table 5.3: Notations

Symbol	Meaning
F	erasure encoded file $F = \{F_{ij}\}$
t	file tag
n	number of data blocks
s	number of sectors per block
σ	block authenticator
M_{owner}, M_{agg}	File metadata for owner and combiner
$H(), h()$	hash functions
Φ	set of authenticators
r and x	random number
c	challenge message
CS_i	set of cloud servers
p	prime number
msk_{id}	master secret key
r	random number
V	challenge message
$MAC_{key}()$	message authentication code using key
p	prime number
(spk, ssk)	secret public and private key
K	number of data owner

3. After receiving the responses from the cloud servers, the combiner combines all the responses.
4. Combiner sends the final combined response to the verifier.
5. The verifier verifies the response message using bilinear map operation. If the response is valid, verifier confirms data blocks are not modified, otherwise he declares data blocks are modified.

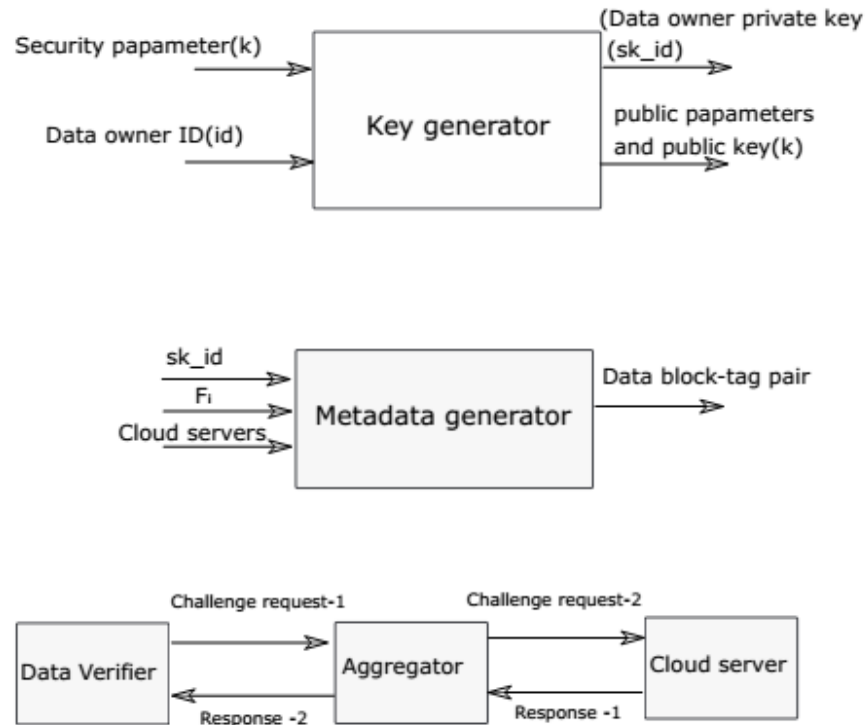


Figure 5.4: Basic Data Auditing System

5.3.3 Identity-based Data Auditing Method

The remote data verification using identity-based method consists of initial file setup and data audit phases. The detailed design of these phases is explained as follows.

A. Initial File setup phase: In this phase, the data owner generates the private and public keys for data verification, and prepares the meta-data of the file before sends to the cloud server.

Key Generator: The key generator selects two random positive integer numbers r and x and calculate $A = g^x$ and $B = g^r$ where g is the generator i.e $g < \text{group } G_1$, keeps x has a secret key and $\{g, A\}$ as public parameters.

For the given data owner (id) key generator calculate the signature using $\sigma = r + x(H(id, B))\%q$ and sends the private key $sk_{id} = (\sigma \text{ and } B)$ to the data owner. Then the data owner verifies the private key using the following Equation. The detailed key generation algorithm is explained in the Algorithm 4

$$g^\sigma \stackrel{?}{=} BA^{H(id,B)}$$

$$\begin{aligned} LHS &= g^\sigma \\ &= g^{r+xH(id,B)} \\ &= g^r \cdot g^{xH(id,B)} \\ &= BA^{H(id,B)} \\ &= RHS \end{aligned}$$

For example

$$g = 3, q = 11, r = 3, x = 4$$

$$A = g^x = 3^4$$

$$B = g^r = 3^3$$

$$\sigma = r + x(H(id, B))\text{mod}q$$

$$= 3 + 4.H(id.B)\text{mod}11$$

$$g^\sigma \stackrel{?}{=} BA^{H(id,B)}$$

$$3^{3+4.H(id.3^3)\text{mod}11} = 3^3 \cdot 3^{4.H(id.3^3)\text{mod}11}$$

$$= 3^{3+4.H(id.3^3)\text{mod}11}$$

Meta-data generation: It is an individual user or an organization which outsource the data to multi-cloud servers for maintenance and management of shared data. The data owner with the valid private key (sk_{id}) prepares the meta-data for the file F and stores the meta-data and the corresponding file on cloud server CS .

Let consider a encrypted file F is split in to n blocks $F = \{F_1, F_2, \dots, F_n\}$ and each block further split into s sectors i.e; $F_i = \{F_{i1}, F_{i2}, \dots, F_{is}\}$.

$$F = \begin{bmatrix} F_{11} & \cdots & F_{1s} \\ F_{21} & \cdots & F_{2s} \\ \vdots & \ddots & \vdots \\ F_{n1} & \cdots & F_{ns} \end{bmatrix}$$

The Data owner calculates the hash value for each sector using SHA-512 hash function i.e. $F'_{ij} = h1(F_{ij})$ and prepares the meta-data M_i for the file block F_i using equation 5.14. The detailed procedure for meta-data generation is defined in the algorithm 4.

$$F' = \begin{bmatrix} F'_{11} & \cdots & F'_{1s} \\ F'_{21} & \cdots & F'_{2s} \\ \vdots & \ddots & \vdots \\ F'_{n1} & \cdots & F'_{ns} \end{bmatrix}$$

$$M_i = (h(CS_{l_i}, i, name_i) \cdot \prod_{j=1}^s u_j^{F'_{ij}})^{\sigma} \quad (5.14)$$

Then, the owner sends the file blocks and meta data $\{F_i$ and $M_i\}$ to the cloud server CS_{l_i} . The data owner prepares the meta-data table $T_{owner} = (i, u, CS_{l_i}, name_i)$ and stores

in the combiner table $T_{combiner}$. Where i is the identifier of each block, j is the sector number in the data block and u_j is the random number *i.e.* $u = (u_1, u_2, \dots, u_s)$

Combiner: It is an independent and trusted entity, which distribute the requests to the servers and aggregate the responses from the servers.

Alg. 4 O-RDAP-Algorithm: Initial File Setup

————— **Data owner**:Generate key—————

input :user identity (id)

output: Master secrete key (x), Public parameters (p, q, g, A, H)

- 1: Select a random number x, r from a set of positive integer numbers Z_q^*
- 2: compute A, B and σ
 $A = g^x$ and $B = g^r$ $\sigma = r + x(H(id, B)) \bmod q$
- 3: keeps x has the master secret key.
- 4: sends private key $sk_{id} = (B, \sigma)$ and public parameters to the data owner.
- 5: Verify the owner's identity id by solving the DDH problem; $g^\sigma \stackrel{?}{=} BA^{H(id, B)}$
- 6: If the equation is verified, then accept the user (id) private key sk_{id} , otherwise reject it.

————— **Data owner**:Generate meta-data—————

input :File (F), sk_{id}

output: Meta data M_i for the file block F_i

- 1: Data owner split the file F in to n blocks $\{F_i\}$, and each encrypted data block in to s sectors *i.e.*, $\{F'_{ij}\}$ where $i \leq n$, and $j \leq s$
 - 2: Data owner selects s random number vector $\{u_i\}$ where $j \leq s$
 - 3: calculate the hash values for each encrypted file block *i.e.*, $F_{ij} = h(F'_{ij})$
 - 4: calculate the meta-data for the i^{th} file block *i.e.*, $M_i = (h(CS_{l_i}, i, name_i). \prod_{j=1}^s u_{ij}^{F_{ij}})^\sigma$
 - 5: Data owner adds $\phi_i = (i, u, CS_{l_i}, name_i)$ to the M_i table and share this table to verifier.
 - 6: Data owner sends M_i to combiner, then the combiner stores in his meta-data table, M_{agg}) and stores file blocks in cloud server CS_{l_i}
-

B. Data Audit Phase:

The data blocks stored on cloud servers CS are audit the by TPA. The data audit is a sequence of request and response message among auditor, combiner and cloud servers with the help of public parameters and meta-data table($T_{combiner}$).

The auditor sends a request for selected number of blocks c to the combiner. The combiner identifies the corresponding cloud server using the meta-data table $T_{combiner}$ and further sends a request to the corresponding cloud server CS_{I_i} . After receiving the request from the combiner, cloud server prepares the response message and sends to the combiner. The combiner combines all the responses and sends the aggregated response to the auditor. The detailed data auditing algorithms is explained in the Algorithm 5.

Alg. 5 Data_Audit(c)

input : Challenge message c data blocks

output: meta data for the file block F_i

- 1: The private or public verifier sends the challenge message (c) to the combiner
- 2: combiner prepares the index set(I_i) for the corresponding c request blocks using *block – tag* pairs stored in cloud server.
- 3: combiner sends the index set(I_i) to the cloud servers.
- 4: for each cloud server v_l calculates $(M^{(i)}, F_j^{(i)})$ and send to the combiner.

$$M^{(i)} = \prod_{v_l \in I_i} \{M_{v_l, j}\}$$

$$F_j^{(i)} = \sum_{v_l \in I_i} \{F_{v_l, j}\}$$

- 5: Combiner prepares the aggregated message (M and F') then sends to the verifier.

$$M = \prod_{cs_i} M^{(i)} \text{ and } F'_l = \sum_{cs_i} F_l'^{(i)}$$

- 6: Verifier solves the following DDH problem and returns the status of the file block.

$$e(M, g) \stackrel{?}{=} e(\prod_{i=1}^c h_i \prod_{j=1}^s u_j, BA^{H(id, B)})$$

- 7: If the above equation holds then it response a success, otherwise it response failure message.
-

The proposed data auditing scheme can be apply for private and public data verification. The block diagram for private and public data verification as shown in Figure 5.5 and Figure 5.6 respectively.

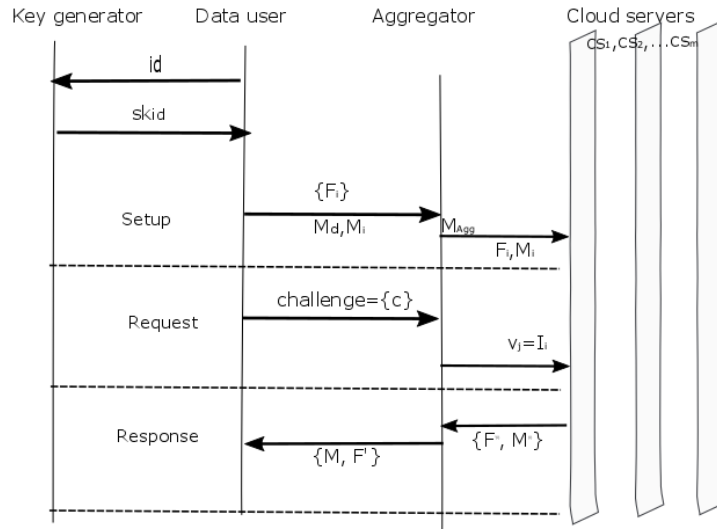


Figure 5.5: Batch Auditing setup and Private Verification

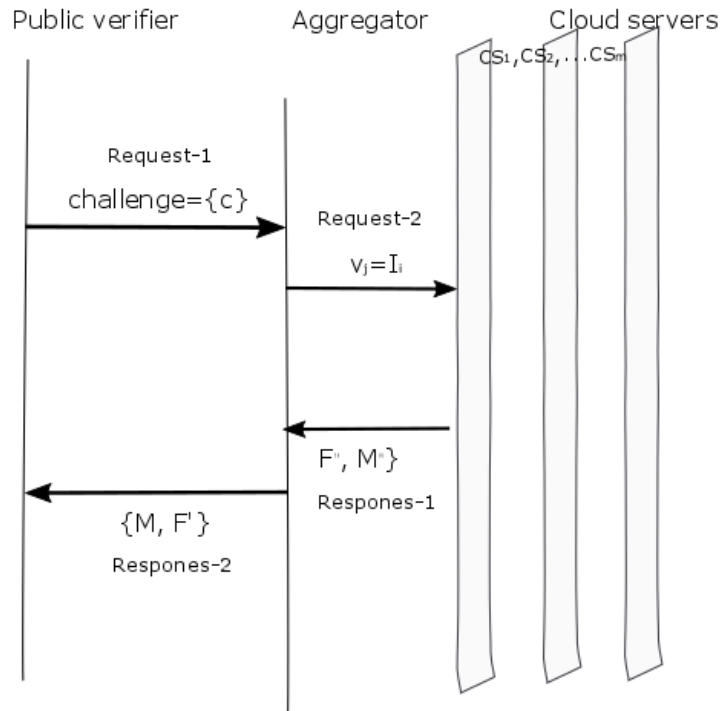


Figure 5.6: Data Auditing using Public Verification

The proof for data audit response verification as follows.

$$e(M, g) \stackrel{?}{=} e(\prod_{i=1}^c h_i^{F_i} \prod_{j=1}^s u_j, BA^{H(id, B)}) \quad (5.15)$$

$$\begin{aligned}
 LHS &= e(M, g) \\
 &= e\left(\prod_{cs_i} M^{(i)}, g\right) \\
 &= e\left(\prod_{cs_i} \prod_{v_l \in M_i} M_{v_l}, g\right) \\
 &= e\left(\prod_{cs_i} \prod_{v_l \in M_i} h_l \prod_{j=1}^s u_j^{F_{v_l j}}, g^\sigma\right) \\
 &= e\left(\prod_{i=1}^c h_i \prod_{j=1}^s u_j^{F_{v_l j}}, BA^{H(id, B)}\right) \\
 &= RHS
 \end{aligned}$$

5.3.4 Linear Authenticator based Data Auditing

In this section, we proposed a public auditing for single data owners file by utilizing linear authenticator scheme for proof of retrievability of outsourced data file. The design steps for this method such as key generation, file setup and public auditing are described as follows.

A. Initial File Setup:

Key generation: To generate the secrete and public parameters for the file, data owner follows the following steps.

- Select the random signing key pair (ssk, spk) to generate the secrete and public parameters
- Select the random integer number x and u from the group G_1 , then compute v such that $v = g^x$
- Secrete parameter $sk = (x, ssk)$ and public parameter $pk = (spk, v, g, u, e(u, v))$

File storage on the cloud: The initial file setup for file outsourcing at cloud server the data owner prepares the meta data using the following steps

1. Prepare the meta-data of the file F

- Split the file F in to n blocks i.e $F_i = \{b_1, b_2, \dots, b_n\}$ of equal size
- Compute the authenticator (σ) for each block b_i as follows;
 - Select the unique identifier for the file F and append block index to it i.e $W_i = id||i$
 - $\sigma_i = (H(W_i).u^{b_i})^x$
- Prepare the authenticator set $\Phi = \{\sigma_i\}$ where $i = 1$ to n
- Generate the file tag t for the integrity of the file identifiers $t = id||Signature_{sk}(id)$

2. Send the file F and meta-data $\{\Phi, t\}$ to the cloud server.

3. Remove file and meta-data from the local storage.

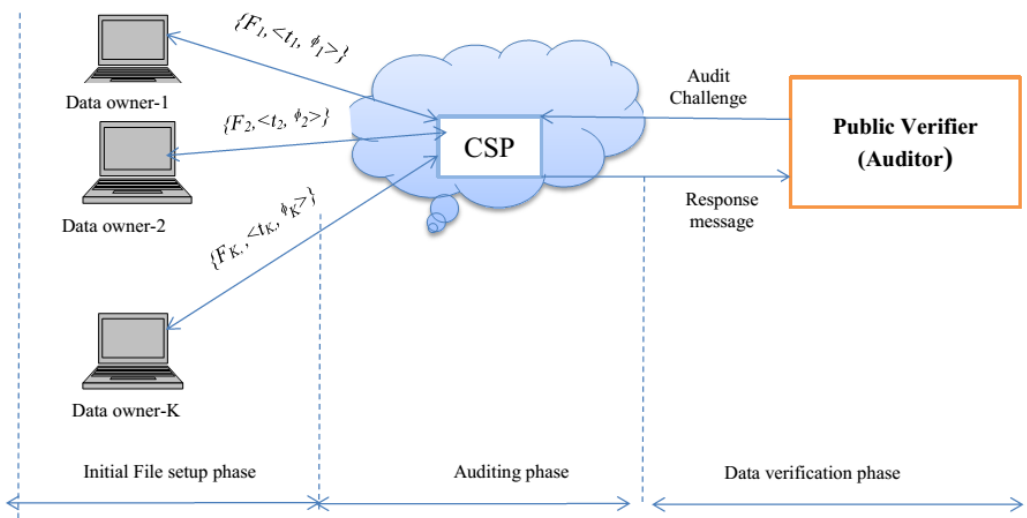


Figure 5.7: Batch Auditing Model

Data Audit: To verify the correctness of the owner's data file stored on untrusted cloud server using public verifier without retrieving the original file is explained as follows.

1. Auditor retrieve the file tag t from the server and verifies the $signature(id)$ using public key spk
2. Generate a challenge message $Chal = i, V_i \in I$ Where $I = \{s_1, s_2, \dots, s_c\}$ and $i \leq Sc < [1, n]$ and V_i is the random number.
3. Sends the challenge message to the server.
4. Server prepares the response message $\{\mu, \sigma, R\}$ using following steps
 - Select the random number r and compute $R = e(u, v)^r \in G_T$
 - Compute the linear combination of sampled data blocks; $\mu' = \Sigma(V_i b_i)$
 - Blind μ with r using ; $\mu = r + \Upsilon \mu' \text{ mod } p$ where $\Upsilon = h(R)$
 - Calculate the aggregated authenticator $\sigma = \prod_{i \in I} (\sigma_i^{V_i})$
5. Send the response message μ, σ, R to the auditor.
6. Auditor compute $\Upsilon = h(R)$ and verifies $\{\mu, \sigma$ and $R\}$ using equation (5.16).

$$R.e(\sigma^\Upsilon, g) \stackrel{?}{=} e((\prod_{i=1}^{S_c} H(w_i)^{V_i})^\Upsilon . u^\mu, v) \quad (5.16)$$

B. Batch Auditing

Auditing multiple users data file using single auditing method is more expensive in

terms of computation and communication overheads for the server as well as auditor. To overcome these overheads, an auditor groups the multiple files in the batches and submit each batch to a cloud server for audit as shown in Figure 5.7. The detailed procedure for this auditing is explained in algorithm 6.

Key Generation: Let k is the number of data owners in the system and each owner has data file $F_k = \{b_{k_1}, b_{k_2}, \dots, b_{k_n}\}$ to store on a cloud server and each file has n number of data blocks. For each, data owner choose the secret private and public keys (ssk , psk) and a random number x_k . The secret key is (x_k, ssk) and the public parameter is $(spk_k, v_k, g, uk, e(u_k, v_k))$, where $v_k = g^{x_k}$

Meta-data generation: The meta-data for the file F_k generation steps as follows;

1. generate the file tag t_k for the k^{th} user data file $t_k = id_k || Signature_{ssk_k}(id_k)$, where id_k is the file identifier

2. calculate the authenticator (σ_k, i) for each data block of the file F_k .

$$\begin{aligned} \sigma_{k,i} &= (H(id_k || i) \cdot u_k^{b_{k,i}})^{x_k} \\ &= (H(W_{k,i}) \cdot u_k^{b_{k,i}})^{x_k}, \text{ where } i = 1 \text{ to } n \end{aligned}$$

$$\phi = \{\sigma_{k,1}, \sigma_{k,2}, \dots\}$$

3. send $\{F_k, t_k, \phi_k\}$ to the cloud server
4. delete F_k, t_k, ϕ_k from the local storage.

C. Auditing for multiuser

The data auditing of multiuser multiple files on a cloud server is as follows;

Alg. 6 Batch Auditing

Data owner:

- 1 Split the file in to n equal sized data blocks $F_k = b_{ij}$
- 2 Generate the secret and public key parameters;
 Secret key $= (x_k, ssk_k)$ and
 public key $= (spk_k, v_k, g, u_k, e(u_k, v_k))$
- 3 Generate the file tag t_k ;

$$t_k = id_k || sig_{ssk_k}(id_k)$$
- 4 Compute the authenticator

$$\sigma_{k_i} = (H(id_k|i) \cdot u_k^{b_{ki}})^{x_k}$$
- 5 Store F_k, σ_{k_i}, t_k at server side

Auditor:

- 6 Retrieve and verify the file tag t_k for the auditing k^{th} user file.
- 7 send the request $\{i, V_i\}$ to the server
- 8 response for this request from the server, auditor verify the storage correctness equation.

Server:

After receiving a request from the auditor, for each request server prepares the response message as follows;

- 9 Compute $\{\mu_k, \sigma_k, R_k\}$ using the following equations.

$$\mu_k = \sum V_i b_{ki}$$

$$\sigma_k = \prod \sigma_{k_i}^{V_i}$$

$$R_k = e(u_k, v_k)^{r_k}$$
- 10 Compute $R = R_1 \cdot R_2 \dots R_K$

$$L = v_{k1} || v_{k2} || \dots || v_{kK}$$

$$\Gamma_k = h(R || v_k || L)$$
- 11 Compute $\mu_k = r_k + \Gamma_k \mu_k \text{ mod } p$
- 12 Send the response message $\{\mu_k, \sigma_k, R\}$ to the auditor.

-
1. Auditor retrieves all file tags t_k from the cloud server and verifies all the tags t_k are matched or not.

2. If tags verification fails, the auditor will discard those tags.
3. Once the tags are verified, Auditor prepares audit challenge message for each audit file and send to server.

$$Chal_k = \{(i, V_i)\} i \in I$$

Where i is the position of the data block b_k , V is the random value, I is the subset of random elements chosen set $[1, n]$

4. After receiving challenge message from the data owners, the server prepares a response message as follows;

- pick a random variable r_k and $R_k = e(u_k, v_k)^{r_k}$
- $R' = R_1 . R_2 . \dots . R_k$
- $L = v_{k_1} || v_{k_2} || \dots || v_{k_K}$
- $\Gamma_k = h(R' || v_k || L)$

$$\mu_k = \Gamma_k \Sigma(V_i b_{ki}) + r_k \text{ mod } p$$

$$\sigma_k = \Pi(\sigma_{ki}^{V_i})$$

$$\text{response message} = \{\mu_k, \sigma_k, R'\}$$

5. the server sends the response message to the auditor
6. The auditor checks the response message validity using the following equation.

$$\Gamma_k = h(R' || v_k || L)$$

$$R'.e(\prod_{k=1}^K \sigma_k^{Y_k}, g) \stackrel{?}{=} \prod_{k=1}^K e((\prod_{i=S_1}^{S_c} H(W_k i)^{V_i})^Y . u_k^{\mu_k}, v_k) \quad (5.17)$$

The verification proof of the above equation is derived as follows

$$\begin{aligned} LHS &= R'.e(\prod_{k=1}^K \sigma_k^{Y_k}, g) \\ &= R_1.R_2 \dots R_k . \prod_{k=1}^K e(\sigma_k^{Y_k}, g) \\ &= \prod_{k=1}^K R_k e(\sigma_k^{Y_k}, g) \\ &= \prod_{k=1}^K (u_k, v_h)^{r_k} e(\sigma_k^{Y_k}, g) \\ &= \prod_{k=1}^K (u_k, v_h)^{r_k} e(\sigma_k^{Y_k}, v_k^{r_k}) \end{aligned}$$

D. Public Storage Correctness:

The storage verification of data blocks, the adversary server response for the auditor request $\{i, V_i\}$, the proof is explained as follows;

Let consider the adversary response for the auditor request $\{i, V_i\}$ from the server output is $\mu'_1, \mu'_2, \mu'_3, \dots, \mu'_s$ with aggregated authenticator σ'

The corresponding response verification at auditor has satisfies the following equation

$$e(\sigma', g) = e(\prod_i H(id||i)^{V_i} . \prod_j^s u_j^{\mu'_j}, v) \quad (5.18)$$

The actual honest response for the challenge $\{i, V_i\}$ from the server is $\{\mu_1, \mu_2, \mu_3, \dots, \mu_s\}$ with σ , where $\sigma = \prod_i \sigma_i^{V_i}$ and $\mu_j = \sum_i V_i b_{ij}$

The honest server response message verification as follows;

$$e(\sigma, g) = e(\prod_i H(id||i)^{V_i} . \prod_j^s u_j^{\mu_j}, v) \quad (5.19)$$

where $v = g^x$ is a challenger's public key.

$\Delta\mu_j = (\mu'_j - \mu_j)$ $j= 1$ to s , if anyone $\Delta\mu_j$ is non zero, which leads to the invalid response to the challenge $\{i, V_i\}$

The auditor then obtains $\{\sigma', (\mu - \mu')/(\Upsilon - \Upsilon')\}$ as a response using the following proof.

$$\begin{aligned}
 e(\sigma/\sigma', g) &= e((\prod H(w_i)^{V_i})^{(\Upsilon - \Upsilon')} . u^{(\mu - \mu')}, v) \\
 &= e((\prod H(w_i)^{V_i})^{(\Upsilon - \Upsilon')}, g^x) . e(u^{(\mu - \mu')}, g^x) \\
 \sigma/\sigma' &= (\prod H(w_i)^{V_i})^{(\Upsilon - \Upsilon')} . u^{x(\mu - \mu')} \\
 u^{x(\mu - \mu')} &= ((\sigma\sigma'^{-1})/\prod_i H(w_i)^{x_i V_i})^{(\Upsilon - \Upsilon')} \\
 u^{(\mu - \mu')} &= (\prod_i u^{x_i V_i})^{(\Upsilon - \Upsilon')} \\
 \mu - \mu' &= (\sum m_i V_i) . (\Upsilon - \Upsilon') \\
 (\sum m_i V_i) &= (\mu - \mu')/(\Upsilon - \Upsilon')
 \end{aligned}$$

5.4 Data Audit using Digital Signatures

This section presents the detailed design and algorithms for remote data checking using Elliptic curve digital signature method in the cloud. The proposed method consists of initial file setup and data audit phase.

5.4.1 Initial File Setup

Consider a cloud storage system in Figure 5.8, consists of three computing entities such as; data owner, public verifier and cloud service provider. The detailed functions of the system model as follows;

consider a data owner wants to store the file (f) on a cloud server, splits the data file into n equal blocks, $\{f_i\}$ where $i = 1$ to n , where n is the total number of blocks.

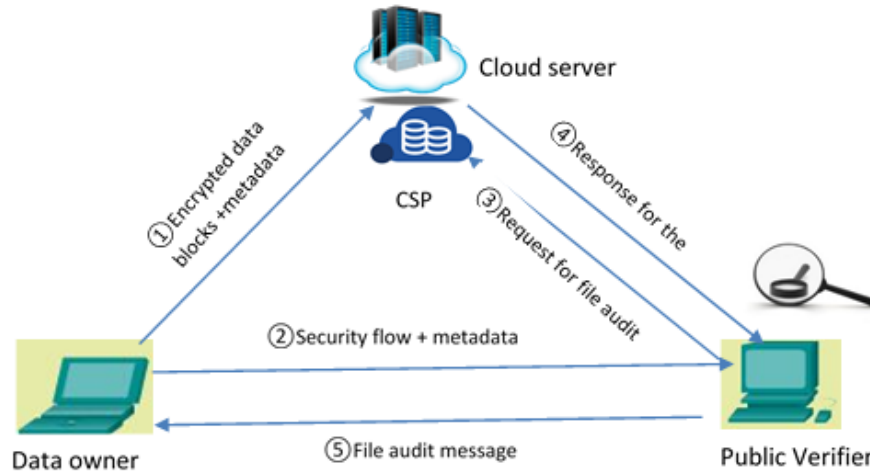


Figure 5.8: System model

The number of data blocks depends on the file size and block size. f_i denotes the i^{th} data block of the file f . For example, if the file contains 100MB of data and block size is 10KB of data, then the total number of data blocks is 1000.

The data owner encrypts all the data blocks using encryption algorithm with key rotation technique i.e, $F_i = Encry(f_i, key_i)$, where key_i is the i^{th} block encryption key. The key generation process using rotation as shown in the Algorithm 7.

Alg. 7 Key generation using key rotation

Algorithm Key_Rotation($M_k, i, keylength$)

Input: M_k =Master key, i =block number, $keylength$ = size of the key

Output: Data block F_i encryption key key_i

- 1 Convert the length of the master key to maximum length of the key i.e
 $k_1 = M_k \& 2^{(keylength-1)}$
 - 2 Rotate k_1 towards right i number of bits; $k_2 = k_1 \gg (i \% keylength)$
 - 3 Rotate M_k towards left $(keylength - i)$ bit $k_3 = M_k \ll (keylength - (i \% keylength))$
 - 4 Generate the block encryption key by combine k_2 and k_3 values $key_i = k_2 | k_3$
-

File tag generation: The data owner generates the signature of the original file F i.e., $t = dataowner_{id} | file_{id}$, which is used to verify the filename on cloud server

before auditing data blocks.

Data block F_i Signature generation: Data owner generates two signatures $(\sigma_{i1}, \sigma_{i2})$ for each data block F_i using ECDSA digital signature algorithm. The signature generation steps as follows;

1. *Initialization:* Initialize the elliptic curve $y^2 = x^3 + ax + b$ over a finite field $F_p(a, b)$, and generate the set of points on the curve. Where a and b are the coefficients, x and y are the coordinates of the curve point and p is the finite field size.
2. *Generate public and private parameters:* select a random number from pr_{key} from $[1, q]$, which acts as private key. compute the public parameter $pb_{key} = pr_{key} \cdot g(x, y)$, where $g(x, y)$ is the curve point used as a generator. Then return the private pr_{key} and public parameters pb_{key} .
3. *Data block signature calculation:* Data owner generates the file data block F_i signatures $(\sigma_{i1}, \sigma_{i2})$ using the private key of the data block.
 - Find the hash value of the data block F_i using secured hash function i.e., $z = h(F_i)$.
 - Initialize $\sigma_{i1} = \sigma_{i2} = 0$.
 - Repeat the following steps until $\sigma_{i1} \neq 0$ and $\sigma_{i2} \neq 0$
 - select a random number k between 1 and q .
 - find the third point on the curve $P(x, y)k * g(x, y)$
 - calculate $\sigma_{i1} = x \%_o q$ and $\sigma_{i2} = ((z + \sigma_{i1} * pr_{key}) * k^{-1}) \%_o q$

– return $(\sigma_{i1}, \sigma_{i2})$

Alg. 8 Data owner: Initial Setup

- 1 Split the data file F in to n different blocks of block size $F = F_i$ where $i = 1$ to n
 - 2 Generate the block encryption key using key rotation algorithm
 - 3 Encrypt each block using data encryption algorithm
 - 4 Generate the tag for all the encrypted data blocks using SHA512 or ECDSA algorithm
 - ECDSA Signature generation**
 - i. initialize the Elliptic curve parameters $(p, a, b, g(x, y), q)$
 - ii. **Key generation:**
 - a. Select the random number pr_{key} from $[1, q]$
 - b. calculate the public key $pb_{key} = pr_{key} * g(x, y)$
 - c. return $(pr_{key}$ and $pb_{key}(x, y))$
 - iii. **Signature generation** (pr_{key}, F_i) :
 - a. Find the hash value of file block; $z = H(F_i)$
 - b. Initialize the signatures; $\sigma_{i1} = \sigma_{i2} = 0$
 - c. **While** $(\sigma_{i1} \neq 0 \vee \sigma_{i2} \neq 0)$ **do**
 - i. $k = random(q)$
 - ii. $x, y = (k * g(x, y))$
 - iii. $\sigma_{i1} = x \% q, \sigma_{i2} = ((z + r * Pr_{key}) * k^{-1}) \% q$
 - endwhile**
 - d. Return σ_{i1}, σ_{i2}
 - 5 Prepare the meta-data for the entire file F
 - 6 Store the encrypted file F_i blocks and meta-data $(\sigma_{i1}, \sigma_{i2})$ on cloud server.
 - 7 Send the σ_{i1} to auditor and store as T_{tpa}
-

The data owner stores all the signatures of the data blocks F_i in the meta-data table (T_{owner}). Then sends the File blocks and meta-data along with file signature t to a cloud server (CSP) and stores the meta-data in his table (T_{csp}). The data owner also sends the file tag and σ_{i1} to the third party auditor (TPA) for public verification and it is stored in his meta-data table (T_{tpa}). The detailed algorithm for initial file setup as

shown in the algorithm 8

Meta-data generation: For initial file setup, the structure of meta-data table as shown in Figure 5.9. The meta-data table consists of six fields such as;

OWNER_ID: which is a unique identifier that identifies the owner of the file.

File – name : which is the name of the file stored on a cloud server.

file – Id: a unique identifier to identifies the file,

tag : it is signature of the entire file.

block – Id: which is the different data block meta-data, which contains the signature of the contents and block identifier,

signType : which holds the type of the algorithm (ECDSA)used to generate the signature.

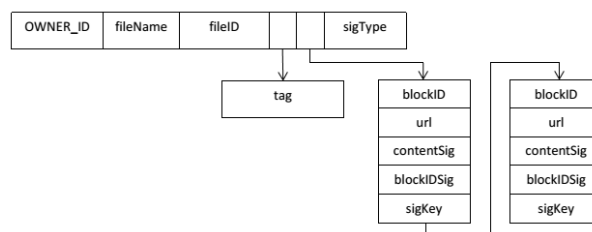


Figure 5.9: Metadata Representation

5.4.2 Data verification

After the encrypted file and meta-data of the file outsourced to a cloud server, the TPA can check the integrity of the data blocks periodically for the favor of data owner.

The data verification is a sequence of request and response message between CSP and TPA. For every request from TPA, the CSP generates a response as a proof and sent to TPA. The data verification consists of a challenge message generation, proof generation and proof verification phases.

Challenge message generation: In each auditing round, TPA sends a file signature request to CSP. After receiving the file signature from CSP, then TPA verifies the signature while comparing the stored signature in his table T_{tpa} . Once the file signature is verified, then TPA prepares a challenge message for verification of outsourced data blocks on a cloud server. The challenge message contains a set of random block identification numbers of the outsourced file and it is sent to the CSP.

Proof generation: After receiving challenge message from TPA, the CSP prepares the proof as a response to the request using meta-tada T_{csp} and data blocks. The proof message calculation for the data block F_i is as follows;

$$w = (\sigma_{i2})^{-1}$$

$$u_1 = (z * w) \%_q, \text{ where } z = h(F_1)$$

$$u_2 = (\sigma_{i1} * w) \%_q$$

$$Q(x, y) = u_1 * g(x, y) + u_2 * pb_{key}(x, y)$$

$$\sigma'_{i1} = x \%_q$$

After completion of proof message σ'_{i1} computation, CSP sent back to TPA as a response to the requested challenge.

Proof verification: The TPA verify the received message σ'_{i1} with the signature of the data block F_i stored in meta-data table T_{tpa} . If the signatures are matches i.e ($\sigma'_{i1} == \sigma_{i1}$), then TPA reports data blocks are not altered message as 1. Otherwise, it reported as 0. The detailed procedure for data verification as shown in the algorithm 9.

Alg. 9 TPA: Data Verification Algorithm

- 1 TPA sends a request to cloud server for verification of file tag id which is on the cloud server
- 2 Once the file id is verified, then send the challenge request message to verify the data blocks on cloud server; $chal = b_i$ to b_j , where b_i is the i^{th} block identifier
- 3 cloud server computes the verification signature (σ'_{i1})

Signature verification($Pb_k, F_i, \mathbf{r}, \mathbf{s}$):

 - a. Find the hash value of file block; $z = H(F_i)$
 - b. calculate the curve point;

$$w = s^{-1}, u_1 = (z * w) \%_q, u_2 = (r * w) \%_q$$

$$x, y = u_1 * g(x, y) + u_2 * pb_{key}(x, y)$$
 - c. extract the x coordinate as signature. $\sigma'_{i1} = x \%_q$
- 4 Retrieve the signatures σ'_{i1} of file blocks
- 5 Verify the validity of the data blocks

if ($\sigma'_{i1} == \sigma_{i1}$) **then**
 Data blocks are verified
else
 Data blocks are modified
endif

5.4.3 Batch auditing

In the proposed data auditing method, the TPA is not only audit the single data owner single file, but also support multiple user and multiple file data auditing tasks. Hence, batch auditing is also introduced in the proposed design.

The batch auditing tasks are considered in two ways; single data owner with multiple files and multiple data owners with multiple files.

Let consider a data owner DO_i having f_{ij} list files, where $i = 1$ to s , j is the i^{th} data owner files. In public auditing, the data owners delegate the batch auditing task to TPA.

Single data owner with multiple files: In case of single data owner $s = 1$ and there are m data files. The TPA prepares the challenge message (DO, l_1, l_2) , where l_1 and l_2 are the lower and higher index of the data block and sends the challenge message to CSP. The CSP generates the proof message as a response for the auditing task and sent back to the TPA.

After receiving the response message from the CSP, the TPA generate a verification message (V) 0 or 1, where 0 means data blocks are altered and 1 means not altered.

$$V = \prod_{i=l_1}^{l_2} proof(i) \quad (5.20)$$

Multiple data owners with multiple files: In case of multiple data owner $s > 1$ and $m > 1$. The data owner sends the challenge message (DO_i, l_1, l_2) to cloud server. After receiving the batch auditing task, the CSP prepares a response message $proof(i, j)$ and then sent back to TPA. The TPA verify all the blocks and generate verification message V .

$$V = \prod_{i=1}^s \prod_{j=l_1}^{l_2} proof(i, j) \quad (5.21)$$

5.5 Summary

This chapter, focuses on proposed data confidentiality and remote data integrity techniques on cloud system. For confidentiality, we explains the data encryption and decryption modern symmetric key algorithms using key rotation technique. And for remote data auditing identity based and linear authentication protocol and ECDSA public key digital signature algorithm is presented. In the next chapter, we present the simulation results and performance analysis of the proposed data auditing methods.

Chapter -6

SIMULATION RESULTS AND ANALYSIS

6.1 Simulation setup

To evaluate the performance of our proposed data auditing method computation overhead we have considered the following simulation setup are considered as shown the Table 6.1. All the proposed algorithms are implemented using Python programming language with built-in cryptography functions in Python library. The simulation result is tested on Amazon Web Service EC2 virtual machines(VM).

Table 6.1: Simulation setup

Parameter	Configuration
TPA VM	t2.small model with 1-CPU, 2GB memory, 8 GB EBS storage
ec2 server VM	t2.small model with 1-CPU, 2GB memory, 8 GB EBS storage
Data owner	t2.small model with 1-CPU, 2GB memory, 8 GB EBS storage
Operating system	Ubuntu 14.04 LTS with PV visualization
security parameter	160 bits
Elliptic curve field	$F_{192}(0, 1)$
Bilinear pairing	Tate pairing
Programming Language	Python 3.4

Table 6.2: File text and Cipher text

File Text	Ciphertext
a b c d e f g h i j k l m n o p q r s t u v w x y z 1 2 3 4 5 6 7 8 9 letasdfsadf sdf dkfjlsfdjsad sadflkjasdlfasdfasdfsaldfjalsjfasdfalsdjflasdjfls lsajfdlasjdfldfjlsajdfldfjlsajdfldfjlsajdfldfjls asdlfjasdfldfjlsajdfldfjlsajdfldfjlsajdfldfjls dfljasdlfkjasldfjajfldsajflksajflksajdfldfjls	@b@M@Q@Y@KG@p[@r@'@W@ @4N @Wz@#@m@s@*@iH@;m@F@%@9@\@ 5j@<J@;@K@B@Q LamWVv nt@mnt@],K YndVQMICWi@vO]Kh9e^RQhKcDbKORn Km^QQtM WdeKCR]K@and]rKhCdbMtJR n YiTRCMKQTFIrkVh [eVWVW rWihvOZnKkF dZQKhCdentJyMIhiTZCIntTe vkQKMamWV WnIt^mntJ8I YiTyvMKW&m kVh,[Ff]VW IrV h3OZInZnfyvnht&e vJ8nKYFfVWnICfihC#RN Kk^RrnKW&FMv

6.2 Data Encryption and Decryption

The experiment is carried out on the repository of text files with varying size. For the testing purpose the text file is composed of alphanumeric characters. The Encoding Map is restricted to have mapping values for lower case alphabets and numerical values. The key used for the experimental purpose is "*doitdueletscshec*" which is of 256 bits in size and a fixed key size [110]. The file is divided into blocks of 256 characters i.e 4096 bits in size. The part of the source file data is encrypted before storing in cloud server as shown in Table 6.2.

The vital or key operations in both processes are CA shifter and CA inverter. The time for encryption/decryption directly depends on these two operations. The number of movements of CA and its inversion process decides the accuracy of encryption/decryption. Along with CA shifter, the key is also rotated for every block character. This ensures that same key is not used for multiple characters. The analysis is performed on different files and number of movements used for shifting CA and key remains same. The number of character shift and inversion operations comparison for

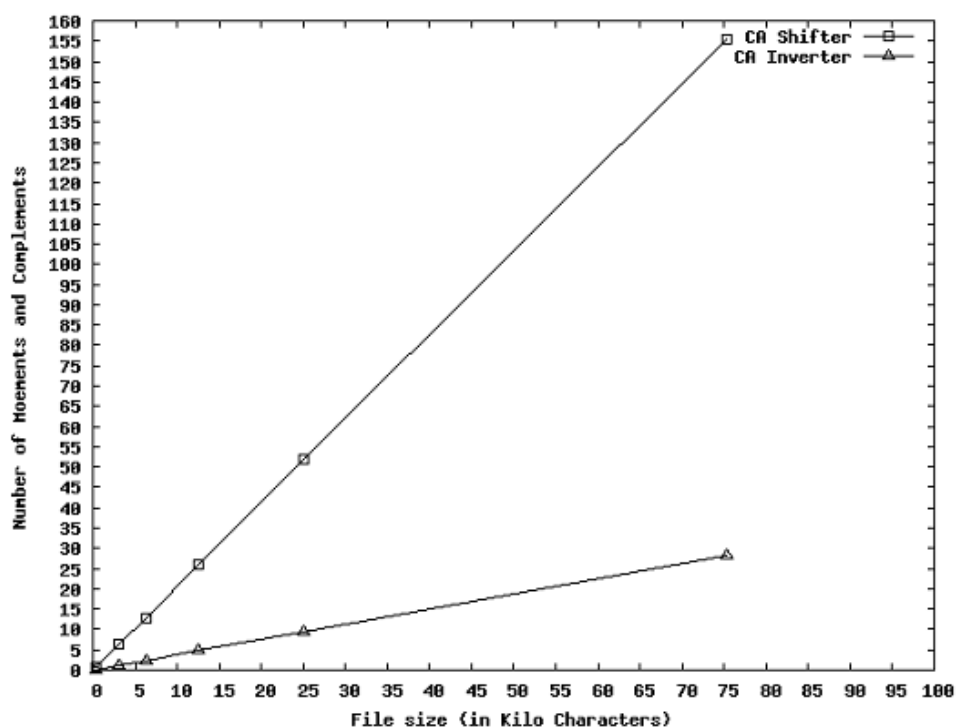


Figure 6.1: CA Shifter and CA inverter comparison

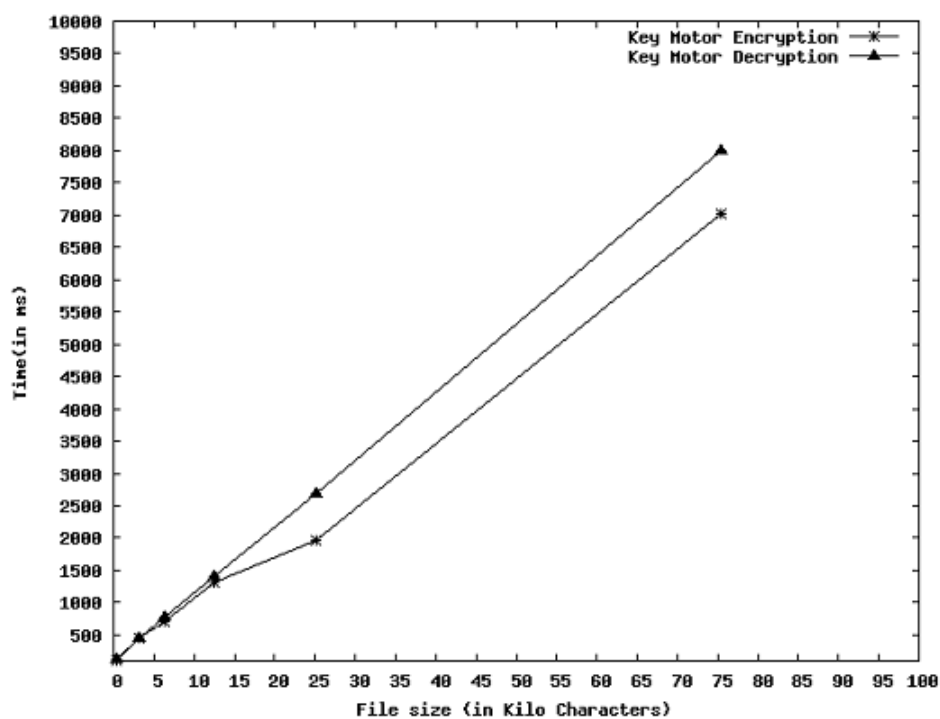


Figure 6.2: Key Motor Encryption and Decryption Time comparison

different file size as shown in Figure 6.1.

In Figure 6.2, the execution time is plotted on the graph. With increase in file size the number of movements and complements are high and hence the execution time is directly proportional to file size. It is observed that decryption is taking more time than encryption process.

6.3 Remote Data Audit using Protocol(RDAP)

In this section, we explain the result and performance analysis of the proposed remote data audit using identity-based and linear authentication techniques [104, 105]. The performance analysis is analyzed interns of communication cost and computation cost. Finally, the performance of the proposed methods is compared.

6.3.1 Communication Cost:

The design of proposed method consists of initial file setup and data audit phase. The auditing phase is a sequence of request and response communication among Data owner, TPA, and CSP. For every data audit, TPA prepares a challenge message and send to the CSP. The challenge message contains the c number of data blocks identifiers, therefore the communication cost between TPA and CSP is $O(c)$. Once the challenge message received, then the CSP prepares the proof and sends to TPA as a response message. So that the communication cost between CSP and TPA is $O(1)$. Finally, TPA verifies the proof message and send the verification message 0 or 1 to data owner, which has $O(1)$ communication overhead. The total communication overhead is the sum of the communication cost among TPA, CSP and Data owner i.e, $O(c) + O(1) + O(1) = O(c)$.

6.3.2 Computation Cost:

In order to measure the computation cost of the proposed remote data auditing method using identity-based and linear authentication based method we are considered the tag generation time, tag verification time and data verification time with different file sizes, data block sizes and audit batch sizes.

Tag generation cost: The cost tag generation using identity-based and linear authentication-based protocol technique for different file size with 265KB of each data block size as shown in Table 6.3. The first column of the table represents the different file size and the second column represents the tag generation time for the entire file. It is observed that the tag generation time is directly proportional to the file size.

Table 6.3: Tag generation time for different file sizes

File Size(MB)	Identity-based (time in seconds)	Linear authenticator (time in seconds)
1	1.87	1.94
2	3.17	3.52
4	4.27	4.52
6	6.85	6.64
8	10.05	10.87
10	14.65	15.26
12	18.59	18.92
14	19.87	19.42

The simulation result for tag generation time for 1MB file size with different data block size as shown in Table 6.4. It is observed that the smaller sized data block takes more computational and storage overhead than the larger sized data block. But for detection of the invalid block on a server, the larger sized data blocks takes more time

in the data audit phase.

Table 6.4: Tag generation time for different data block sizes

Data block size(KB)	Identity-based (time in seconds)	Linear authenticator (time in seconds)
256	1.85	1.91
500	1.05	1.12
768	0.67	0.68
1024	0.38	0.46

Data Verification Cost: Table 6.5 shows the comparison of tag verification cost on cloud server using the identity and linear authentication techniques. In this scenario, we considered a 1MB file size with different data block sizes for tag verification. The result shows that larger block sized file take less time than the smaller blocks. Besides, the identity-based technique tag verification takes more time than linear authentication technique. This change is due to the storage of data blocks on different servers in the identity-based method.

Table 6.5: Data verification cost with a different block size

Data block size(KB)	Identity-based (time in seconds)	Linear authenticator (time in seconds)
256	0.085	0.091
500	0.045	0.152
768	0.037	0.178
1024	0.016	0.196

Batch Auditing Cost: In Table 6.6., shows, the performance comparison of batch auditing cost on cloud server using the identity and linear authentication techniques. In this scenario we considered a 10MB file with 256KB of each data block and the performance of the proposed method is analyzed based on the different number of data

blocks in each auditing task. The result shows that the performance of larger batch size auditing is better than smaller batch size.

Table 6.6: Data verification cost with different batch size

Data block size(KB)	Identity-based (time in seconds)	Linear authenticator (time in seconds)
4	3.1856	3.012
8	5.0127	4.671
12	8.5327	7.178
24	10.0154	9.196

Modified Data Block Detection: Due to internal or external attack on the out-sourced the cloud service provider can replace the attacked block with other valid data block and its signature. The probability of detecting the modified data blocks on a cloud server is calculated as follows;

Let consider n is the total number of data blocks of a file F on a cloud server, d is the number of modified data blocks on a cloud server, and c is the number of challenged blocks for each auditing task. Then, the probability of detecting the invalid data blocks Pr is defined as;

$$1 - \left(\frac{n-d}{n}\right)^c \leq Pr \leq 1 - \left(\frac{n-c+1-d}{n-c+1}\right)^c$$

Where Pr is computed as;

$$Pr = \left\{ 1 - \left(\frac{n-d}{n}\right) \left(\frac{n-1-d}{n-1}\right) \dots \left(\frac{n-c+1-d}{n-c+1}\right) \right\}$$

6.4 RDADS Simulation Results

This section presents the simulation results of the proposed RDADS [106] using ECDSA algorithm techniques. The performance of the proposed remote data auditing methods computation and communication overheads during initial file setup and data audit phases are presented in the following section.

6.4.1 Communication Cost

The communication cost in the initial file setup is the same order of growth in all the proposed data auditing methods. But the communication cost between TPA and cloud service provider varies in the data audit phase. So that, we compare the computation cost for data audit in the proposed method.

Consider a batch auditing with K data owners and C cloud servers, a number of challenging blocks in each task is t , and the size of each block is s . The total cost during the challenging phase is $O(ts)$, so that the communication cost for proof generation depends on the number of challenging blocks and size of each block. Finally, the server sends an only proof message to the TPA, so that the communication cost from CSP to TPA is $O(1)$. The total communication cost in the auditing phase is the sum of the challenging task, proof generation and proof communication between CSP and TPA i.e, $O(1) + O(ts) + O(1) = O(ts)$.

6.4.2 Computation Cost

Due to the large data file, we use the sampling auditing method to verify the outsourced data in the cloud. The computation cost of the TPA and CSP to audit data blocks on a single server and multiple data owners is presented in the following section.

The performance analysis of the RDASDS and RDAP methods interns of computation cost is analyzed using the following parameters; Signature generation cost, File setup and upload time, Data block verification time, Detection of the modified data block, CSP vs TPA computation time, RDAP vs RDADS signature and data verification.

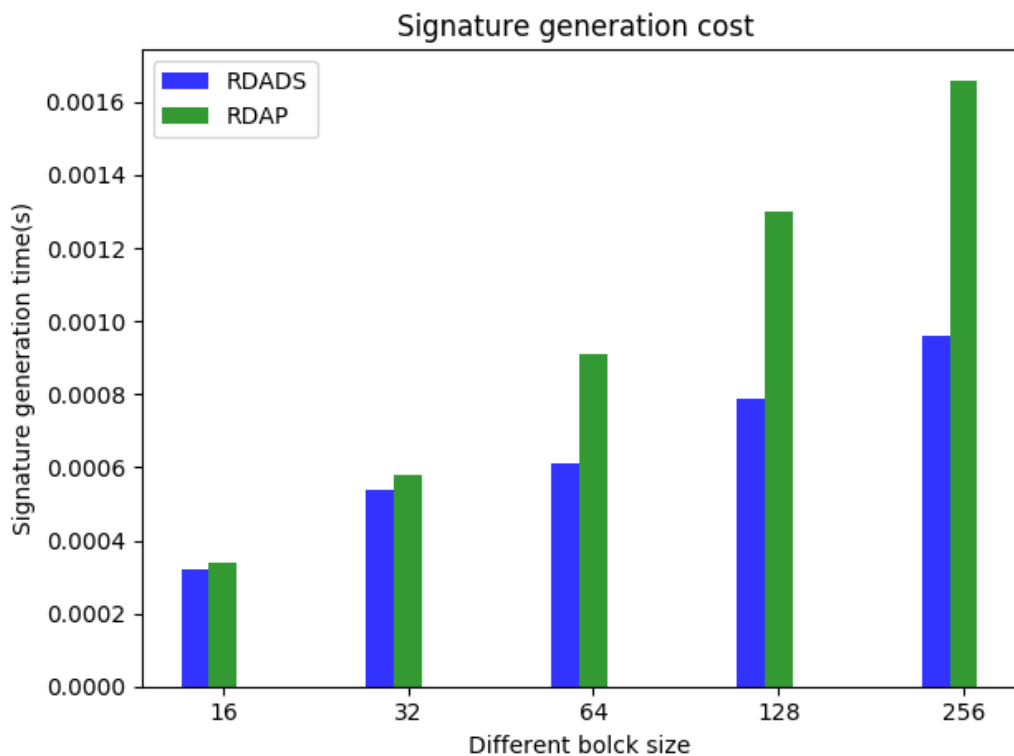


Figure 6.3: Signature generation cost for different blocks

Signature Generation Cost: The computational comparison between RDAP using a linear authenticator and RDADS using elliptic curve signature generation of a 10MB data file with different block size as shown in Figure 6.3. In Figure 6.3, the X-axis represents the different data block sizes in terms of Kilo Bytes and Y-axis represents the signature generation cost in seconds(s). The simulation result shows that the computational overhead for data block signature generation of RDADS method has

a lower order of growth than RDAP method. For smaller sized data blocks signature generation cost of both the methods has same growth order. But for the larger sized data blocks, RDAP has a higher order of growth order than RDADS. This change is due to the number of computation operations for a signature generation in RDAP is more expensive than RDADS method.

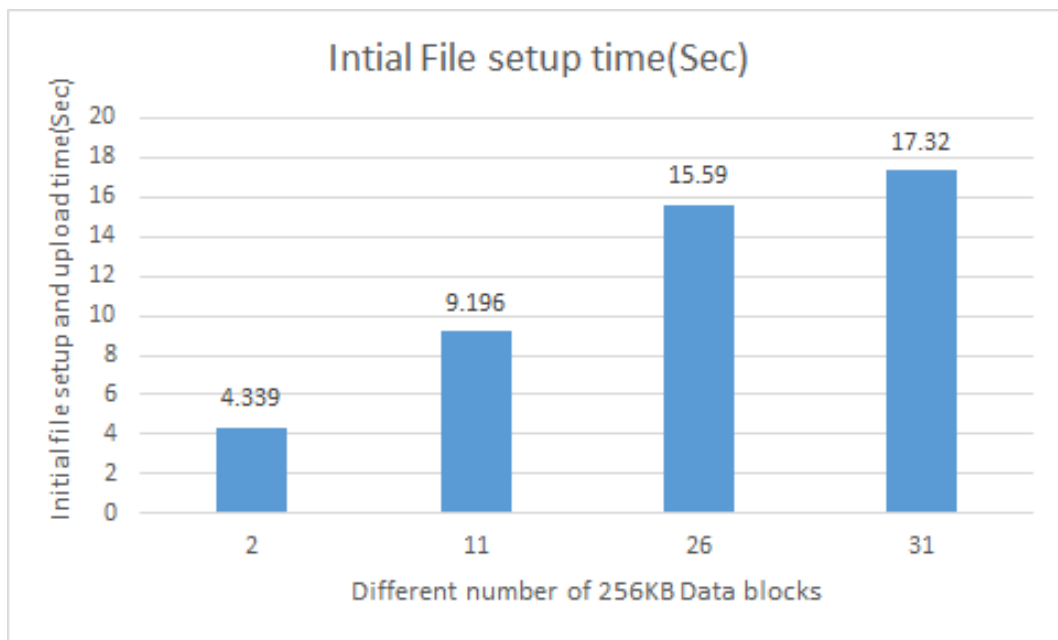


Figure 6.4: Initial File setup and Upload time(Sec)

File Setup and Upload Time: Figure 6.4 shows the computational and communication costs for signature generation and storing of data blocks in a cloud server in the initial file setup phase.

In Figure 6.4, the X-axis represents the different number of the 256KB data block(different file sizes) and Y-axis represents the file setup cost in seconds(s). Apparently, it shows that, the computational overhead of initial file setup for larger file is better than the smaller file. The communication overhead between data owner and cloud service provider of RDADS and RDAP are same in the initial setup phase.

Data Audit Time: The computational cost comparison between RDADS and RDAP method for audit different number of 256KB data blocks challenging task as shown in Figure 6.5.

In Figure 6.5, X-axis represents the different number of data blocks in each batch and Y-axis represents the data verification cost between TPA and CSP. The result shows that for the larger batch size RDADS method takes more cost than the RDAP method. This changes due to the expensive elliptic curve points operations are involved in RDADS during the data verification phase. But in the security point of view, RDADS method is better than the RDAP method.

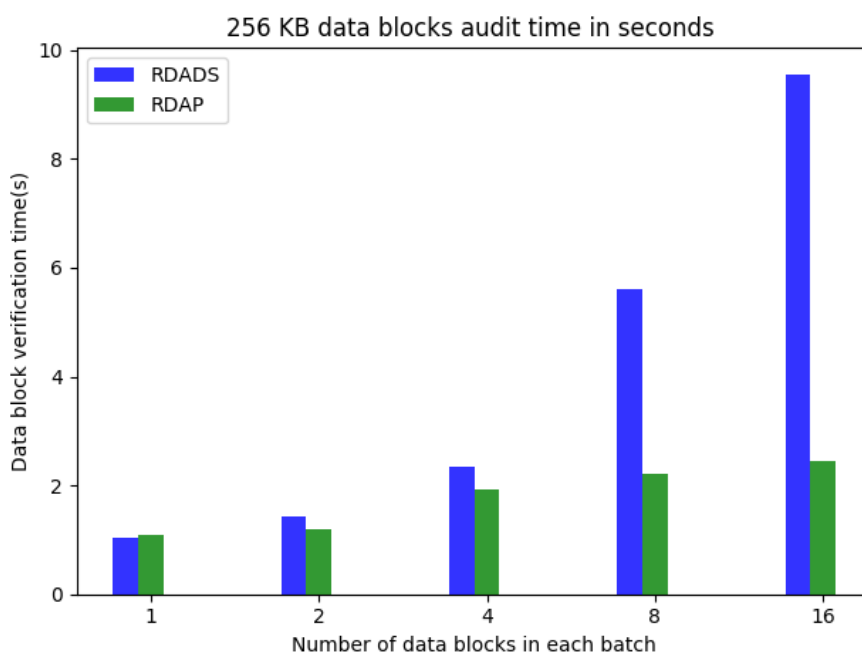


Figure 6.5: Data block verification cost

Modified Data Block Audit Time: The performance analysis of the RDADS and RDAP for corrupted data block verification as shown in Figure 6.6. As compared to

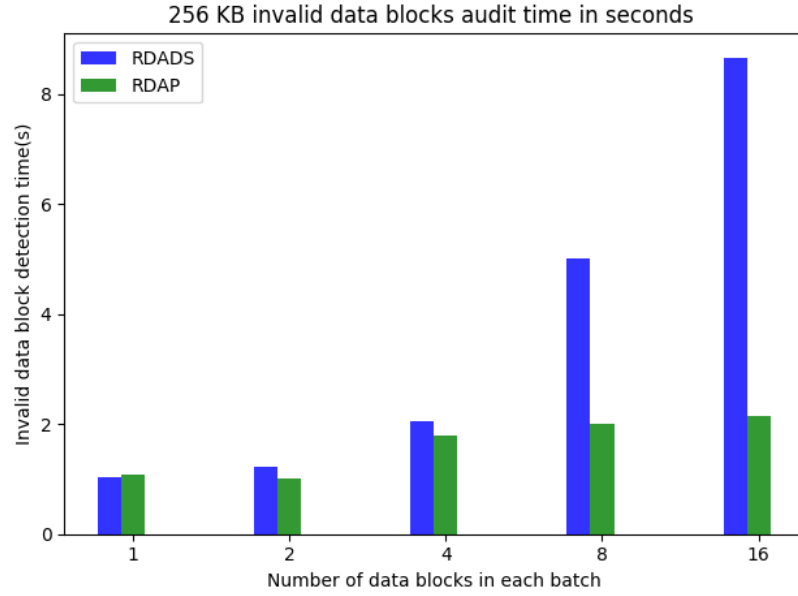


Figure 6.6: Incorrect verification blocks time

RDAP method of data auditing, the RDADS method is higher computation cost for larger batch size, because RDADS contains expensive elliptic curve points operations in the auditing phase.

Depending on the trust between the data owner and CSP the frequency of auditing is decided. The TPA select the t number of the data block in each auditing task to verify the integrity of data blocks on the cloud. The probability of detection on any corrupted data block sector s is defined as $Pr(t, s) = 1 - (1 - \rho)^{ts}$, where ρ is the probability of data corrupted on cloud and t is the auditing batch size.

Computation Cost Comparison between CSP and TPA: As Figure 6.7 and Figure 6.8, shows that audit time comparison between TPA and CSP to verify the outsourced 256KB and 50KB data blocks using RDADS method respectively.

In Figure 6.7, X-axis represents the different batch sizes of 256KB blocks and Y-axis represents the computation overhead in seconds. The simulation result shows

that, TPA takes negligible computation cost than CSP, because of the TPA delegate the auditing task to the CSP so that CSP computation overhead varies for different bath size.

The computation overhead comparison between TPA and CSP for 50KB of data blocks audit as shown in the Figure 6.8. As compared to 256KB of data blocks, TPA takes same computation cost as 50KB blocks. But the CSP computation cost varies based on the batch size in both the cases.

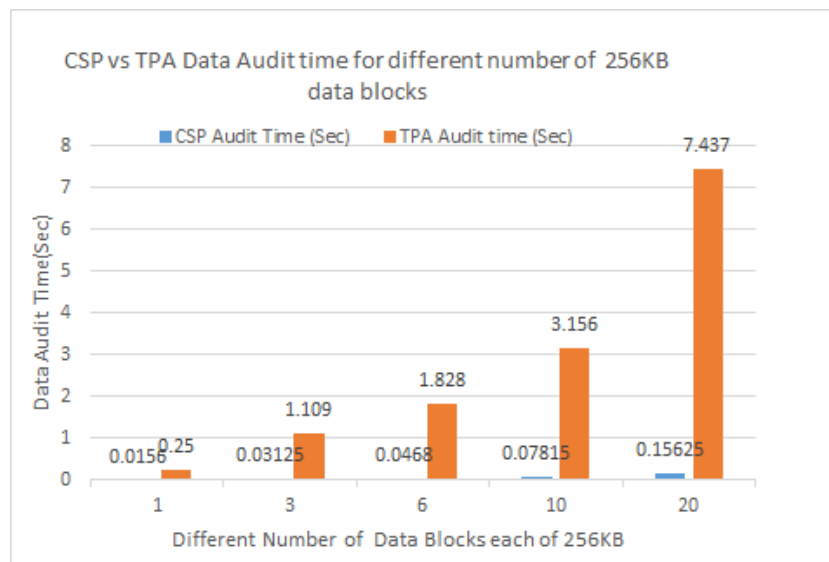


Figure 6.7: CSP and TPA 256KB Data Blocks Audit Time(Sec)

Computation Cost Comparison between RDAP and RDADS : In Figure 6.9, depicts the comparison on data blocks signature generation time using RDAP and RDADS method for different file size with 256KB of data blocks. It is easily observed that RDADS method has less computation time compared to RDAP method. This difference is due to RDADS algorithm takes less number of operations to generate the block signature compared to RDAD method.

Figure 6.10, shows, the comparison of data blocks audit computation cost using

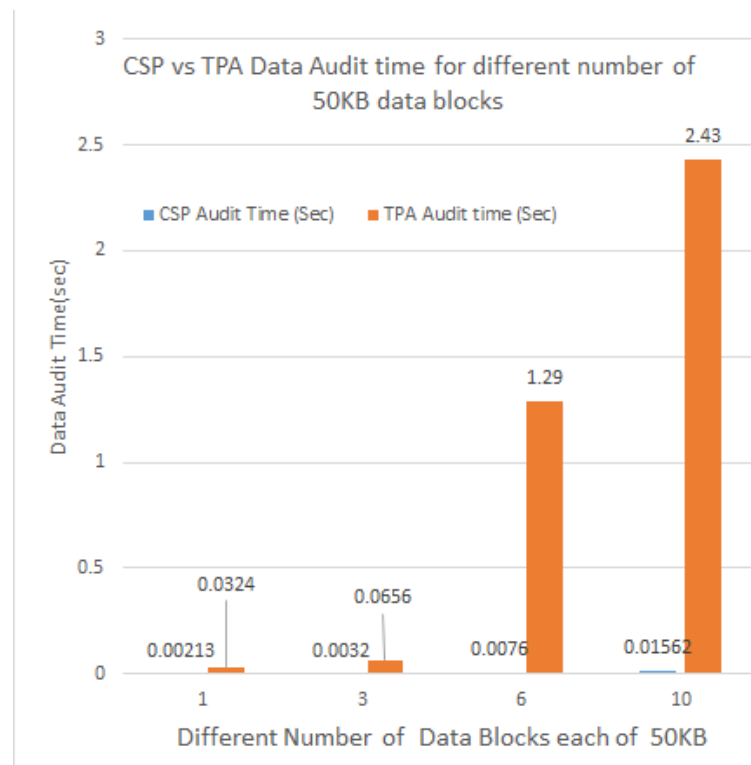


Figure 6.8: CSP and TPA 50KB Data Blocks Audit Time(Sec)

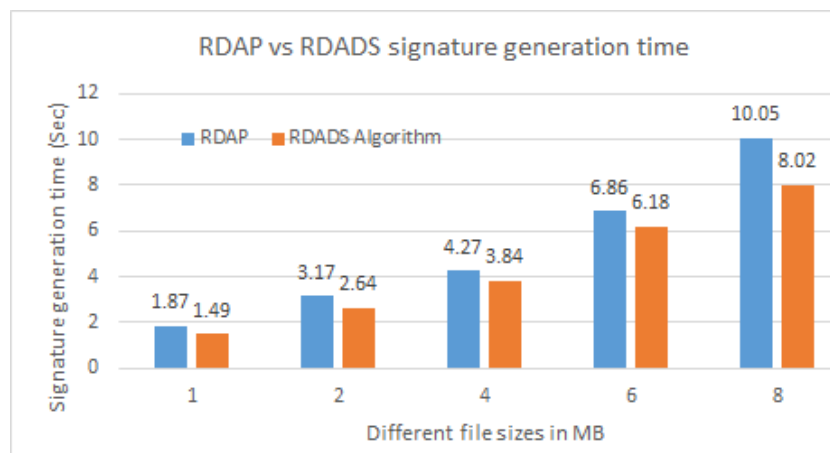


Figure 6.9: RDAP vs RDADS Signature generation time for 256KB blocks

RDAP and RDADS method for a different number of data blocks each of 256KB. It is easily observed that RDADS method has less computation cost for the smaller number of data blocks compared to RDAP method and more computation cost for the larger number of data blocks. This change is due to RDADS Elliptic curve method points complex operations is increases for the larger blocks. But for the security point

of view, RDADS algorithm is better to secure data audit method compared to RDAP method.

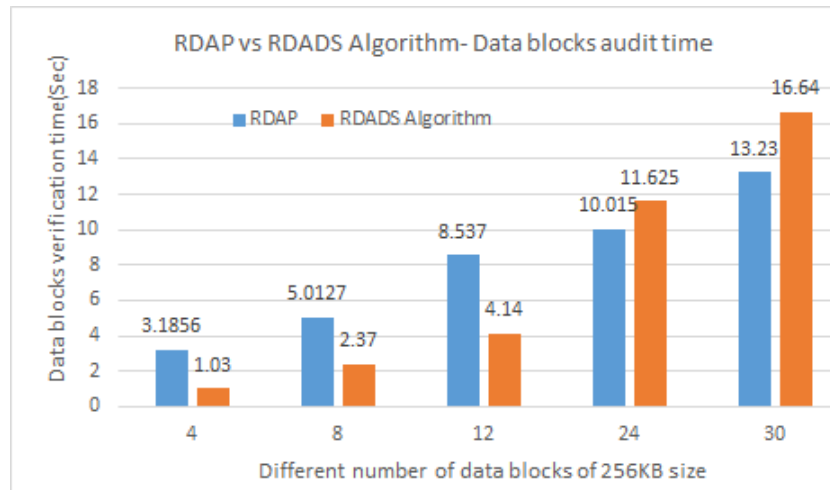


Figure 6.10: RDAP vs RDADS 256KB data blocks verification time

6.5 Summary

In this chapter, we have presented the simulation results of data encryption, decryption, remote data auditing using protocol and ECDSA schemes on the cloud server. The performance of data auditing schemes with different data block size is analyzed with respect to data block signature generation cost, verification cost, and data auditing cost. Finally, we concluding remarks of this thesis and further future direction of this work is presented in the next chapter.

Chapter -7

CONCLUSIONS

In this thesis, our main objective is to ensure data privacy and security of outsourcing data on cloud storage system. To protect the data we recommend a third-party auditor for secure data auditing.

Our first objective is to define a new technique to improve data confidentiality in the cloud for sharing the data over internet between different users. To fulfill our first objective, we have proposed a data encryption with key rotation techniques to improve the communication and computation overhead during data audit process. In chapter four, we present a novel symmetry key based on blocked level data encryption with key rotation algorithm for securing and sharing the outsourced data over the internet with the authorized users. The proposed solution ensures that the confidentiality of the cloud data with flexible access control and efficient data audit operation. In this method, the data owner uploads the encrypted file blocks to cloud server and stores the encrypted symmetric key for deciphering into the metadata, which ensures the confidentiality. Besides, our proposed scheme utilizes the Diffie Hellman key exchange algorithm for secure symmetric key strength between user and cloud server. That means, only authorized users can access symmetric key for data deciphering. By using this without updating the private key of the other user's revocation can be achieved.

Our second objective addresses secure remote data verification on cloud considering security level verifiability and storage overhead. In order to fulfill this objective,

we have proposed a secure public data auditing techniques using protocol and digital signature scheme. Based on the comparison of our digital signature data auditing scheme benefits from security level and minimum computational overhead. In addition, our proposed method is designed for constant storage and computational support for each auditing task.

Thus, in response to our third and fourth objectives focus on secure block level data auditing between auditor and cloud service provider with low computation and storage overhead. The proposed data auditing scheme utilizes the block level data auditing with masking response message on encrypted data to provide data privacy to untrusted entities. In response to the fourth objective, data audit using digital signature method provides acceptable computation and storage communication overhead at server side due to light-weight elliptic curve points operations.

Regarding the data privacy and security of the proposed remote data auditing scheme on an untrusted cloud service provider, several other security issues are still a challenging task in cloud computing. Implementing cost effective security model is also important for cloud service provider to protect the owner's data on cloud storage system.

For future perspective, our proposed key rotation symmetry data encryption process is relying on personalized symmetric key and it takes heavy computational overhead to decrypt the data. So that, generating an identity based deciphering key is an alternative solution. In cloud computing, more than 60% of resource-constrained devices are used to share data over the internet. Due to this, implementing light-weight security model can improve the power efficiency of these devices. The communication cost

of our proposed data auditing schemes affecting the bandwidth consumption due to the location of the auditor and the location of storage . So that, we can implement a customizable data owners auditor to evaluate the impact of data owners location. We have shown the data privacy performance of data auditing schemes within a same cloud service providers. Consequently, it would be important to evaluate the performance of our proposed scheme on multiple cloud service providers. We can extend this work for dynamic block level operations such as block update, insertion, deletion) as a future research.

To conclude, our main objective was to address the cloud data privacy and security issues using data confidentiality and remote data integrity verification. We have provided a protocol and digital signature based on cryptography approaches to address the data security issues in the cloud.

Finally, we believe that cloud data storage security challenges are not limited and also it is an important research area in cloud computing for secure data sharing.

Bibliography

- [1] Badger, L., Grance, T., Patt-Corner, R., and Voas, J, “Draft cloud computing synopsis and recommendations,” in *National Institute of Standards and Technology (NIST) Special Publication*, <http://csrc.nist.gov/publications/drafts/800-146/Draft-NIST-SP800-146.pdf>, pp. 800–146, 2011.
- [2] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, and A. V. Vasilakos, “Security and privacy for storage and computation in cloud computing,” *Information Sciences*, vol. 258, pp. 371–386, 2014.
- [3] J.D. Meier, Alex Mackman, Michael Dunner, Srinath Vasireddy, Ray Escamilla and Anandha Murukan, “Improving web application security: Threats and countermeasures,” in *Microsoft Corporation*, 2006.
- [4] Eric Bauer, “Improving operational efficiency of applications via cloud computing,” *IEEE Transactions on Cloud Computing*, vol. 5, no. 1, pp. 12–19, 2018.
- [5] Al Kovalick, “Cloud computing for the media facility: Concepts and applications,” *IEEE SMPTE Motion Imaging Journal*, vol. 120, no. 2, pp. 20–29, 2011.
- [6] Linlin Wu; Saurabh Kumar Garg; Steve Versteeg; Rajkumar Buyya, “Sla-based resource provisioning for hosted software-as-a-service applications in cloud computing environments,” *IEEE Transactions on Services Computing*, vol. 7, no. 3, pp. 465 – 485, 2014.

- [7] Ismail Butun, Melike Erol-Kantarci, Burak Kantarci, Houbing Song, “Cloud-centric multi-level authentication as a service for secure public safety device networks,” *IEEE Communications Magazine*, vol. 54, no. 4, pp. 47–53, 2016.
- [8] Ke Li, Weiming Zhang, Ce Yang, Nenghai Yu, “Security analysis on one-to-many order preserving encryption-based cloud data search,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 9, pp. 1918–1926, 2015.
- [9] Taeho Jung, Xiang-Yang Li, Zhiguo Wan, Meng Wan, “Control cloud data access privilege and anonymity with fully anonymous attribute-based encryption,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 1, pp. 190–199, 2015.
- [10] S. Seo, M. Nabeel, X. Ding, and E. Bertino, “CL-PRE: A certificateless proxy reencryption scheme for secure data sharing with public cloud,” in *IEEE Trans. Knowl. Data Eng.*, vol. 26, no. 9, pp. 2107–2119, 2013.
- [11] Seung-Hyun Seo, Mohamed Nabeel, Xiaoyu Ding, Elisa Bertino, “An efficient certificateless encryption for secure data sharing in public clouds,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 26, no. 9, pp. 2107 – 2119, 2014.
- [12] Yunchuan Sun, Junsheng Zhang, Yongping Xiong, and Guangyu Zhu, “Data security and privacy in cloud computing,” in *International Journal of Distributed Sensor Networks*, 2014.

- [13] C. Wang, N. Cao, K. Ren, and W. Lou, “Enabling secure and efficient ranked keyword search over outsourced cloud data,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 8, pp. 1467–1479, 2012.
- [14] C. I. Fan, V. S. M. Huang, and H. M. Ruan, “Arbitrary-state attribute-based encryption with dynamic membership,” *IEEE Transactions on Computers*, vol. 63, no. 8, pp. 1951–1961, 2014.
- [15] Jin Li, Xiaofeng Chen, State Key, and Xinyi Huang, “Csecure distributed deduplication systems with improved reliability,” *IEEE Transactions on Computers*, vol. 64, no. 12, pp. 3569 – 3579, 2015.
- [16] Lan Zhou, Varadharajan V, Hitchens M, “Integrating trust with cryptographic role-based access control for secure cloud data storage trust,” in *12th IEEE International Conference on Security and Privacy in Computing and Communications (TrustCom)*, pp. 560–569, 2013.
- [17] G. Y. F. G. Rongmao Chen, Yi Mu and X. Wang, “Dual-server public-key encryption with keyword search for secure cloud storage,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 4, pp. 789–798, APRIL 2016.
- [18] Jie Xu, Qiaoyan Wen, Wenmin Li, and Zhengping Jin, “Circuit ciphertext-policy attribute-based hybrid encryption with verifiable delegation in cloud computing,” in *IEEE Transactions on Parallel and Distributed Systems*, vol. 27, no. 1, pp. 119–134, 2016.

- [19] Rivest, R. L. and Shamir, A. and Adleman, L., “A method for obtaining digital signatures and public-key crypto systems,” *Commun. ACM*, vol. 21, pp. 120–126, Feb. 1978.
- [20] Rivest, R., “The MD5 Message-Digest Algorithm,” <https://tools.ietf.org/html/rfc1321>, 2014.
- [21] Yan Zhu, Gail-Joon Ahn, Hongxin Hu, Stephen S. Yau, Ho G. An, and Chang-Jun Hu, “Dynamic audit services for outsourced storages in clouds,” in *IEEE Transactions on Services Computing*, vol. 6, pp. 227–238, APRIL-JUNE 2013.
- [22] Jiang Deng, Chunxiang Xu, Huai Wu and Liju Dong, “A new certificateless signature with enhanced security and aggregation version,” *Concurrency and Computation: Practice and Experience*, vol. 28, no. 4, pp. 1124–1133, 2016.
- [23] Tao Jiang, Xiaofeng Chen, and Jianfeng Ma, “Public integrity auditing for shared dynamic cloud data with group user revocation,” *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2363–2373, 2016.
- [24] Cong Wang, Sherman S.M. Chow, Qian Wang, Kui Ren, and Wenjing Lou, “Privacy-preserving public auditing for secure cloud storage,” in *IEEE Transactions on Computers*, vol. 22, no. 2, 2013.
- [25] Jiawei Yuan; Shucheng Yu, “Public integrity auditing for dynamic data sharing with multiuser modification,” *IEEE Transactions on Services Computing*, vol. 10, pp. 1717–1726, 2015.
- [26] A. Juels and B. S. Kaliski, Jr, “Pors: Proofs of retrievability for large files,” vol. 5, pp. 584 – 597, 2007.

- [27] C. Gentry, “Fully homomorphic encryption using ideal lattices,” in *Proc. 41st Ann. ACM Symp. Theory of Computing (STOC 09)*, pp. 169 – 178, 2009.
- [28] Cong Wang and Kui Ren, Jin Li , “Toward publicly auditable secure cloud data storage services,” in *IEEE Network*, 2010.
- [29] Giuseppe Ateniese, Randal Burns, Reza Curtmola, Joseph Herring, Lea Kissner, Zachary Peterson, Dawn Song, “Provable data possession at untrusted stores,” in *12th IEEE International Conference on CCS’07*, pp. 598 – 610, 2007.
- [30] Boyang Wang, Baochun Li, and Hui Li, “Panda: Public auditing for shared data with efficient user revocation in the cloud,” in *IEEE Transactions on Services Computing*, 2013.
- [31] Boyang Wang; Hui Li; Xuefeng Liu; Fenghua Li; Xiaoqing Li, “Efficient public verification on the integrity of multi-owner data in the cloud,” *Journal of Communications and Networks*, vol. 16, no. 6, pp. 592–599, 2014.
- [32] L. Chen et al, “An efficient and privacy-preserving semantic multi-keyword ranked search over encrypted cloud data,” in *International Journal of Security and Its Applications*, vol. 8, no. 2, pp. 323 – 332, 2014.
- [33] Ayad Barsoum and Anwar Hasan, “Enabling dynamic data and indirect mutual trust for cloud computing storage systems,” in *IEEE Transactions on Parallel and Distributed Systems*, 2012.

- [34] Aaram Yun, Chunhui Shi, Yongdae Kim, “On protecting integrity and confidentiality of cryptographic file system for outsourced storage,” in *IEEE Proceedings of the CCSW’09*, pp. 67 – 75, 2009.
- [35] Alexandra Boldyreva, Vipul Goyal, Virendra Kumar, “Identity-based encryption with efficient revocation,” in *IEEE Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS 2008, ACM Press*, 2008.
- [36] Ayad Ibrahim Abdulsada, Aqeel N. Mohammad Ali, Zaid Ameen Abduljabbar, Haider Sh.Hashim, “Secure image retrieval over untrusted cloud servers,” *International Journal of Engineering and Advanced Technology (IJEAT)*, vol. 3, no. 1, pp. 140–147, 2013.
- [37] Jing-Jang Hwang, Taoyuan, Taiwan, Yi-Chang Hsu, Chien-Hsing Wu,, “A business model for cloud computing based on a separate encryption and decryption service,” in *International Conference on Information Science and Applications (ICISA)*, pp. 1–7, 2011.
- [38] Junzuo Lai, Deng R H, Chaowen Guan, Jian Weng, “Attribute-based encryption with verifiable outsourced decryption,” in *IEEE Transactions on Information Forensics and Security*, vol. 8, no. 8, pp. 1343 – 1354, 2013.
- [39] B. Qin, R. H. Deng, S. Liu, and S. Ma, “Attribute-based encryption with efficient verifiable outsourced decryption,” *IEEE Transactions on Information Forensics and Security*, vol. 10, no. 7, pp. 1384–1393, 2015.

- [40] J. Hur, “Improving security and efficiency in attribute-based data sharing,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2271–2282, 2013.
- [41] J. Bethencourt, A. Sahai, and B. Waters, “Ciphertext-policy attribute-based encryption,” *IEEE Symposium on Security and Privacy*, vol. 07, pp. 321–334, 2007.
- [42] Xu, Jia and Chang, Ee-Chien, “Towards efficient provable data possession,” *IACR Cryptology ePrint Archive*, vol. 2011, p. 574, 2011.
- [43] A. Sahai and B. Waters, “Fuzzy identity based encryption,” in *IEEE Proceeding of 30th Annual International Conference on Theory Applied Cryptography and Technology*, pp. 457 – 473, 2015.
- [44] J. Lai, R. H. Deng, C. Guan, and J. Weng, “Attribute-based encryption with verifiable outsourced decryption,” in *IEEE Transactions on Information Forensics Security*, vol. 8, no. 8, pp. 1343 – 1354, 2013.
- [45] Jin Li, et al., “Enabling efficient fuzzy keyword search over encrypted data in cloud computing,” in *Proceedings of the IEEE Conference on Information Security*, 2011.
- [46] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption for fine-grained access control of encrypted data,” in *Proceedings of the 13th ACM conference on Computer and communications security*, pp. 89–98, 2006.

- [47] Fatemi Moghaddam F, Karimi O, Alrashdan M T, “A comparative study of applying real-time encryption in cloud computing environments,” in *IEEE 2nd International Conference on Cloud Networking (CloudNet)*, pp. 185–189, 2013.
- [48] Mazhar Ali, Revathi Dhamotharan, Eraj Khan, Samee U. Khan, Athanasios V. Vasilakos, Keqin Li, Fellow, IEEE, and Albert Y. Zomaya, “Sedasc: Secure data sharing in clouds,” in *IEEE SYSTEMS JOURNAL*, vol. 11, no. 2, pp. 395–404, 2017.
- [49] L. Xu, X. Wu, and X. Zhang, “Cl-pre: A certificateless proxy reencryption scheme for secure data sharing with public cloud,” in *Proceeding of 7th ACM Symposium Information Computing Communication and Security*, pp. 87–88, 2012.
- [50] A. N. Khan, M. M. Kiah, S. A. Madani, M. Ali, and S. Shamshir-band, “Incremental proxy re-encryption scheme for mobile cloud computing environment,” in *International Journal of Super Computing*, vol. 68, no. 2, pp. 624 – 651, 2014.
- [51] Y. Chen and W. Tzeng, “Efficient and provably-secure group key management scheme using key derivation,” in *IEEE 11th International Conference on Trust-Com*, pp. 295 – 302, 2012.
- [52] R. A. Sana Belguith, Abderrazak Jemai, “Enhancing data security in cloud computing using a lightweight cryptographic algorithm,” in *IEEE ICAS 2015 : The Eleventh International Conference on Autonomic and Autonomous Systems*, 2015.

- [53] A. Juels and B.S. Kaliski Jr., “Pors: Proofs of retrievability for large files,” in *14th ACM Conference on Computer and Communications Security*, pp. 584 – 597, 2007.
- [54] G. Ateniese et al., “Provable data possession at untrusted stores,” in *Proceeding 14th ACM Conference on Computing Communication and Security*, pp. 598 – 609, 2007.
- [55] G. Ateniese, R. Di Pietro, L. V. Mancini, and G. Tsudik, “Scalable and efficient provable data possession,” in *Proceeding of 4th International Conference on Security, Privacy, Communication Networks*, p. 9, 2008.
- [56] H. Shacham and B. Waters, “Compact proofs of retrievability,” in *Advances in Cryptology - ASIACRYPT, Heidelberg, Germany: Springer*, vol. 5350, pp. 90 – 107, 2008.
- [57] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, “Pors: Proofs of retrievability for large files,” in *Proceeding of ACM Workshop Cloud Computing, Security*, vol. 5, no. 3, pp. 31 – 42, 2010.
- [58] C. C. Erway, A. Kupci, C. Papamanthou, and R. Tamassia, “Dynamic provable data possession,” in *ACM Transactions on Information System and Security*, vol. 17, no. 4, p. 15, 2015.
- [59] S.-T. Shen and W.-G. Tzeng, “Delegable provable data possession for remote data in the clouds,” in *Proceeding of ICICS*, pp. 93 – 111, 2011.

- [60] Z. Mo, Y. Zhou, S. Chen, and C. Xu, “Enabling non-repudiable data possession verification in cloud storage systems,” in *Proceeding of IEEE 7th International Conference on Cloud Computing (CLOUD)*, pp. 232 – 239, 2014.
- [61] Y. Ren, J. Shen, J. Wang, and L. Fang, “Analysis of delegable and proxy provable data possession for cloud storage,” in *Proc. 10th IEEE International Conference on Intelligence Information Hiding Multimedia Signal Process(IIH-MSP)*, pp. 779 – 782, 2014.
- [62] Y. Ren, J. Shen, J. Wang, J. Han, and S. Lee, “Mutual verifiable provable data auditing in public cloud storage,” in *International Journal of Internet Technologies*, vol. 16, no. 2, pp. 317 – 323, 2015.
- [63] J. Zhang, P. Li, and M. Xu, “On the security of an mutual verifiable provable data auditing in public cloud storage,” in *International Journal of Networks Security*, vol. 19, no. 4, pp. 605 – 612, 2017.
- [64] Tsu Yang Wu, Yuk-Min Tseng, Sen-Shan Huang, and Y-Chen Lai, “Non-repudiable provable data possession scheme with designated verifier in cloud storage systems,” in *IEEE Journal of Access*, vol. 5, pp. 19333–19341, 2017.
- [65] A. F. Barsoum and M. A. Hasan, “Provable possession and replication of data over cloud servers,” *Centre For Applied Cryptographic Research, Report 2010/32*, <http://www.cacr.math.uwaterloo.ca/techreports/2010/cacr2010-32.pdf>, 2010.

- [66] Erway, C. Chris and Küpçü, Alptekin and Papamanthou, Charalampos and Tamassia, Roberto, “Dynamic provable data possession,” *ACM Trans. Inf. Syst. Secur.*, vol. 17, no. 4, pp. 15:1–15:29, 2015.
- [67] Cong Wang, Sherman S.M. Chow, Qian Wang, Kui Ren, and Wenjing Lou, “Privacy-preserving public auditing for secure cloud storage,” in *IEEE Transactions on Parallel and Distributed Computers*, vol. 62, no. 2, 2013.
- [68] Wen Jun Lu, Avinash L. Varna, and Min Wu, “Confidentiality-preserving image search: A comparative study between homomorphic encryption and distance-preserving,” in *IEEE Transactions and content mining*, pp. 125–141, 2014.
- [69] Ning Cao, Cong Wang, Ming Li, Kui Ren, and Wenjing Lou, “Privacy-preserving multi-keyword ranked search over encrypted cloud data,” in *The 30th IEEE Conference on Computer Communications (INFOCOM’11)*, 2011.
- [70] Huaqun Wang, Debiao He, Shaohua Tang, “Identity-based proxy-oriented data uploading and remote data integrity checking in public cloud,” *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 6, pp. 1165 – 1176, 2016.
- [71] Huaqun Wang, “Identity-based distributed provable data possession in multi-cloud storage,” *IEEE Transactions on Service Computing*, 2014.
- [72] Huaqun Wang, “Proxy provable data possession in public clouds,” *IEEE Transactions on Services Computing*, vol. 6, no. 4, pp. 551–559, 2013.
- [73] Yuan Yuan; Fu Xie, “Identity-based proxy signature multiple-file pdp for mobile cloud computing,” *IEEE International Conference on Computational Science*

and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing (EUC), vol. 1, pp. 381 – 387, 2017.

- [74] Qian Wang, Cong Wang, Kui Ren, Wenjing Lou, and Jin Li, “Enabling public auditability and data dynamics for storage security in cloud computing,” *in IEEE Proceedings of the 14th European Symposium on Research in Computer Security (ESORICS’09)*, pp. 355–370, 2009.
- [75] Junbeom Hur, “Improving security and efficiency in attribute-based data sharing,” *in IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 10, pp. 2271–2282, 2013.
- [76] Mohamed E M, Abdelkader H S, El-Etriby S, “Enhanced data security model for cloud computing,” *in IEEE 8th International Conference Informatics and Systems (INFOS)*, pp. 12–17, 2012.
- [77] Dubey A K, Dubey A K, Namdev M, Shrivastava S S, “Cloud-user security based on rsa and md5 algorithm for resource attestation and sharing in java environment,” *in in CSI 6th International Conference on Software Engineering (CONSEG)*, pp. 1–8, 2012.
- [78] Juels, Ari and Kaliski, Jr., Burton S., “Pors: Proofs of retrievability for large files,” *in Proceedings of the 14th ACM Conference on Computer and Communications Security, CCS ’07*, pp. 584–597, ACM, 2007.
- [79] B. Q. W. S. R. H. D. Yujue Wang, Qianhong Wu and J. Hu, “Identity-based data outsourcing with comprehensive auditing in clouds,” *IEEE Transactions*

- on Information Forensics and Security*, vol. 12, no. 4, pp. 940–952, APRIL 2017.
- [80] L. X. J. Y. TAN Shuang, TAN Lin, “An efficient method for checking the integrity of data in the cloud,” *China Communications*, vol. 64, no. 9, pp. 68–81, 2014.
- [81] J. M. Jianhong Zhang, Pengyan Li, “An oriented-group supporting multi-user public auditing for data sharing,” *IEEE International Conference on Smart City/SocialCom/SustainCom together with DataCom 2015 and SC2*, pp. 592 – 599, 2015.
- [82] K. Yang and X. Jia, “An efficient and secure dynamic auditing protocol for data storage in cloud computing,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 9, pp. 1717–1726, 2013.
- [83] L. X. Liu Yang, “An efficient and secure public batch auditing protocol for dynamic cloud storage data,” *IEEE International Computer Symposium*, pp. 671–675, 2016.
- [84] M. Bellare, R. Canetti, and H. Krawczyk, “Keying hash functions for message authentication,” *International Cryptology Conf. Advances in Cryptology (CRYPTO 96)*, pp. 1–15, 1996.
- [85] Boyang Wang, Baochun Li, and Hui Li, “Oruta: Privacy-preserving public auditing for shared data in the cloud,” in *IEEE Transactions on Cloud Computing*, vol. 2, pp. 43–56, 2014.

- [86] Ateniese, Giuseppe, Seny Kamara, et Jonathan Katz, “Proofs of storage from homomorphic identification protocols advances in cryptology-asiacrypt ’09. springer,” in *IEEE Conference on Theory of Cryptography, Springer*, 2009.
- [87] Kevin D. Bowers, Ari Juels, and Alina Oprea, “Proofs of retrievability: Theory and implementation,” *Proceeding CCSW ’09 Proceedings of the 2009 ACM workshop on Cloud computing security*, pp. 43–54, 2009.
- [88] H. R. H. W. Jian Liu, Kun Huang and M. Xian, “Privacy-preserving public auditing for regenerating-code-based cloud storage,” *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 10, no. 7, pp. 1513–1528, 2015.
- [89] D. X. Jingwei Li, Jin Li and Z. Cai, “Secure auditing and deduplicating data in cloud,” *IEEE Transactions on Computers*, vol. 65, no. 8, pp. 2386–2395, 2016.
- [90] Y Zhu, H Hu, GJ Ahn and M Yu, “Cooperative provable data possession for integrity verification in multicloud storage,” *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 12, pp. 2231–2244, 2012.
- [91] Wei Ren, Linchen Yu, Ren Gao, Feng Xiong, “Lightweight and compromise resilient storage outsourcing with distributed secure accessibility in mobile cloud computing,” *Tsinghua Science and Technology*, vol. 16, pp. 520–528, 2011.
- [92] X. M. W. D. LUO Yuchuan, FU Shaojing, “Enable data dynamics for algebraic signatures based remote data possession checking in the cloud storage,” *China Communications*, vol. 64, no. 9, pp. 114–124, November 2014.

- [93] R. Curtmola, O. Khan, R. Burns, and G. Ateniese, “Mr-pdp: multiple-replica provable data possession,” in *Proceedings of the 28th IEEE ICDCS Conference on Communications Security*, pp. 411–420, 2008.
- [94] Yong Yu, Man Ho Au, Giuseppe Ateniese, Xinyi Huang, Willy Susilo, Yuan-shun Dai, and Geyong Min, “Identity-based remote data integrity checking with perfect data privacy preserving for cloud storage,” *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY*, vol. 12, no. 4, pp. 767–778, APRIL 2017.
- [95] X. L. H. L. Y. M. Yuan Zhang, Chunxiang Xu and X. Zhang, “Efficient public verification of data integrity for cloud storage systems from indistinguishability obfuscation,” *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 3, pp. 676–688, MARCH 2017.
- [96] Amazon Web Services, <https://aws.amazon.com/products/storage/>.
- [97] Google Drive, <https://www.google.com/drive/>.
- [98] Drop Box, <https://www.dropbox.com/>.
- [99] K. Ren, C. Wang and Q. Wang, “Security challenges for public cloud,” *IEEE Internet Computing*, vol. 16, no. 1, pp. 69–73, 2012.
- [100] D. Song, I. Fischer and U. Shankar, “Cloud data protection for masses,” *IEEE Computing*, vol. 45, no. 1, pp. 39–45, 2012.
- [101] Rivest R., “The md5 message-digest algorithm,” 2014.

- [102] Q. W. Zhengwei Ren, Lina Wang and M. Xu, “Dynamic proofs of retrievability for coded cloud storage systems,” *IEEE Transactions on Services Computing*, 2016.
- [103] S. K. Madria, “Security and risk assessment in the cloud,” *IEEE Computurtes*, pp. 110–113, 2016.
- [104] Prakash G L, Manish Prateek and Inder Singh, “Data verification using block level batch auditing on multi-cloud server,” *International Journal of Network Security(IJNS)*, vol. 20, no. 4, 2018.
- [105] Prakash G L, Manish Prateek and Inder Singh, “Secure public auditing using batch processing for cloud data storage,” *Smart Innovation, Systems and Technologies, Springer Singapore*, vol. 79, pp. 137–147, 2017.
- [106] Prakash G L, Manish Prateek and Inder Singh, “Performance analysis of cloud data verification using md5 and ecdsa method,” *Data Science and Analytics, Springer Singapore*, vol. 799, pp. 616–628, 2018.
- [107] J R Winkler, “Securing the cloud: Cloud computing security techniques and tactics,” *Elsevier Inc., USA*, 2011.
- [108] Tim Mather, Subra Kumaraswamy, and Shahed Latif, “Cloud security and privacy,” *Published by O Reilly Media, Inc.,*, 2009.
- [109] “<http://security.setecs.com>,” *Security Architecture for Cloud Computing Environments White paper*, 2011.

[110] Prakash G L, Manish Prateek and Inder Singh, "Data security algorithms for cloud storage system using cryptographic method," *International Journal of Scientific and Engineering Research(IJSER)*, vol. 5, no. 3, 2014.

List of Publications on this Research Work

1. Refereed International Journals

1. **Prakash G L, Manish Prateek, and Inder Singh**, *Data Security Algorithms for Cloud Storage System using Cryptographic Method*, International Journal of Scientific and Engineering Research, Volume 5, Issue 3, ISSN 2229-5518, March-2014.
2. **Prakash G L, Manish Prateek and Inder Singh**, *Data Encryption and Decryption Algorithms using Key Rotations for Data Security in Cloud System*, In International Journal of Engineering and Computer Science, Volume 3, Issue 4 April-2014.
3. **Prakash G L, Manish Prateek and Inder Singh**, *Data Verification using Block level Batch Auditing on Multi-Cloud Server*, International Journal of Network Security(IJNS), Volume 20, Number 4, 2018.

2. Presented in Refereed International Conference

4. **Prakash G L, Manish Prateek and Inder Singh**, *Efficient Data Security Method to Control Data in Cloud Storage System using Cryptographic Techniques*, in IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), May 09-11, 2014, Jaipur, India.
5. **Prakash G L, Manish Prateek and Inder Singh**, *Data Encryption and Decryption Algorithms using Key Rotations for Data Security in Cloud System*, IEEE International Conference on Signal Propagation and Computer Technology (ICSPCT 2014), Ajmer, July 12-13, 2014.

6. **Prakash G L, Manish Prateek and Inder Singh**, *Secure Public Auditing using Batch Processing for Cloud Data Storage*, 1st International Conference on Smart Systems, Innovations and Computing(SSIC), Springer, Manipal University, Jaipur, April 14 -16 , 2017.
7. **Prakash G L, Manish Prateek and Inder Singh**, *Performance Analysis of Cloud Data Verification using MD5 and ECDSA Method*, in 4th International Conference on Recent Developments in Science, Engineering and Technology (REDSET 2017), GD Goenka University, Springer, October13-14 , 2017.
8. **Prakash G L, Manish Prateek and Inder Singh**, *Data Verification using Block level Batch Auditing on Multi-Cloud Server*, in 3rd IEEE International Conference on Advances in Computing, Communication and Automation (ICACCA 2017) , Tulas Institute, Dehradun, September 15-16, 2017.

3. Published in Book Chapter

9. **Prakash G L, Manish Prateek and Inder Singh**, *Secure Public Auditing using Batch Processing for Cloud Data Storage*, Smart Innovation, Systems and Technologies, Vol. 79, pp 137-148, Springer, Singapore, 2017.
10. **Prakash G L, Manish Prateek and Inder Singh**, *Performance Analysis of Cloud Data Verification using MD5 and ECDSA Method*, Data Science and Analytics, CCIS - 799, pp 616-628, Springer, Singapore, 2018.

Prakash G L

Curriculum Vitae

Education

- 2001–2002 **Masters of Engineering**, *Computer Science and Engineering, Bangalore University, Bangalore, Bangalore Institute of Technology, First Class.*
- 1995–1999 **Bachelor of Engineering**, *Computer Science and Engineering, Bangalore University, Bangalore, University Visveswaraya College of Engineering, First Class.*

Experience

- 2011–Present **Assistant Professor (Selection Grade)**, *University of Petroleum and Energy Studies, School of Computer Science and Engineering, Dehradun.*
- 2006–2010 **Assistant Professor and**, HOD, Department of Computer Science and Engineering, ACE, Bangalore, VTU, Belgum.
- 2004–2005 **Assistant Professor**, , Department of Information Science and Engineering, SJBIT, Bangalore, VTU, Belgum.
- 2002–2003 **Lecturer**, , Department of Computer Science and Engineering, GAT, Bangalore, VTU, Belgum.
- 2001–2002 **Lecturer**, , Department of Computer Science and Engineering, BMSCE, Bangalore, VTU, Belgum.
- 2001–2002 **Project Assistant**, , Supercomputing Engineering and Research Center, IISc, Bangalore.

Resource Person

- 2016 IEEE R10 MINI POCO@Dehradun 11th June 2016, Keynote speaker
- National Workshop on Technical Writing using Latex, Resource person.
- Faculty Development Program on Latex, Resource person

Technical Training attended

- Abhigyan: Industry Immersion Program-2018, IBM, Bangalore, January 2nd, 2018 to January 19, 2018.
- Faculty Enablement Program (FEP) on Cloud Computing, Infosys Campus Connect, Chandigarh. 4th-8th December 2017.
- IBM T3 Training

Publications

Journal Publications

- International Journal
- Prakash G L, Manish Prateek and Inder Singh, *Data Verification using Block level Batch Auditing on Multi-Cloud Server*, International Journal of Network Security(IJNS), Volume 20, Number 4, 2018.
 - Prakash G L, Manish Prateek and Inder Singh, *Graph Structured Data Security using Trusted Third Party Query Process in Cloud Computing*, International Journal of Computer Network and Information Security (IJCNIS), ISSN: 2074-9090, March, 2015.
 - Prakash G L, Manish Prateek, and Inder Singh, *Data Security Algorithms for Cloud Storage System using Cryptographic Method*, International Journal of Scientific and Engineering Research, Volume 5, Issue 3, ISSN 2229-5518, March-2014.
 - Prakash G L, Manish Prateek and Inder Singh, *Data Encryption and Decryption Algorithms using Key Rotations for Data Security in Cloud System*, In International Journal of Engineering and Computer Science, Volume 3 Issue 4 April 2014.
 - Prakash G L, Samson Saju, Snehil Mitra and Vedant Sharma, *Optimal Decision Support System Using Multilayer Neural Networks for Incinerator Control*, International Journal of Computer Science and Information Technologies, Vol. 5 (6) , 2014, 6920-6925

Conference Publications

- International Conference
- Prakash G L, Manish Prateek and Inder Singh, Performance Analysis of Cloud Data Verification using MD5 and ECDSA Method, 4th International Conference on Recent Developments in Science, Engineering and Technology (REDSET 2017), Springer, The School of Engineering at GD Goenka University, Gurgaon, India, 13, 14 october 2017
 - Prakash G L, Manish Prateek and Inder Singh, 3rd IEEE International Conference on Recent Developments in Science, Engineering and Technology (2017), Tulas , Dehradun, 2017.
 - Prakash G L, International Conference on Advancements in Science & Technology, Conference, 20th April 2017 to 21st April 2017, India , Mohali 2017.
 - Prakash G L, Manish Prateek and Inder Singh, Secure Public Auditing using Batch Processing for Cloud Data Storage, 1st INTERNATIONAL CONFERENCE ON SMART SYSTEMS, INNOVATIONS AND COMPUTING (SSIC), Springer, 2017.
 - Prakash GL, Nishant Chauhan, Anmol Bhardwaj, Performance Analysis of Various Encryption Algorithms in Multi-Cloud Architecture 2017,International Conference on Advancements in Science & Technology, Conference, 20th April 2017 to 21st April 2017,India , Mohali.
 - Prakash G L, Manish Prateek and Inder Singh, Efficient Data Security Method to Control Data in Cloud Storage System using Cryptographic Techniques, IEEE International Conference on Recent Advances and Innovations in Engineering (ICRAIE-2014), May 09-11, 2014, Jaipur India.
 - Prakash G L, Manish Prateek and Inder Singh, Data Encryption and Decryption Algorithms using Key Rotations for Data Security in Cloud System, IEEE International Conference on Signal Propagation and Computer Technology (ICSPCT 2014), Ajmer, July 12-13, 2014
 - Prakash GI, Sambasivarao K, Priyanka Kirsali and Vibhuti Singh, Short Term Load Forecasting for Uttarakhand using Neural Network and Time Series models, 3rd IEEE INTERNATIONAL CONFERENCE ON Reliability, Infocom Technologies and Optimization (ICRITO 2014), DOI: 10.1109/ICRITO.2014.7014667
 - Prakash G L, Samson Saju, Snehil Mitra, Vedant Sharma, "Neural Network based decision support system for optimal incinerator control", ISSPIT, 2014, 2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2014 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT) 2014, pp. 205-208, doi:10.1109/ISSPIT.2014.7300607
 - Prakash G L, Thejaswini M, S H Manjula, K R Venugopal, L M Patnaik, Efficient Data Aggregation using Query Processing in Sensor Networks ,International Journal on Information Processing, vol 2. No. 4.,pp. 1-14 January 2008.
 - Prakash G L, Prathibha A B, Vikram K, Yashaswini K, Rekha M S, Tejaswi V, Thriveni J, K R Venugopal, L M Patnaik, "EEAPA: Energy Efficient Adaptive Precision Allocation for Data Aggregation in Wireless Sensor Networks", Fourth International Conference on Information Processing (ICIP - 2011), Bangalore, pp. 259-257, 2011.
 - Prakash G L, Thejaswini M, S H Manjula, K R Venugopal, L M Patnaik, "Energy Efficient In-Network Processing in Sensor Networks", ICT 2008 : "International Conference on Information and Communication Technologies" WCSET 2008: World Congress on Science, Engineering and Technology, Bangkok, Thailand December 17-19, 2008

- International Conference
- Prakash G L, Thejaswini M, S H Manjula, K R Venugopal, L M Patnaik, "Energy Efficient Data Processing in Sensor Networks", International Conference on Computer Science: ICCS'09, Hong Kong, 18-20 March, 2009.
 - Thriveni J, Prakash G L, K R Venugopal, L M Patnaik, "Maximizing Network Lifetime in MANET using Average Energy Flooding", International Journal on Information Processing, vol 2. No. 1., January 2008.
 - Thriveni J, Prakash G L, K R Venugopal, L M Patnaik, "Reliable Delivery with Varying Data Rate in Ad Hoc Wireless Networks", International Journal on Information Processing, vol 2. No. 2., April 2008.
 - Thriveni J, Alekhya V L, Deepa N, Uma B, Alice A, Prakash G L, K R Venugopal and L M Patnaik, "Preemptive Routing with Bandwidth Estimation for Enhanced QoS in Ad Hoc Networks", International Conference on Information Processing, Bangalore(ICIP 2008), August 8-10 2008.
 - Thriveni J, Ashwini B, Latha A, Sandhyashree K R, Prakash G L, Alice A, K R Venugopal, L M Patnaik, "Dual Covered Broadcast with Negative Acknowledgements for Enhanced Reliability in Mobile Ad Hoc Networks", IEEE, TENCON 2008, Hyderabad, November 18-21 2008.
 - Thriveni J, Alekhya V L, Deepa N, Uma B, Alice A, Prakash G L, K R Venugopal, L M Patnaik, "QoS Preemptive Routing with Bandwidth Estimation for Improved Performance in Ad Hoc Networks", INDICON 2008, IIT Kanpur, December 11-13 2008.
 - Thriveni J, Ashwini B, Latha A, Sandhyashree K R, Prakash G L, K R Venugopal, L M Patnaik, "Enhanced Double Covered Broadcast with Negative Acknowledgments for Improved Reliability in MANET", IFIP Wireless Days Conference 2008, UAE, November 24 - 27 2008.

Workshop Attended

2017

- Cisco Networking Academy Conference, UPES India, 20/9/2017.
- 1-Day Workshop on Free and Open Source Software (FOSS) & LaTeX, ICFAI University Dehradun, 18 February, 2017.
- 2-Days National Workshop on IOT (Internet of Things) scheduled on 31st March 2017-1st April 2017 at Galgotias University organized by the School of Computing Science and Engineering in association with In Association With (Automobile Club) IIT- Delhi.
- 1-Day workshop on Research Paper writing and Intellectual Property Right, Tula's Institute, Dehradun is organizing a on 25th March 2017

2016

- Work shop on DATA SCIENCE, 17/10/16, at UPES, Dehradun
- NGCT-2016 conference (14,15,16 October 16)
- IIT delhi workshop(23,24 October 16)

Computer skills

Programming Language	C, C++, JAVA, Python, Assembly level programming, Latex.
Operating System	Microsoft Windows, Linux

82, Silver Heights – Dehradun, Uttarakhand 248007

☎ 8979108446 • 📠 8979108446 • 📠 8979108446 • ✉ gprakash78@gmail.com 4/5

Advanced Computer Hardware and Support

— Languages

Kannada **Mothertongue**

English **Intermediate**

Hindi **Basic**

— Interests

- Cooking

- Running

- Reading