

LIGHT-WEIGHT ACCOUNTABLE PRIVACY  
PRESERVING PROTOCOL IN CLOUD COMPUTING  
BASED ON A THIRD-PARTY AUDITOR

Mohamed Ben Haj Frej

Under the Supervision of Dr. Julius Dichter

DISSERTATION  
SUBMITTED IN PARTIAL FULFILMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF DOCTOR OF PHILOSOPHY IN COMPUTER SCIENCE  
AND ENGINEERING  
THE SCHOOL OF ENGINEERING  
UNIVERSITY OF BRIDGEPORT  
CONNECTICUT

November 2019

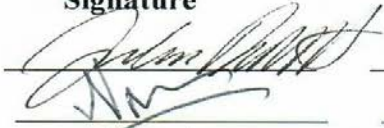
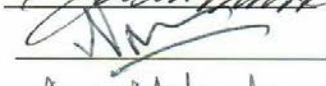



# LIGHT-WEIGHT ACCOUNTABLE PRIVACY PRESERVING PROTOCOL IN CLOUD COMPUTING BASED ON A THIRD-PARTY AUDITOR

Mohamed Ben Haj Frej

Under the Supervision of Dr. Julius Dichter

## Approvals

### Committee Members

Name	Signature	Date
Dr. Julius Dichter		1-22-2020
Dr. Navarun Gupta (co-Advisor)		1/31/20
Dr. Ausif Mahmood		1-22-2020
Dr. Jeongkyu Lee		1-31-2020
Dr. Syed S. Rizvi		12-17-2019

### Ph.D. Program Coordinator

Dr. Khaled M. Elleithy

 2/4/2020

### Chairman, Computer Science and Engineering Department

Dr. Ausif Mahmood

 1-22-2020

### Dean, School of Engineering

Dr. Tarek M. Sobh

 2/4/2020

# LIGHT-WEIGHT ACCOUNTABLE PRIVACY PRESERVING PROTOCOL IN CLOUD COMPUTING BASED ON A THIRD-PARTY AUDITOR

© Copyright by Mohamed Ben Haj Frej 2019

# LIGHT-WEIGHT ACCOUNTABLE PRIVACY PRESERVING PROTOCOL IN CLOUD COMPUTING BASED ON A THIRD-PARTY AUDITOR

## ABSTRACT

Cloud computing is emerging as the next disruptive utility paradigm [1]. It provides extensive storage capabilities and an environment for application developers through virtual machines. It is also the home of software and databases that are accessible, on-demand. Cloud computing has drastically transformed the way organizations, and individual consumers access and interact with Information Technology. Despite significant advancements in this technology, concerns about security are holding back businesses from fully adopting this promising information technology trend.

Third-party auditors (TPAs) are becoming more common in cloud computing implementations. Hence, involving auditors comes with its issues such as trust and processing overhead. To achieve productive auditing, we need to (1) accomplish efficient auditing without requesting the data location or introducing processing overhead to the cloud client; (2) avoid introducing new security vulnerabilities during the auditing process.

There are various security models for safeguarding the CCs (Cloud Client) data in the cloud. The TPA systematically examines the evidence of compliance with established security criteria in the connection between the CC and the Cloud Service Provider (CSP).

The CSP provides the clients with cloud storage, access to a database coupled with services. Many security models have been elaborated to make the TPA more reliable so that the clients can trust the third-party auditor with their data.

Our study shows that involving a TPA might come with its shortcomings, such as trust concerns, extra overhead, security, and data manipulation breaches; as well as additional processing, which leads to the conclusion that a lightweight and secure protocol is paramount to the solution. As defined in [2] privacy-preserving is making sure that the three cloud stakeholders are not involved in any malicious activities coming from insiders at the CSP level, making sure to remediate to TPA vulnerabilities and that the CC is not deceitfully affecting other clients.

In our survey phase, we have put into perspective the privacy-preserving solutions as they fit the lightweight requirements in terms of processing and communication costs, ending up by choosing the most prominent ones to compare with them our simulation results. In this dissertation, we introduce a novel method that can detect a dishonest TPA: The Light-weight Accountable Privacy-Preserving (LAPP) Protocol. The lightweight characteristic has been proven simulations as the minor impact of our protocol in terms of processing and communication costs. This protocol determines the malicious behavior of the TPA. To validate our proposed protocol's effectiveness, we have conducted simulation experiments by using the GreenCloud simulator. Based on our simulation results, we confirm that our proposed model provides better outcomes as compared to the other known contending methods.

## **DEDICATIONS**

In the name of God, the most Beneficent the most Merciful

This dissertation is written in loving memory of my Dad, Tahar, and my Mom, Latifa, who have always been and will remain role models for generations to come. Through their selfless devotion and sacrifices, they instilled in us the love of God and guided us, through cherished memories of dedication, to fully engage in any task that we tackled, neglected and otherwise.

To my wife Khaoula, for her patience with me throughout this journey and all of her encouragement and understanding. To my daughters, Khaoula, Khouchoua, and Khachia, as this whole idea was intended to raise the bar for them.

To my brothers, Khaled, Zoubeir, Naceur, and Youssef, and my sister Kalthoum, as I am so blessed to have them as close family members and am so proud of them.

To my mother-in-law, Najoua, who has been to me as a second mom. To the memory of my late father-in-law, Ali. To my brothers and sisters-in-law Moez, Wafa, Raja, Oumaima, and Ahmed, as well as to my extended family and friends.

## **ACKNOWLEDGEMENTS**

I am so thankful to my advisor, Dr. Julius Dichter, and my co-advisor, Dr. Navarun Gupta, for all their help and support.

I am so grateful to Dr. Khaled Elleithy and Dr. Tarek Sobh for their dedication to the Engineering program and their hard work, encouraging the smooth developments and successes in the department.

Many thanks to the distinguished committee members, Dr. Ausif Mahmood, Dr. Jeongkyu Lee, and Dr. Syed S. Rizvi, for their constructive feedback, recommendations, and ideas, which have enhanced my work.

I am deeply indebted to my dear friend Dr. Abdul Razaque, a graduate from the University of Bridgeport, for his unconditional help and support.

A special thanks to my friends Vignesh Mandalapa Bhoopathy, and Hai Van Nguyen for all of their assistance.

# ACRONYMS

AAA	Authentication, Authorization, and Accounting
AES	Advanced Encryption Standards
CAIQ	Consensus Assessments Initiative Questionnaire
CAS	Cloud Storage Servers
CA	Certificate Authority
CC	Cloud Client
CIA	Confidentiality, Integrity, and Confidentiality
CPS	Cyber Physical System
CPSS	Cyber Physical Social System
CS	Cloud System
CSA	Cloud Service Alliance
CSP	Cloud Service Provider
CSS	Cloud Storage Servers
ECDH	Elliptical Curve Diffie-Hellman
IaaS	Infrastructure as a Service
HLA	Homomorphic Linear Authenticator
KGC	Key Generation System
KP-ABE	Key Policy Attribute-Based Encryption
LAPP	Light-weight Accountable Privacy Preserving Protocol
MIM	Man in the Middle (attacks)
MSCC	Multi-serve Secure Cloud Computing
P2P	Peer to Peer
PaaS	Platform as a Service
PASS	Privacy by Authenticating and Secret Sharing
PNL	Privacy Negotiation language



PoOR	Proof of Ownership and Retrievability
PoR	Proof of Retrievability
PPA	Panda Public Auditing
PPM	Privacy Preserving Model
PPPAS	Privacy Preserving Public Auditing
PRE	Proxy-Re-Encryption
RSA	Rivest, Shamir, and Adelman
RSASS	RSA based Storage Security
SaaS	Software as a Service
SEPPPA	Secure and Efficient Privacy Preserving Public Auditing
SCC	Secure Cloud Computing
SCLPV	Secure Certificateless Private Verification
SCV	Security Controls Validation
SHA	Secure Hash Algorithm
SLA	Service Level Agreement
SOA	Service Oriented Architecture
SPS	Security and Privacy for Storage
TPA	Third-Party Auditor
TPSD	Top-level Security Domains
TSAS	Third-party Storage Audit Service
VM	Virtual Machine

# TABLE OF CONTENTS

ABSTRACT.....	iv
DEDICATIONS.....	vi
ACKNOWLEDGEMENTS.....	vii
ACRONYMS.....	viii
TABLE OF CONTENTS.....	x
LIST OF TABLES.....	xii
LIST OF FIGURES.....	xiii
CHAPTER 1: INTRODUCTION.....	1
1.2 Research Problem and Scope.....	6
1.3 Motivation Behind the Research.....	7
1.3.1 Importance of Cloud Computing.....	7
1.3.2 Importance of security and trust between CSP and CC.....	8
1.4 Potential Contribution of the Proposed Research:.....	8
CHAPTER 2: LITERATURE SURVEY – SECURITY IN CLOUD COMPUTING BASED ON A TPA.....	11
2.1 Classification of the Security Methods Based on a TPA.....	11
2.2 Taxonomy of the Surveyed Methods.....	12
2.2.1 Based on Privacy-Preserving Model (PPM).....	12
2.2.2 Cloud Data Integrity Using a Designated Public Verifier:.....	16
2.2.3 Based on Elaborated Key Exchange Algorithm.....	18
2.2.4 Based on Proof of Retrievability.....	21
2.2.5 Based on Erasure Correcting Code.....	23
2.2.6 Feedback Based Audit Scheme.....	25
2.2.7 Based on Oruta and Knox Approach.....	26
2.2.8 Based on Bi-Linearity Property:.....	27
2.2.9 Based on the Consensus Assessments Initiative Questionnaire (CAIQ).....	27
2.2.10 Based on Encryption and Secret Key.....	28

2.2.11 Based on a Centralized Approach .....	29
2.3 Recapitulation of the Surveyed Methods .....	29
2.4 Discussions and Recommendations on the Studied Methods .....	35
2.4.1 Comparison factors:.....	35
2.4.2 Issues Recurring from Adopting TPA:.....	36
CHAPTER 3: PROPOSED SOLUTION.....	38
3.1 System Model.....	38
3.2 Proposed LAPP Protocol – Mathematical Model .....	41
CHAPTER 4: ALGORITHMS .....	56
4.1 Key Validation process to avoid the malicious role of TPA.....	56
4. 2 The key-extraction process of three stakeholders .....	57
4. 3 Detecting the malicious activity of TPA and CSP .....	59
CHAPTER 5: SIMULATION SETUP AND EXPERIMENTAL RESULTS .....	61
5.1 Simulation Parameters.....	61
5.2 Simulations.....	62
5.2.1 Communication Cost Vs. Block Size .....	63
5.2.2. Auditing Time per Task Vs. Fraction of Invalid Responses .....	65
5.2.3 Reliable Auditing Detection Vs. No. of Cloud Auditing Users .....	67
5.2.4 Computation Time on Auditing Vs. the No. of Challenged Data Blocks .....	69
5.2.5 Accuracy Vs. the Number of Malicious Attempts .....	72
5.2.6 Time Complexity Vs. Input Files .....	74
5.2.7 Time Vs. Total Number of Data.....	76
5.2.8 The Average Auditing Time Vs. the Number of Clients.....	78
5.3 Interpretation of the Results .....	81
CHAPTER 6: CONCLUSION .....	83
REFERENCES .....	85

## LIST OF TABLES

Table 2.1 Recapitulation and Classification Table .....	30
Table 2.2. Recapitulation Based on the Key Schemes.....	35
Table 3.1 Naming Convention .....	54
Table 5.1 Simulation Parameters .....	61

## LIST OF FIGURES

Figure 1.1. Security Threats.....	3
Figure 1.2. Security Requirements, Vulnerabilities, and Threats .....	4
Figure 1.3. Cloud Computing Based on TPA Diagram .....	9
Figure 2.1. TPA Classification based on studied methods .....	11
Figure 2.2. TPA Schemes .....	37
Figure 3.1. System Model.....	39
Figure 3.2. Mathematical Model Diagram.....	55
Figure 5.1. Communication Cost Vs. Block Size (0% Ma).....	63
Figure 5.2. Communication Cost Vs. Block Size (1% Ma).....	63
Figure 5.3. Communication Cost Vs. Block Size (2% Ma).....	64
Figure 5.4. Communication Cost Vs. Block Size (5% Ma).....	64
Figure 5.5. Auditing Time per Task Vs. the Fraction of Invalid Responses (0% Ma) .....	65
Figure 5.6. Auditing Time per Task Vs. the Fraction of Invalid Responses (1% Ma) .....	65
Figure 5.7. Auditing Time per Task Vs. the Fraction of Invalid Responses (2% Ma) .....	66
Figure 5.8. Auditing Time per Task Vs. the Fraction of Invalid Responses (5% Ma) .....	66
Figure 5.9. Reliable Auditing Detection Vs. No. of Cloud Auditing Users (0% Ma) .....	67
Figure 5.10 Reliable Auditing Detection Vs. No. of Cloud Auditing Users (1% Ma) .....	68
Figure 5.11 Reliable Auditing Detection Vs. No. of Cloud Auditing Users (2% Ma) .....	68
Figure 5.12 Reliable Auditing Detection Vs. No. of Cloud Auditing Users (5% Ma) .....	69
Figure 5.13. Computation time on Auditing (No. of Challenged Blocks) (0% Ma) .....	70

Figure 5.14. Computation time on Auditing (No. of Challenged Blocks) (1% Ma) .....	70
Figure 5.15. Computation time on Auditing (No. of Challenged Blocks) (2% Ma) .....	71
Figure 5.16. Computation time on Auditing (No. of Challenged Blocks) (5% Ma) .....	71
Figure 5.17. Accuracy (Number of Malicious Attempts) (0% Ma).....	72
Figure 5.18. Accuracy (Number of Malicious Attempts) (1% Ma).....	72
Figure 5.19. Accuracy (Number of Malicious Attempts) (2% Ma).....	73
Figure 5.20. Accuracy (Number of Malicious Attempts) (5% Ma).....	73
Figure 5.21. Time Complexity (Input Files) (0% Ma).....	74
Figure 5.22. Time Complexity (Input Files) (1% Ma).....	75
Figure 5.23. Time Complexity (Input Files) (2% Ma).....	75
Figure 5.24. Time Complexity (Input Files) (5% Ma).....	76
Figure 5.25. Time (Total Number of Data) (0% Ma) .....	77
Figure 5.26. Time (Total Number of Data) (1% Ma) .....	77
Figure 5.27. Time (Total Number of Data) (2% Ma) .....	78
Figure 5.28. Time (Total Number of Data) (5% Ma) .....	78
Figure 5.29. The Average Auditing Time (Number of Clients) (0% Ma) .....	79
Figure 5.30. The Average Auditing Time (Number of Clients) (1% Ma) .....	79
Figure 5.31. The Average Auditing Time (Number of Clients) (2% Ma) .....	80
Figure 5.32. The Average Auditing Time (Number of Clients) (5% Ma) .....	80

## CHAPTER 1: INTRODUCTION

The term cloud refers to the storing of data anywhere and accessing it anytime. Only the users who have sufficient and required permissions can access the stored data. There are many characteristics associated with the cloud, as defined in [3].

We can also define cloud computing as accessing data and utilizing the required applications from remote servers instead of storing or having data locally, which could require a considerable amount of storage and resources. Thus, many companies, organizations, and anyone who is possessing a substantial amount of data, that needs a scalable environment, can store it in the cloud and can access it from anywhere. Cloud computing is Internet-based computing.

Networking threats have been modeled in the CIA (Confidentiality, Integrity, and Confidentiality) - AAA (Authentication, Authorization, and Accounting) security framework. Authentication consists of making sure that the communicating parties are who they pretend they are, allowing to avoid the man in the middle attacks (MIM). The authorization consists of making sure that the communicating parties are authorized to access the resources. As for Accounting, is the about holding every networking stakeholder accountable for their acts and keeping tracks actions in security logs.

The CIA scheme will be detailed below:

- Confidentiality and privacy:

Confidentiality refers to limit access to protected data to only authorized parties. The

threat of having data compromised increases in the cloud. Due to the increased number of parties, devices, and applications involved resulting from the increased number of points of access [4]. Data confidentiality in the cloud is correlated to user authentication. Protecting a user's account from theft is an instance of a more significant problem of controlling access to objects, including memory, devices, software, etc. [5].

- Integrity:

By integrity, we mean that our assets (hardware and software) coupled with data, can only be modified by authorized parties and in approved ways. Data Integrity [6] refers to protecting data from unauthorized deletion, modification, or fabrication [7].

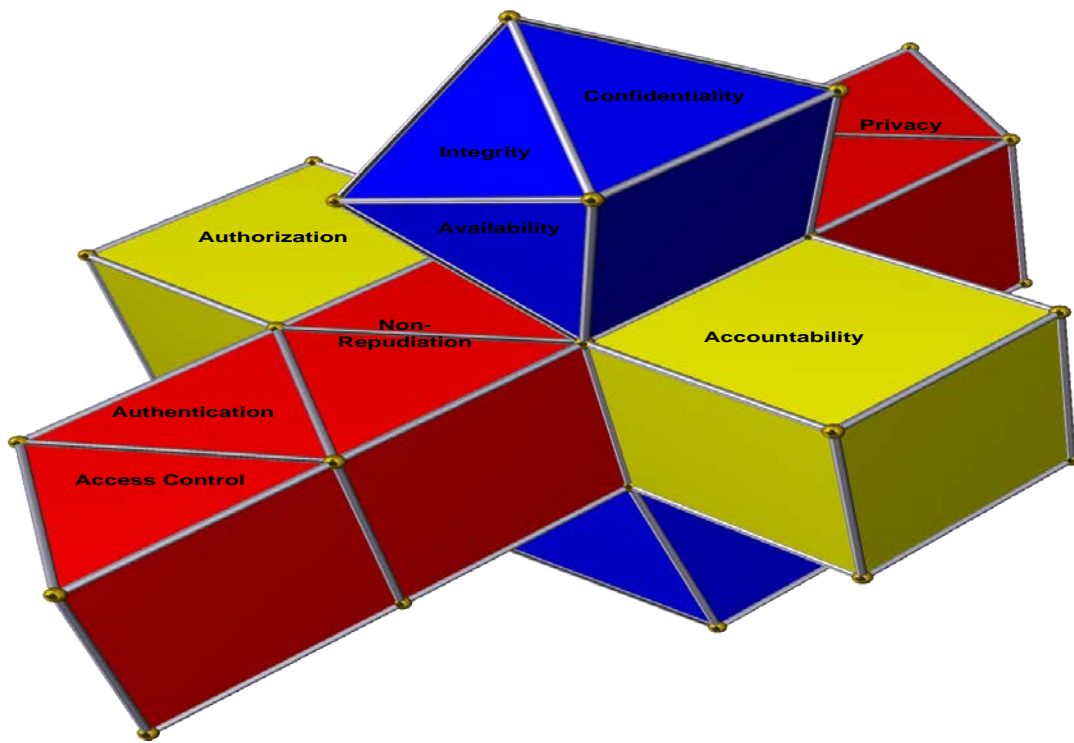
- Availability:

Availability refers to the property of a system being accessible and usable upon demand by an authorized entity. System availability includes a system's ability to carry on operations even when some authorities misbehave. The system must be able to continue operations even in the possibility of a security breach [8].

## **1.1 Cloud Vulnerabilities**

The data on the cloud should be encrypted, which could lead to processing overhead. Considering the benefits of cloud computing, various organizations are moving towards cloud-based IT solutions. However, before starting the journey to the cloud, adopters must consider the possible vulnerabilities (Figure 1) that may convert their dreams of enhancing scalability and saving management costs into a nightmare of either data loss and misuse [6].

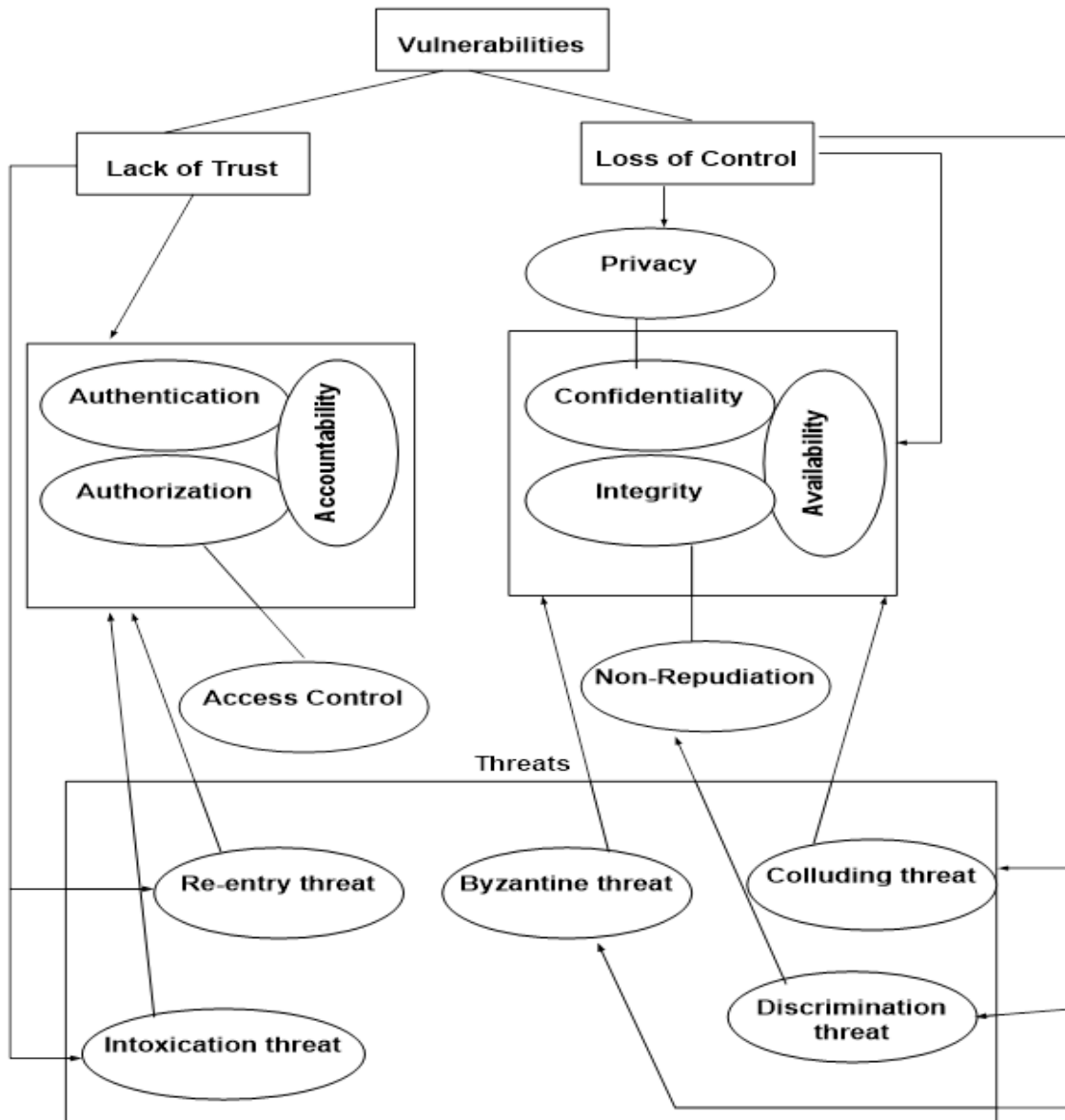




*Figure 1.1. Security Threats*

Hence, users must consider the risks involved with cloud adoption. In the case of third-party management models, most security problems stem from [9]: loss of control, lack of trust (mechanisms), and multi-tenancy.

Gartner's seven security issues that cloud clients should take into consideration could be summarized [10] as privileged user access, regulatory compliance, data location, data segregation, recovery, investigative support, and long-term viability (Figure 1.1).



*Figure 1.2. Security Requirements, Vulnerabilities, and Threats*

*a. Access Control:*

In Figure 1.2, we are summarizing the security requirements, the vulnerabilities as well as the threats in the cloud computing realm. In a multi-tenancy environment, interoperability defects could result in breaches into control access which affects authentication and identification. As for availability, if the service or data in the cloud is not available, then it complicates the retrieval of

data. Policy integration refers to the case of the heterogeneous cloud where different cloud servers may have different mechanisms, making security breaches more likely. Information on public clouds is susceptible to data brokers and hackers due to multiple tenancies due to eavesdropping software such as Easter eggs [11].

*b. Collusion Attacks:*

Consists of an attack from malicious cloud users who use feedbacks to manipulate trust model results. It is called a Collusive malicious feedback attack [12]. It consists of three types:

- Self-Promoting: consists of promoting a cloud service provider. Malicious cloud users enter significant positive feedback.
- Slandering: to defame a cloud service provider. Malicious cloud users enter Significant negative feedback [13].
- Occasional Collusion Feedback attack: in this case, the user occasionally enters essential positive or negative feedback. It takes time to identify them.

*c. Sybil Attacks:*

A Sybil attack is an attack from cloud users using multiple identities to manipulate test results [12]. A malicious cloud user is producing various fake ratings using a minimal value of product purchase in less time. We can classify Sybil attacks in three categories:

- Self-Promoting: also known as a ballot stuffing attack; where the cloud user adds significant positive feedback to promote a CSP.
- Slandering: it is also known as a bad-mouthing attack. In this case, the cloud user adds significant negative feedback to defame a CSP.

- Occasional Sybil Feedback attack: the user occasionally enters essential positive or negative feedback. We can only identify them within time. In the article, Noor T.H et al. [12] proposed a credibility model to detect collusion and Sybil attacks.

*d. On - Off Attack or intoxication Attack*

Malicious users behave alternatively in good or bad ways [14]. In other words, the user first acts in the right direction; then after a while, the user starts to misbehave after earning trust. These types of users are difficult to identify. This vulnerability is also called the dynamic personality of peers in the p2p network system [14].

*e. Discrimination Attacks*

When a CSP delivers different qualities of services provided to CCs, it could result in different ratings to these providers and impact their trust. Then the group who offered contradictory results might be labeled as dishonest [15]. To date, there is no practical solution to mitigate such an attack.

*f. Newcomer or reentry Attack*

Consists of the case when the user who previously produced bad behavior reenters with a new identity to attack again [15]. It is called newcomer or reentry Attack. By comparing credential recodes using location, unique id, we can reduce reentry Attacks.

## **1.2 Research Problem and Scope**

The cloud storage is an easy and flexible platform that allows the Cloud Client (CC) to store its confidential data from local computing to the cloud. Nowadays, many CCs store their data in the cloud; however, this platform introduces new concerns and security trials. To overcome these concerns, a Third-Party Auditor (TPA) is added to safeguard the confidential data and restore

CC's confidence. However, since the human factor is introduced, and for the sake of building a foolproof system, we should expect that the TPA could be dishonest. The TPA could share the CC's confidential information to illegitimate parties to gain financially as well as other benefits. Hence, in this dissertation, we introduce a novel model that enables CCs to protect their private data from TPAs: Light-weight Accountable Privacy-Preserving (LAPP) protocol.

## **1.3 Motivation Behind the Research**

The motivation behind this research is twofold:

- The importance of cloud computing
- The importance of security and trust between CSP and CC

### **1.3.1 Importance of Cloud Computing**

Cloud computing is promising to become the next Information Technology (IT) trend, taking in consideration of:

- The importance of the CSPs in the industry (Amazon, Google, Microsoft, Rackspace, etc. as well as the cloud developed by the open-source community.
- The rapid development of technology.

Computing services, such as database transactions, storage, software, computing, and applications, are delivered to local devices through the Internet. Cloud computing allows resolving the under or over-estimation of IT infrastructures' projections, thus avoiding a long wait for servers to be shipped or configured. Or sometimes, to prevent having configured servers sitting idle because the team is busy resolving other issues. They are ideal for environments with seasonal peaks for over underuse of resources as cloud computing is based on the "pay as you use" principle,

and most providers allow the clients to pre-configure how they want to scale their use of cloud resources [16].

### **1.3.2 Importance of security and trust between CSP and CC**

Cloud computing uses many technologies and strategies to protect clients' data. The cloud service providers compete on using the latest security schemes. Nonetheless, there are still many ambiguities, security-wise, that are making many companies skeptical of adopting the cloud concept fully [17].

In cloud computing safety, security and privacy of data are very important. Cloud computing is vastly used in different social, economic, and national areas of our society. It is used in several industries, financial institutions, government offices, educational institutions, etc. So, people are storing critical and sometimes very confidential data through cloud computing. Hence, before adopting cloud computing, proper knowledge about deployment, security, and privacy requirements is required. Many companies, as well as individuals, are still skeptical about cloud computing when considering the security vulnerabilities associated with it. There are yet no specified privacy and security protection laws explicitly created for cloud computing [18] [19].

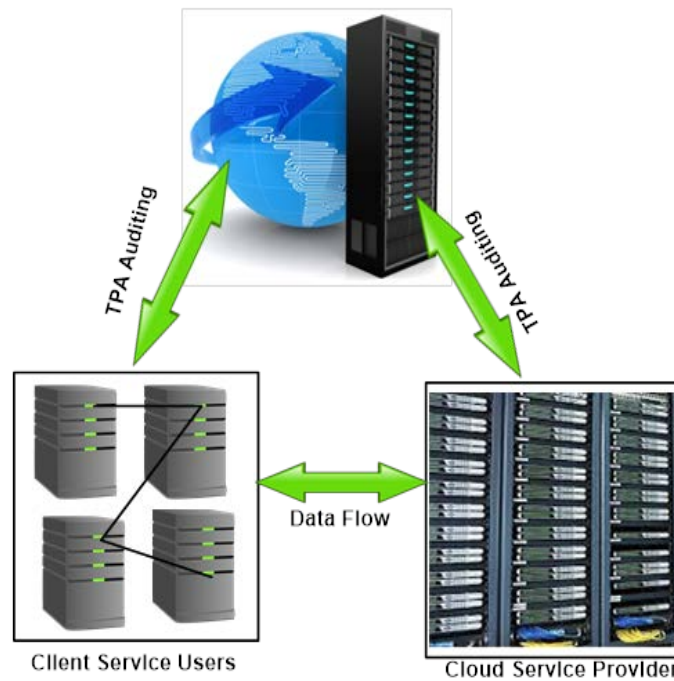
Many researchers are focusing on identifying the security and privacy issues which can be faced by cloud computing customers. Another area of research explores how to select trustable and suitable cloud providers to minimize security and privacy risks [20] [21].

## **1.4 Potential Contribution of the Proposed Research:**

As cloud computing is emerging as the next utility service based on pay as you use model, cloud clients could benefit from the financial savings if they deploy to the cloud correctly. However, security vulnerabilities are the main factor in holding back many companies from even

considering the cloud. Involving a TPA could increase the willingness for adoption. However, it comes with its own set of issues.

The third-party auditor is assumed to be trusted to assess the CSP's storage security upon request from the CC and the provider (Figure 1.3). This scheme gives explicit data support and uses correcting code to provide redundancy in preparation for file distribution.



*Figure 1.3. Cloud Computing Based on TPA Diagram*

Nonetheless, adding a TPA comes with its issues, namely processing overhead (as well as data redundancy) and security trust (tampering with the CC's data). In this dissertation, we are proposing LAPP, a novel security model allowing the CC to audit the auditor, by validating the key presented by the TPA when initiating the audit process, as well as detecting any malicious activity of the TPA and the CSP. It also allows us to ensure that the three stakeholders (CC, CSP, and the TPA) are using the same keys as issued by the trusted party.

The potential contributions of this proposed model are:

- A recapitulation of state of the art for security methods in cloud computing based on a TPA: we have surveyed and classified around hundred and fifty recent papers on cloud security based on a TPA.
- Our proposed LAPP allows the CC to audit the TPA and the CSP for malicious activities, which increases the confidence and the willingness of more companies to embrace the cloud realm.
- On the other hand, once embraced the cloud, LAPP is situated to increase trust and grants more control to the CC to detect issues timely to take practical actions.
  - Ensuring minimum overhead while successfully issuing the secret key to the three stakeholders (CSP, CC, and TPA).
  - Determining and avoiding the malicious role of the TPA, if any, with a lightweight and straightforward algorithm.
  - Enforcing the trust between the TPA and the CC by introducing a malicious-detection algorithm that enables both parties to keep a check and balance on each other.
  - Assuring a more secure communication at a minimum communication-cost.
  - Accurately detecting malicious activities.
  - Improving the Quality-of-Service (QoS) provision, our time complexity simulation results were of a paramount significance in determining this factor.



# CHAPTER 2: LITERATURE SURVEY – SECURITY IN CLOUD

## COMPUTING BASED ON A TPA

### 2.1 Classification of the Security Methods Based on a TPA

Figure 2.1 depicts the classification of the studied methods based on their adopted algorithms.

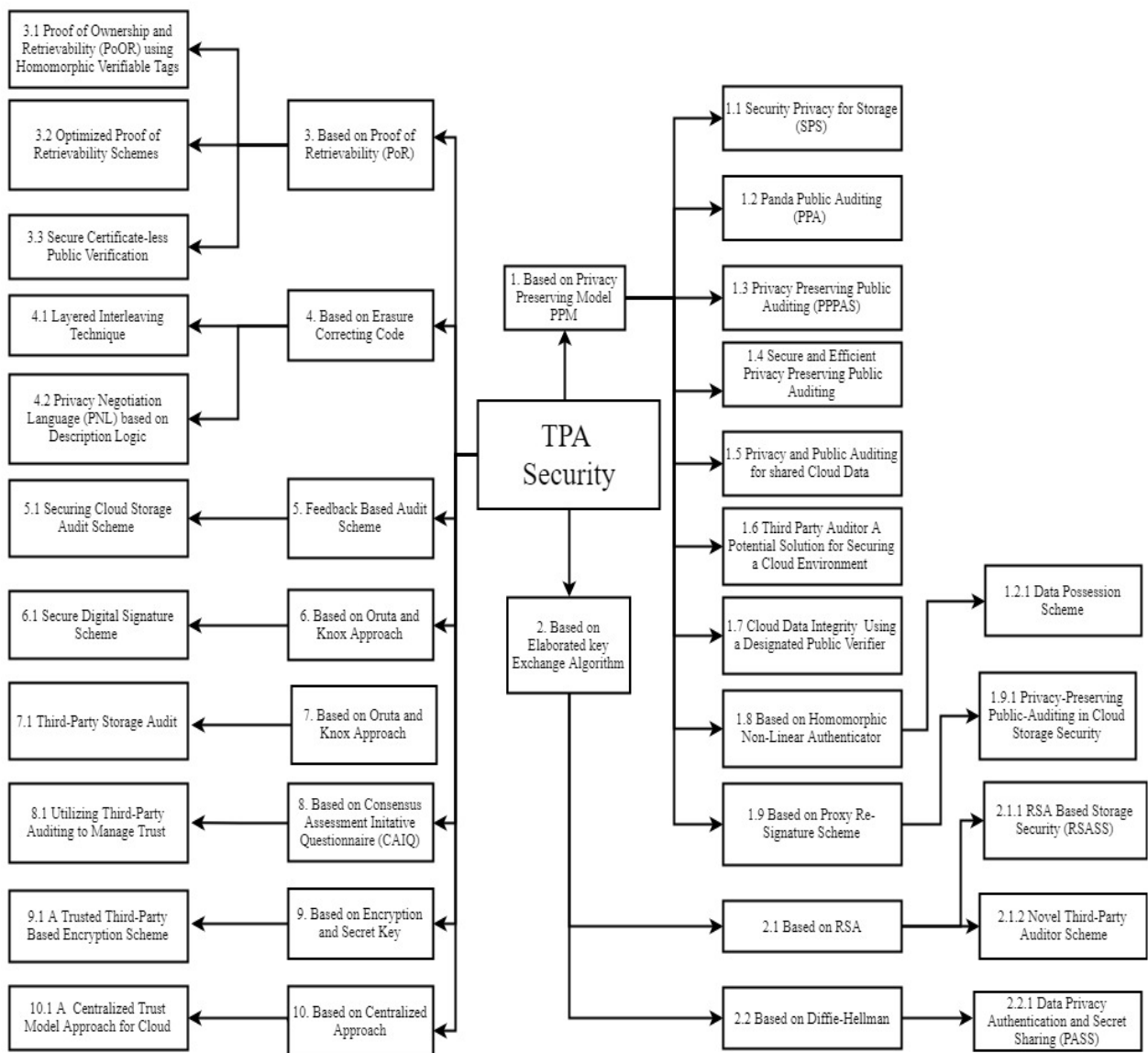


Figure 2.1. TPA Classification based on adopted algorithms

## 2.2 Taxonomy of the Surveyed Methods

### 2.2.1 Based on Privacy-Preserving Model (PPM)

#### 2.2.1.1 Security and Privacy for Storage (SPS):

“Secure Public Auditing Cloud Storage Enabling Data Dynamics in the Standard Model” [22]. SPS is a proposed protocol to audit and protect the data’s integrity using the RSA’s assumptions as a base then extending it to enable a TPA to audit the user’s data without being able to learn its contents. It also allows supporting the operations of data dynamics, such as “insertion,” “modification,” and “deletion.”. In [22], the user has a substantial amount of data to store, the CSP has the means to store all this data at an economical price, and the TPA is proficient in providing auditing that is unbiased and efficient. The authors assume that the CSP is to be untrusted for the reasons discussed before. The TPA’s services are needed to allow the CC to gain trust in the CSPs. The TPA is to be trusted, but it is also assumed that it could be curious, and it could be dangerous if it could learn any of the sensitive information in the data outsourced [23]. The protocol starts with the CC encrypting the data to be outsourced using as a base the strong RSA assumption.

From there, the auditing starts, and it is composed of five algorithms:

- “KeyGen”: used by the CC to create a public key and a secret one. “Outsource”: used by the CC to send the processed data to the CSP. “Audit”: used by the TPA to create an audit query to submit to the CSP.
- “Prove”: used by the CSP after receiving as input the audit query from the TPA and creates a proof by using the data stored.
- “Verify”: used by the TPA to receive the proof from the CSP to check, using the public key, if

the evidence is correct.

The performance analysis has been divided into two parts, the communication, and the computation cost. The communication cost consists of the interaction between the CSP and the TPA. It is based on the proof provided by the CSP to the TPA.

The computation cost for the TPA is determined by the time it takes to audit and verify the data, which is supposed to be fast. For the CSP, it is determined by the time it takes to prove to possess the data, which is determined by the block size, how long the audit query is, and how long it takes to create the information to authenticate[24].

#### *2.2.1.2 PANDA Public Auditing (PPA)*

“Public Auditing for Secure Data Storage in Cloud through a Third Party Auditor Using Modern Ciphertext” [25] proposes a scheme of auditing, using cipher cryptography instead of encryption for the communication with the third-party auditor. The paper focuses on data integrity and storage.

In this method, the TPA performs audits without the need for copies of the outsourced data. The scheme consists of five algorithms.

The CC has to use the new ciphertext to encrypt the data to be outsourced. Then the auditing process begins, using five algorithms:

- "KeyGen," that is used by the CC and the TPA to generate keys.
- "SigGen," that the TPA runs to create the verification metadata.
- "GenProof," that is used by the CSP to check if the data is correctly stored and create a proof of its state; and "VerifyProof" that the TPA uses to test the evidence given by the CSP and

verify its correctness.

During the setup phase, these algorithms are applied as follows, after the CC encrypts the data, it uses “KeyGen” to generate an owner key, and then sends the key and the processed data to the TPA through a private channel. The TPA runs “KeyGen” to create a challenge key and then runs “SigGen” to generate the verification key and then encrypts the processed data to make crypto-metadata to be sent to the CSP.

In the auditing phase, the TPA sends a challenge, using the challenge key, to the CSP. Then using "GenProof" generates an audit key to the TPA, which then uses "VerifyProof" to check if the audit key is equal to the verification key allowing to verify the integrity of the stored data [17].

The system performance analysis is divided into computational, communication, and storage costs. The aim of the computational is to achieve low complexity. Compared with other models with the same scheme, it uses the advanced encryption standard on a bilinear map. In the communication cost, the goal is to have the length of the auditing requests shorter than the length of those in the other schemes based on the bilinear map protocol. Finally, in the storage cost, it is also compared to other methods based on the bilinear map protocol [25].

#### *2.2.1.3 Privacy-Preserving Public Auditing (PPPAS)*

“Privacy-Preserving Public Auditing for Secure Cloud Storage” (PPPAS) [26], is the oldest one of the batch and is alleging to be one of the pioneers to implement public auditing that preserves the data’s privacy in the cloud using the “homomorphic linear authenticator” (HLA) [24, 27]. The HLA is based on keys. This technique allows us to audit without having to use a local copy of the data and incorporating it with arbitrary masking.

This method aims to make the TPA unable to learn the audited data's contents. This scheme uses the same four algorithms as the previous protocol used as well ("KeyGen," "SigGen," "GenProof," and "VerifyProof"). The performance of their auditing is shown to be on par with the state-of-the-art, with a warranty of privacy-preserving [28].

#### *2.2.1.4 Secure and Efficient Privacy-Preserving Public Auditing (SEPPPA) Protocol:*

"Secure and efficient privacy-preserving public auditing scheme for cloud storage"(SEPPPA) [29] declares to have an auditing scheme that has the TPA audit without needing the entire data, maintaining its privacy and integrity, as well as being able to audit by batches. It uses a bilinear map to encrypt the data [30]. This scheme uses four algorithms as well: "KeyGen" that the CC uses to create a pair of keys, one public, available to all the auditing participants, but only authorized TPAs can use it to audit, and a private key for the CC. "SigGen" that creates signatures for all the outsourced files. "ProofGen" is run by the CSP when challenged and uses the data to develop a proof of its integrity. "VerifyProof" is used to check the integrity of the data using the evidence provided by the CSP and the public key, and is run by the TPA [31]. The performance of the scheme was categorized as increasing communication and computation overhead. In the former, the authors explain that data outsourcing, challenge-response auditing, and data retrieval are the main reasons for complexity in the message exchange [32]. It is considered that the outsourcing and retrieval overhead is inevitable, so they focus on the challenge-response, to which they concluded that the system's complexity is constant [33].

#### *2.2.1.5 Privacy-Preserving Public Auditing for shared Cloud Data:*

In this method [34] the integrity of the shared data can be audited by the TPA without the need for the entire data stored in the cloud. The public verifier does not learn the group member's private identity information.

The performance is measured on dynamic groups as well as the public auditing. In active groups, once there is a new user in the group, the private key is shared with him by the original user. Re-signing is done on all the block once the user is revoked from the group; it avoids him to download all the shared data again. In public auditing, the integrity of the information is audited, and the identities of the signers in dynamic groups are preserved. Encryption is done by active broadcast to distribute the private key to the active group members securely. Proxy signatures are needed when the users are revoked, and new users need to be added. In this method, TPA Consumes more time and bandwidth to achieve high error detection probability. The main advantage of this method is dynamic group efficiency is high.

In [35], a bilinear aggregate signature technique is utilized to enable the TPA to handle multiple auditing tasks. This method overcomes the issue of the remote integrity check for data dynamics and simultaneous public auditability. This technique uses the Merkle hash tree algorithm, and batch auditing for multi-client data is done using the BLS signature scheme. This scheme gives a solution that provisions public auditability and data dynamics. There are some steps that are carried out at the server-side and the client-side by the TPA, which are generating the keys and proof checking the keys. By invoking KeyGen, clients' public and private keys are generated. A data file denoted as  $F$  is pre-processed by running SigGen. Homomorphic authenticators and metadata are also produced.

#### *2.2.1.6 Third-Party Auditor: A Potential Solution for Securing a Cloud Environment*

Another scheme is proposed in [36], to detect the malicious insiders in the cloud. It also prevents the number of malicious attempts in the cloud. The performance of this scheme is evaluated based on the successful prevention of malicious access attempts.

### **2.2.2 Cloud Data Integrity Using a Designated Public Verifier:**

The authors use a public verifier in their auditing process to provide data confidentiality and integrity. The system model has three entities: Cloud Service User, Cloud Service Provider, and a designated public verifier [37].

Cloud Service Provider performs computation services based on user's requirements. All the channels perform point to point communication using secure socket layers. It ensures the data privacy of cloud users. This scheme is based on Privacy-Preserving Model, which does the auditing process. The disadvantage of this scheme is when the number of users in the cloud increases, it will affect the TPA's efficiency. This reduction is due while performing auditing there will be an increase in the number of malicious users.

#### *2.2.2.1 Based on Homomorphic Non-linear Authenticator:*

##### *Data Possession Scheme*

The authors developed an attributed based provable data possession scheme to check the data integrity in cloud computing storage, [38]. It utilizes the attributed based signature to construct a homomorphic authenticator. This scheme consists of three different networks which are a client, Cloud Storage Server, and Third-Party Auditor. The Cloud Storage Server used in this scheme is stateless and is verifier independent. In this method, a homomorphic authenticator contains some attributed strategy. The person who satisfies the policy can check data integrity. The delegation key generated by the data owner can fail in subsequent work. The third-party auditor can act as a verifier of the data if it has the public key, and the server cannot be trusted in this case. Clients interact with the servers for accessing the applications. This method consists of cyclic groups which form the signature scheme for the user. The computation becomes hard in bilinear groups, and it is difficult for the adversary to have correctly computed values. The main advantage of this

scheme consists of strong anonymity and sound resistance. It is also able to unlink. The main disadvantage of this method is that the third-party auditor should be trustworthy or else the client's data will be susceptible to compromise [39] [40].

#### *2.2.2.2 Based on Proxy Re-signature Scheme:*

#### *Privacy-Preserving Public Auditing in Cloud Storage Security*

For securing the user's cloud storage such that the TPA cannot learn any information, a method was developed in [41] which uses a homomorphic non-linear authenticator, and the random masking guarantees that the TPA cannot acquire any information while auditing. Random masking is done in non-linear blocks in the server's response because of this; the TPA cannot determine the user's data. This method utilizes short signature scheme [42], which is used for auditing protocol and public auditing. The design goals of this method are public audit, storage consistency, privacy-preserving, batch auditing, and lightweight. The proposed module consists of three algorithms:

- Algorithm 1 is used for token pre-computation.
- Algorithm 2 is used for accuracy, verification, and for locating errors.
- Algorithm 3 is used for error recovery.

They base their method on security consistency for batch auditing, which is needed for storage correctness and preserving privacy.

### **2.2.3 Based on Elaborated Key Exchange Algorithm**

#### *2.2.3.1 Based on RSA*

##### *a. RSA based Storage Security*



The RSASS method consists of two phases [43]: a setup phase and the integrity phase. This scheme consists of continually monitoring and is mainly based on the PDP scheme. Their primary purpose is to achieve storage correctness. It supports dynamic operation and identifies misbehaving servers. It generates a signature that can be used for a file with large and variable sizes. Also, the possession of the records is verified by frequently checking the integrity of the shared data. This method can be incorporated in real-time, and data storage security can be effectively improved [44] [45].

#### *b. Novel third Party Auditor Scheme*

This scheme involves two sections [46], the first section relates to the interaction between the cloud server(s) and the user and the second section refers to the communication between the organization's server and the cloud server [47].

System setup: in this step, the user sends a request to the cloud server to store the data files. The file creates and stores unique keys for the user while at the same time sending keys of itself to the user.

New data file packet: in this step, the user first reformats the data file and encrypts it with a secret key before sending it.

Data file stored: after the user sends the data, the storage server searches for its unique identification in the cloud.

The second section relating to the second phase of interactions between the organization's server and the cloud server involves the following steps:

- System setup: this step relates to the identification between the cloud and the organization servers. Storage servers can then identify each other via this unique identifier in the cloud.
- Keys and other information exchanges: when an information change is made on either set of servers, they should exchange information with the other servers. For example, whenever a cloud server changes a key, it should inform the organization's servers.

#### *2.2.3.2 Based on Diffie-Hellman:*

##### *Data Privacy by Authenticating and Secret Sharing(PASS)*

Protecting data privacy and security can also be done through secret sharing. PASS (data Privacy by Authentication and Secret Sharing) adopts public key cryptosystem that increases the transmission cost and will not store the secret key. Only if the client device is compromised, the secret key may be compromised. Secure Cloud Computing (SCC) is designed using Elliptical Curve Diffie-Hellman (ECDH) and symmetric bivariate polynomial based secret sharing to mitigate this issue, [48] [49].

- Elliptic Curve Diffie-Hellman (ECDH): this is key protocol based on the elliptic curve discrete algorithm. ECHD is faster than the multiplicative group algorithm.
- Symmetric bivariate polynomial based sharing: there are two types of sharing: one is symmetric, and the other is an asymmetric-based sharing. However, the symmetric bivariate based sharing is used to adapt its informative feature of symmetric property to design SCC.

The authors propose two types of SCC. One requires a TPA, and the other does not. Then they could extend this type of SCC to Multi-serve SCC (MSCC). The key agreement protocol without the TPA has three phases: the key sharing phase, the mutual authentication phase, and the key recovery phase. The key agreement protocol with TPA

is the same as without TPA except for the key sharing phase. The main security features that can be achieved using the proposed SCC are mutual authentication between client and server.

On the other hand, the client does not need a complex public cryptosystem to send the share to the cloud server. Even if the client-server and local client devices are compromised, the secret key cannot be obtained. By adding the symmetric property in secret sharing, it reduces the cost to share information between the client and the server [50].

## **2.2.4 Based on Proof of Retrievability**

### *2.2.4.1 Proof of Ownership and Retrievability (PoOR) using homomorphic Verifiable Tags*

The elimination of duplicate information (deduplication) and evidence of information retrievability in cloud capacity under the setting that both customer and cloud servers are not fully trusted, which is an issue that must be addressed. The authors introduced a scheme [51] called Proofs of Ownership and Retrievability (PoOR) [52] to remediate this issue. In this plan, the customers need to prove their ownership of the records they need to transfer without, indeed exchanging the documents to the servers. The cloud computing concept could be compared with Cluster Computing and Grid Computing. In clusters, the resources are grouped into a single administrative domain, and in the grid systems, the resources are geographically distributed across multiple administrative areas. In this scenario, four relevant entities are involved, namely Users/Brokers, SLA Resource Allocator, Virtual Machines (VM's), and Physical Machines. Cloud computation has drawn broad consideration from both academic and industry levels. By

adding a bunch of existing and new strategies like Service-Oriented Architectures (SOA) and virtualization, consisting of:

- Achieving fine-graininess, scalability and data confidentiality
- Empowering the data manager to delegate most computation heavy assignments to cloud servers without client access benefit data
- Securing under standard models

To attain secure, scalable and fine-grained access control on outsourced information in the cloud, the authors combined three cryptographic techniques: Key Policy Attribute-Based Encryption (KP-ABE), Proxy Re-Encryption (PRE) and lazy re-encryption. In KP-ABE, information is related to qualities; for each one of them, they characterize an open key segment [53] [54].

#### *2.2.4.2 Optimized Proof of Retrievability Scheme*

A new scheme called PoOR was proposed [55] with two independent cloud servers. One is used for auditing, and other for storage. They decrease the size of the audit server's capacity. The audit server audits the files remotely stored in the cloud storage by considering the reset attack against the storage during the upload phase. To ensure remote data integrity, an efficient verification scheme that proves secure against reset attacks. The PoOR method supports dynamics and imposes heavy computation overhead at the client-side, new. Hence, the users still must compute all the tags before uploading. All the above techniques do not take reset attacks into account. The construction can reset attacks triggered by the cloud storage server in the upload phase and clients for ensuring the integrity of data storage [56] [57].

The system architecture based on three different entities:

- Client: an entity or organization that has extensive data files to store in the cloud
- CSS (Cloud Storage Servers): an entity managed by CSP, and requires cloud audit server during the integrity check phase
- CAS (Cloud Audit Server): a TPA having specific expertise and capabilities, and is trusted to access services on behalf of the client upon request

#### *2.2.4.3 Secure Certificateless Private Verification (SCLPV):*

This method [58] uses certificate-less verification for the cloud user's storage. Cyber-Physical System (CPS) integrates the cyber world with the physical world, where information is exchanged and transformed. Cyber-Physical Social System (CPSS) has a social entity related to it. For public verification: it uses the PoR (Proof of Retrievability) technique. The TPA can prove that all the verification work successfully. The Key Generation Center (KGC) is controlled by authority and trusted by the users [59].

The security model consists of public certificate-less verification, security, and efficiency. The main advantage of the above method is that a more considerable verification overhead guarantees the protection of the data to prevent malicious auditors. The security threats involved in this method will lead to higher verification costs, and multiple verification tasks may not be adequately performed.

### **2.2.5 Based on Erasure Correcting Code**

#### *2.2.5.1 Layered Interleaving Technique:*

The architecture of the technique will be as follows [60] [61]:

*Third-Party Auditor:*

It should not receive the user's data content through delegated data auditing. The user sends all attributes required for verifying the cloud server in a secure encrypted fashion.

*Cloud Service Provider:*

It contains resources and expertise in building and managing distributed cloud storage servers. It owns, operates, and leases the live cloud computing systems.

*Security Analysis:*

*Step1:* Challenge token creation. When a user stores a file in the cloud, he pre-computes a few verification tokens and distributes them to different servers. Then, each server should make a signature and re-transmit each back to the user to provide him with a handshaking response for the data that the user stored in the cloud storage.

*Step2:* Correctness verification: The response value from the servers not only determines the correctness of distributed storage but also verifies it with the secure server.

*Step3:* Data recovery: in this step, the user checks whether the malicious affected the data on the servers.

*2.2.5.2 Privacy Negotiation Language (PNL) based on Description Logic*

Besides the significant cloud services provided to users, the user's confidential information is at risk. It is of utmost necessary for preserving privacy [62] and ensuring the correctness [63] of the users' data in cloud systems (CS). Hence, few methods were proposed to mitigate these security issues. To negotiate privacy property between the CS and user, Privacy Negotiation Language (PNL) based on description logic was developed. This method can effectively protect the user data

from being misused and illegally propagated by the service provider. This method is proposed to ensure the correctness of users' stored data, and to protect against Byzantine failure, malicious data modification attack, and server colluding attacks. The iterating frequency of algorithm utilized is limited yet provides the correct and valid solution. This method supports the Dynamic Data Operation [64].

There is also the provision of public auditing of stored data in CS. This scheme uses the audit report from the TPA, helps the customers to evaluate the risk of their subscribed cloud data services. And for the CSP, to ensure its functionality and face security challenges.

### **2.2.6 Feedback Based Audit Scheme**

#### *Securing the Cloud Storage Audit Scheme:*

The authors are proposing another scheme that is based on feedback to remediate to the limitations of the Third-Party protocols.

TPAs tend to be semi-trusted or even potentially malicious in some situations. Moreover, a TPA may not always be reliable and independent. It may also collude with the CSP to pass the verification for hiding some CSP's corrupted incident [65]. In this paper, the authors propose a feedback-based audit scheme allowing users to gain trust in the CSP, as well as allowing it to check the integrity of stored data by themselves instead of using the TPA's services [66].

This scheme consists of four phases, which are set up, release plan, execute plan, and review plan.

The TPA has an aggregate-feedback-algorithm, which is required by the user to revoke and invoke it. The following aspects should be considered to establish this feedback-based auditing

model: the user can authenticate whether any TPA has cheated the data owner or indeed executed the designated computational audit task.

The user's data privacy is protected against malicious TPAs. Also, the user could revoke the malicious TPA via the proposed scheme. The proposed model can prohibit the frame and collude attack thoroughly while other existing protocols cannot. So, it is said to be an effective and lightweight protocol where the user himself executes the final verification task. The TPA plays the role of processing proofs and aggregating feedbacks. Processing proofs are required to process the response regarding computing mechanism, and the TPA continuously send the processed data to the server at the receiver side. Running time analysis is done to investigate the number of sampled blocks added to the effect of the audit plan. In this protocol, the user executes the final verification task. So not only relying on the third party gives more reliability but also implements more trust as for solely building on the TPA's services [67].

### **2.2.7 Based on Oruta and Knox Approach**

#### *Secure Digital Signature Scheme:*

The authors discussed The active adversary attacks in three auditing mechanisms for shared data in the cloud, including two identity privacy-preserving auditing mechanisms called Oruta and Knox, and a distributed storage integrity auditing mechanism [68] [69].

It involves the following steps:

- Analysis of Oruta
- Analysis of Knox
- A solution to the security issue

Information in the cloud storage should be protected. Basically, in cloud storage, users store their data using third-party Internet Service Providers (ISPs). The governments used the



third-party doctrine as the legal basis for the government's ease of access to information stored by individuals or businesses contracting with third-party ISPs.

### **2.2.8 Based on Bi-Linearity Property:**

#### *Third-Party Storage Audit Service*

As previously discussed, cloud storage systems' data owners host their data on cloud servers, and this data can be accessed remotely by the data users, resulting in security challenges. The authors elaborated on the need for an efficient and secure dynamic protocol to convince users; resulting in data correctly stored in the cloud. They propose the use of a Third-party Storage Audit Service (TSAS). The TSAS mainly discusses the security challenges that occur in cloud systems [70] [71] [24].

The auditing protocol should have the following properties.

- Confidentiality
- Dynamic auditing
- Batch auditing

And moreover, it should take in consideration the communication and processing costs.

### **2.2.9 Based on the Consensus Assessments Initiative Questionnaire (CAIQ)**

#### *Utilizing Third Party Auditing to manage Trust in the Cloud*

This approach to managing trust in the cloud is based on the Consensus Assessments Initiative Questionnaire (CAIQ) [72], it consists of different security domains. Each has several security controls along with a varying number of controls. A group called Cloud Service Alliance (CSA) designed this questionnaire.

The CSA has a validation process depending upon the response received. At the Top-level Security Domains (TPSD), the validation process is performed. Also, the TPA has many Security Controls Validation (SCV) mechanisms. Mapping is done between the TPSD and the SCV for the auditing process. Based on the cloud services, this method is used to assist the CC in selecting adequate CSP.

#### **2.2.10 Based on Encryption and Secret Key**

##### *A Trusted Third-Party Based Encryption Scheme for Ensuring Data Confidentiality in Cloud Environment*

The main idea of this scheme is to provide active encryption key management to improve the benefits of cloud computing. The CC utilizes a symmetric key protocol for encrypting the data. The TPA maintains a database of secret keys. Shelf protocols are used to provide security to the three entities in the cloud environment.

The TPA module has four steps: Holding the secret key, acquiring the public key certificates, Secret key exchange, and verification of the client's data. By using the encryption scheme, according to the authors, we assure data confidentiality and reduce the computational burden [73].

For the encryption of data in the cloud, the authors use four algorithms in [74] for ensuring cloud data storage security.

AES (Advanced Encryption Standards): It is a single key algorithm. It uses the same key to encrypt and decrypt the data. It has the key sizes of 128, 192, and 256 bits. This algorithm is considered secure. Using the same key reduces the computational load of encryption and decryption. This method has a disadvantage of sharing the key with both the CC and the cloud,

which could result, in this case, in the compromise of the security of the key, which could result in more damage to the system [75].

SHA-1: SHA-1 algorithm has a place with a cryptographic family; it produces a twenty-byte hash. In SHA-1 algorithm message digest length is fixed to 160 Bytes.

This algorithm has high efficiency, but it allows the user to read the data only if one of the keys matches the attributes in the given set of keys [76] [77].

Apart from these standard encryption techniques, there are two user-defined algorithms which are the Correctness Verification and Error Localization algorithm, and the Error Recovery algorithm.

#### **2.2.11 Based on a Centralized Approach**

##### *A Centralized Trust Model Approach for Cloud Computing*

In this approach [78], a trust model is proposed based on a few performance factors. At first, the authors start by studying subjective versus objective trust. Then the role of a third-party auditor is to rate the cloud service providers and give a score to them based on the services they are offering. Also, the feedback of the end-user is needed while providing a rating to the Cloud service provider. This process allows maintaining trust between the cloud user and the cloud provider.

### **2.3 Recapitulation of the Surveyed Methods**

First, we proceed with a recapitulation and a classification of the studied methods as shown in Table 2.1, then a recapitulation based on the key schemes as illustrated in Table 2.2.

*Table 2.1 Recapitulation and Classification Table*

Security Model	Security Requirements	Threats	Advantages
<b>“Secure and efficient privacy-preserving public auditing scheme” (SPS) [22]</b>	<ul style="list-style-type: none"> <li>• Third-party auditing</li> <li>• Supports data dynamics</li> <li>• Supports privacy-preserving public auditing</li> <li>• Use of private channels to relay information</li> </ul>	<ul style="list-style-type: none"> <li>• TPA somehow trusted</li> <li>• Communication overhead</li> </ul>	<ul style="list-style-type: none"> <li>• Practical for cloud systems on large-scale</li> <li>• Considers vulnerabilities of dynamic data</li> </ul>
<b>“Public Auditing for Secure Data Storage in Cloud through a Third-Party Auditor Using Modern Ciphertext” (PPA) [25]</b>	<ul style="list-style-type: none"> <li>• Third-party auditing</li> <li>• Supports data dynamics</li> <li>• Double block transportation</li> <li>• Supports privacy-preserving public auditing</li> </ul>	<ul style="list-style-type: none"> <li>• TPA used as an intermediary to send encrypted data</li> <li>• Hidden Server Failure</li> </ul>	<ul style="list-style-type: none"> <li>• Practical for cloud systems on large-scale</li> <li>• TPA doesn’t need a local copy of data</li> </ul>
<b>“Privacy-Preserving Public Auditing for Secure Cloud Storage” (PPPAS) [26]</b>	<ul style="list-style-type: none"> <li>• Third-Party auditing</li> <li>• Supports batch auditing</li> <li>• Supports privacy-preserving public auditing</li> </ul>	<ul style="list-style-type: none"> <li>• TPA somehow trusted</li> </ul>	<ul style="list-style-type: none"> <li>• TPA doesn’t need a local copy of data</li> <li>• Identification of Invalid Response</li> <li>• Support for Dynamic Data</li> </ul>
<b>“Secure and Efficient Privacy-Preserving Public Auditing” (SEPPPA) [29]</b>	<ul style="list-style-type: none"> <li>• Third-Party auditing</li> <li>• Supports batch auditing</li> <li>• Supports privacy-preserving public auditing</li> </ul>	<ul style="list-style-type: none"> <li>• TPA somehow trusted</li> </ul>	<ul style="list-style-type: none"> <li>• TPA doesn’t need a local copy of data</li> <li>• To date, a pioneer in privacy-preserving schemes for cloud</li> </ul>
<b>Privacy-preserving Public Auditing for Shared Cloud Data [34]</b>	<ul style="list-style-type: none"> <li>• They use the proxy re-signature scheme for outsourcing the updated operations</li> <li>• The common private key is shared between the group’s shared data</li> <li>• Encryption is done by dynamic broadcast; to securely distribute the private key</li> </ul>	<ul style="list-style-type: none"> <li>• TPA consumes more time and bandwidth to achieve high error detection probability</li> </ul>	<ul style="list-style-type: none"> <li>• Highly efficient for progressive groups</li> <li>• Public auditability and data are compelling for the remote data integrity check</li> </ul>

Security Model	Security Requirements	Threats	Advantages
<b>Securing the cloud environment using TPA [36]</b>	<ul style="list-style-type: none"> <li>• An auditing protocol for ensuring the integrity of the third-party auditor using the time-released session keys</li> <li>• It also uses PPM technique</li> <li>• It ensures integrity using time-bounded session keys</li> </ul>	<ul style="list-style-type: none"> <li>• The public verifier is not trusted</li> </ul>	<ul style="list-style-type: none"> <li>• Malicious insiders and attempts are reduced</li> <li>• Data privacy is protected</li> </ul>
<b>Designated public verifier using PPM [37]</b>	<ul style="list-style-type: none"> <li>• Data Security scheme is utilized for the public verifier to audit the data of the cloud user</li> <li>• It uses Privacy-Preserving Model technique</li> <li>• The designated public verifier is a trusted entity like the third-party auditor</li> </ul>	<ul style="list-style-type: none"> <li>• Multiple auditing is not supported</li> </ul>	<ul style="list-style-type: none"> <li>• Efficiency and reliability are greatly improved</li> <li>• Reduced computational burden</li> </ul>
<b>Data Possession Scheme [38]</b>	<ul style="list-style-type: none"> <li>• An attributed based signature is utilized to construct a homomorphic authenticator to check on data integrity</li> <li>• Cloud Storage Server is Stateless and verifier independent</li> <li>• TPA has the public key, and it acts as a verifier</li> </ul>	<ul style="list-style-type: none"> <li>• Cloud Storage Server cannot be trusted</li> <li>• TPA should be trustworthy</li> </ul>	<ul style="list-style-type: none"> <li>• Maintains strong anonymity in the cloud environment</li> <li>• Good resistance</li> </ul>
<b>Privacy-preserving Public Auditing in Cloud Storage Security [41]</b>	<ul style="list-style-type: none"> <li>• Using homomorphic non-linear authenticator and random masking technique</li> <li>• Security consistency is required for batch auditing to secure the correctness of the stored data</li> <li>• Uses the short signature scheme for the auditing protocol and the public auditing</li> </ul>	<ul style="list-style-type: none"> <li>• A local copy of the data can be present in the TPA</li> </ul>	<ul style="list-style-type: none"> <li>• Secures the User's outsourced data in the cloud</li> <li>• TPA achieves better efficiency while performing multiple auditing tasks</li> </ul>
<b>RSA based Storage Security (RSASS) [43]</b>	<ul style="list-style-type: none"> <li>• RSA algorithm is used to generate the signature for handling large data files</li> <li>• Mainly based on Provable Data Possession scheme to achieve storage correctness</li> <li>• Security is constantly maintained</li> <li>• Generates signature which can be used for files of large and different sizes</li> </ul>	<ul style="list-style-type: none"> <li>• TPA has the private key which may be unsafe</li> </ul>	<ul style="list-style-type: none"> <li>• Supports the dynamic operation and identifies misbehaving servers in the cloud.</li> <li>• Dramatically improves data storage security in cloud computing</li> </ul>

Security Model	Security Requirements	Threats	Advantages
<b>Novel Third-Party Auditor Scheme [46]</b>	<ul style="list-style-type: none"> <li>• RSA: used for encryption algorithm and Bilinear Diffie-Hellman: used to secure the keys while exchanging them</li> <li>• Bilinear Diffie-Hellman is the proper method to interchange keys which allows two entities to share secret keys without any prior knowledge</li> </ul>	<ul style="list-style-type: none"> <li>• Data storage security</li> </ul>	<ul style="list-style-type: none"> <li>• Reduce of computing complexity</li> <li>• Assures confidentiality</li> <li>• Authentication is secured</li> <li>• Unauthorized access is restricted</li> </ul>
<b>Data Privacy by Authenticating and Secret Sharing (PASS) [50]</b>	<ul style="list-style-type: none"> <li>• SCC (Secure Cloud Computing) is designed using Elliptical curve Diffie-Hellman and symmetric bivariate polynomial based secret sharing</li> <li>• Two types of SCC: One requiring TPA and the other does not require it</li> </ul>	<ul style="list-style-type: none"> <li>• TPA is assumed, to be honest in SCC.</li> <li>• Cloud Server cannot send the server's share to the client.</li> </ul>	<ul style="list-style-type: none"> <li>• PASS ensures mutual authentication between the client and the server.</li> <li>• Reduce if information cost in secret sharing.</li> <li>• Allows to establish multi-Serve SCC</li> </ul>
<b>Proofs of Ownership and Retrievalability (PoOR) [51]</b>	<ul style="list-style-type: none"> <li>• P0OR uses erasure code, Merkle tree, and homomorphic verifiable tags</li> <li>• Assuring efficiency analysis with the help of data size, computation complexity, size of metadata and communication cost</li> </ul>	<ul style="list-style-type: none"> <li>• Data duplication is a problem which increases data redundancy</li> </ul>	<ul style="list-style-type: none"> <li>• Satisfies the requirements of the cloud environment</li> <li>• Optimized traffic cost</li> <li>• Computation performance is relatively satisfactory</li> </ul>
<b>Proof of Retrievalability Scheme (PoR) [55]</b>	<ul style="list-style-type: none"> <li>• The different entities present in this scheme are the Client, Cloud Storage Server, and Cloud Audit Server</li> <li>• Remotely filed stored are audited by using a cloud server that is independent of the storage server</li> </ul>	<ul style="list-style-type: none"> <li>• Reset attacks occur during the upload phase against storage</li> </ul>	<ul style="list-style-type: none"> <li>• Significantly reduced computation overhead</li> <li>• Both dynamic data operation and public verifiability are supported</li> </ul>

Security Model	Security Requirements	Threats	Advantages
<b>Secure Certificate-less public verification [58]</b>	<ul style="list-style-type: none"> <li>• Uses Proof of Retrievability technique for public verification</li> <li>• Consists on public certificate-less verification, security, and efficiency</li> </ul>	<ul style="list-style-type: none"> <li>• More verification cost is needed</li> <li>• Performs multiple verification tasks</li> </ul>	<ul style="list-style-type: none"> <li>• A malicious auditor user cannot impact the security of SCLPV</li> <li>• Significant verification overhead guarantees the protection of the data</li> </ul>
<b>Layered Interleaving Technique [60]</b>	<ul style="list-style-type: none"> <li>• Erasure correcting code to tolerate multiple failures</li> <li>• TPA delegates the task of verification to save time on the user's side</li> <li>• Based on token challenge verification, correctness verification</li> </ul>	<ul style="list-style-type: none"> <li>• During data auditing, the TPA should not have access to the user's data content</li> </ul>	<ul style="list-style-type: none"> <li>• Highly efficient in recovering the singleton losses</li> <li>• Recovering the bursty data losses</li> </ul>
<b>Privacy Negotiation Language (PNL) Mechanism [62]</b>	<ul style="list-style-type: none"> <li>• The mechanism is based on description logic</li> <li>• They use erasure code in file distribution to guarantee the availability</li> <li>• Public auditing is required for stored data; hence the TPA is used</li> </ul>	<ul style="list-style-type: none"> <li>• Does not guarantee the security of user privacy data</li> </ul>	<ul style="list-style-type: none"> <li>• Protects the user data from being misused</li> <li>• Protects against Byzantine failures by dynamic data operation and server colluding attacks in the cloud</li> </ul>
<b>Securing the cloud storage audit service [65]</b>	<ul style="list-style-type: none"> <li>• Based on feedback audit scheme.</li> <li>• Light-weight protocol and for the computational audit, it adopts multi-TPAs</li> <li>• Three phases: Setup, release and execute</li> <li>• The user completes the final verification task</li> </ul>	<ul style="list-style-type: none"> <li>• Processing proofs are required</li> <li>• Running time analysis should be done</li> </ul>	<ul style="list-style-type: none"> <li>• Prevents frame and colluding attacks</li> </ul>
<b>Secure Digital Signature Scheme [68]</b>	<ul style="list-style-type: none"> <li>• This scheme utilizes Oruta and Knox approach, and the digital signature makes it more secure</li> <li>• Preserves the integrity of the shared data during the auditing process</li> </ul>	<ul style="list-style-type: none"> <li>• An adversary may corrupt the data in the verification phase and prevent the user from using correct data.</li> </ul>	<ul style="list-style-type: none"> <li>• Storage correctness is preserved when the cloud server fails to authenticate its response</li> </ul>

Security Model	Security Requirements	Threats	Advantages
<b>Third-Party Storage Audit Service (TSAS) [70]</b>	<ul style="list-style-type: none"> <li>Combination of cryptography and the bi-linearity properties.</li> <li>The requirements are confidentiality, Dynamic auditing, batch auditing</li> </ul>	<ul style="list-style-type: none"> <li>Auditing protocol becomes insecure due to dynamic operations</li> <li>Replay attack and forge attack occurs</li> </ul>	<ul style="list-style-type: none"> <li>Protects data privacy</li> <li>Less computation costs</li> </ul>
<b>Managing trust using TPA [72]</b>		<ul style="list-style-type: none"> <li>Cloud service user feedback is not supported</li> </ul>	<ul style="list-style-type: none"> <li>Security strength is demonstrated to be effective</li> </ul>
<b>Encryption scheme using TPA [73]</b>	<ul style="list-style-type: none"> <li>Trusted Third-Party based scheme to encrypt the cloud data and algorithms</li> <li>Uses secret key for communication</li> <li>The TPA performs user authentication and data integrity</li> </ul>	<ul style="list-style-type: none"> <li>High communication overhead</li> </ul>	<ul style="list-style-type: none"> <li>Improved Data Confidentiality</li> <li>Reduced Computational burden</li> </ul>
<b>A Centralized trust model approach [78]</b>	<ul style="list-style-type: none"> <li>Based on a centralized model approach</li> <li>Uses feedback mechanism from CC to obtain trust values</li> </ul>	<ul style="list-style-type: none"> <li>Cloud service user feedback cannot always be trusted</li> </ul>	<ul style="list-style-type: none"> <li>Establishes trust for cloud users</li> <li>Updating changes in the server is made easy</li> </ul>



Table 1.2. Recapitulation Based on the Key Schemes

Key Methods/ ALGORITHMS	KEY- GEN	SIG GEN	GEN PROOF	Verify PROOF	HOMOMORPHIC LINEAR AUTHENTICATOR (HLA)	BILINEAR SIGNATURE	SYMMETRIC KEY SCHEME	DATA SECURITY SCHEME	GENERATING SIGNATURE
SPS [22]	✓		✓	✓					
PPA [23]	✓	✓	✓	✓					
PPPAS [48]	✓	✓	✓	✓	✓				
SEPPPA [50]	✓	✓	✓	✓					
DPVPPM [60]								✓	
EPASS [65]	✓	✓			✓	✓			
RSASS [67]									✓
TSAS [92]						✓			
ESTTP[95]							✓		

## 2.4 Discussions and Recommendations on the Studied Methods

In comparing the studied schemes, our study elaborated on the below factors (Figure 2.2):

### 2.4.1 Comparison factors:

*a- Dynamic Auditing:*

As summarized in Figure 2.2, we have developed our research on the following elements:

- RSA based storage security supports active operation and identifies misbehaving servers in the cloud. However, TPA has control over the private keys, which is deemed unsafe.
- Privacy Negotiation Mechanism protects against byzantine failures by dynamic auditing and server colluding attacks. However, it does not guarantee the privacy of the user's data.
- Privacy-preserving public auditing is highly efficient for dynamic groups, but it consumes more time and bandwidth to achieve high error detection probability.
- Third-party storage audit service protects the privacy of the data, and it has less communication cost. In some cases, due to dynamic operations, which tend to make the auditing protocol insecure.

*b- Lightweight security:*

- The novel third-party auditor scheme ensures data confidentiality using encryption techniques leading to secure authentication.
- Data Privacy by authenticating and secret sharing achieves mutual authentication between the client and the server, which reduces the information cost for secret sharing.

*c- Act as a verifier:*

- While protecting data privacy in the public cloud, the verifier is not trusted.
- In Knox and Oruta approach, during the verification phase adversary may corrupt the data.

## **2.4.2 Issues Recurring from Adopting TPA:**

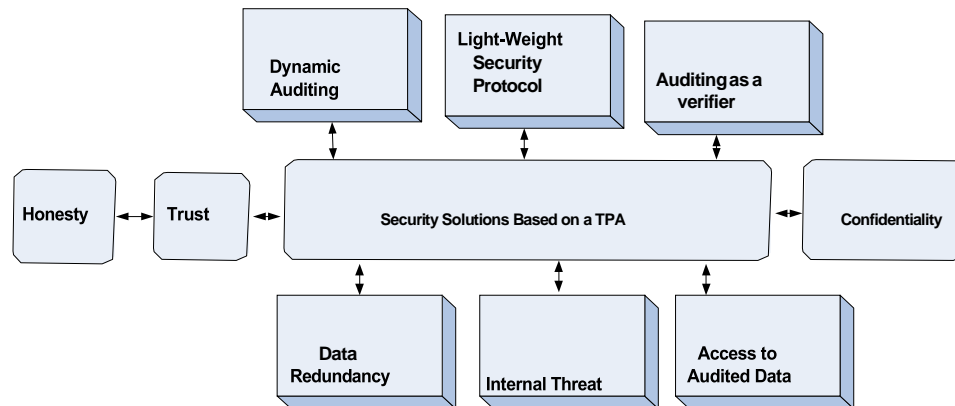
- a- Trust: We could reduce the computation and communication overhead by using the public auditing mechanism. Nonetheless, there can be a problem with internal attacks.

b- *Access to audited data:*

- TPA achieves better efficiency while performing multiple auditing tasks using homomorphic non-linear authenticator and random masking techniques. However, a local copy of the data can be present for the TPA.
- The data possession scheme can be used to maintain strong anonymity in the cloud environment.

c- *Data Redundancy:*

- In PoOR scheme, traffic cost is optimized. However, data duplication is a problem as it leads to data redundancy.
- The centralized trust model approach significantly establishes the trust for cloud users, and makes updating changes easy. However, CC's feedback cannot always be trusted.



*Figure 2.2. TPA Schemes*

As the aim of our study is to orient our efforts towards a more simplistic lightweight, and secure solution that will help alleviate the gap in the cloud users' decision compromise. We will focus on the four most eminent solutions based on privacy-preserving to compare our proposed method to them.

## CHAPTER 3: PROPOSED SOLUTION

The cloud storage is an easy and flexible platform that allows the CC to store his confidential data from his local computers to the servers on the cloud. Nowadays, many CCs store their sensitive data in the cloud; however, this platform introduces new concerns and security trials. To overcome these concerns, the TPA is proposed to safeguard the sensitive data and restore the confidence of the CCs, but it is also expected that the TPA could be dishonest. The TPA could share confidential information with illegitimate parties for the sake of gaining financial benefits. Therefore, the purpose of our proposed method is to introduce a light-weight (in terms of processing and communication cost) solution, that allows the CC to have more control over the auditing process when a TPA is involved.

### 3.1 System Model

In this section, we describe our model's security requirements. The proposed protocol is constructed in a cloud computing environment. To illustrate the actual needs, we start by discussing the system model that helps to audit for the CC and makes the TPA accountable in front of the CC. As depicted in Figure 3.1, the system model supports three entities (TPA, CSP, and CC).



Figure 3.1. System Model

- **Cloud Client:** It is the primary entity in our proposed approach, which is composed of cloud users, devices, and customers. The privacy of the client is of paramount significance along with the utilization of the limited storage and computing resources efficiently. The CC could request the TPA to audit its data while having a lack of trust in the TPA. Hence, the CSP issues a secret key, and a random number to the CC and the TPA explained in the arrows 1a and 1b respectively. Thus, the CC sends a key update request to the TPA for auditing process referred in (2).

Furthermore, the CC wants to confirm whether the TPA possesses a similar secret key and a random number; as the CC sends a message, to TPA presented as an audit request message referred in (4). If the same key and random number are verified, then the TPA is

considered as reliable; Otherwise, the TPA is regarded as illegitimate and sharing the private information of the CC with someone else for personal gains. Thus, the CC wants to ensure that this is not happening.

- **Third-Party Auditor:** It is considered as a highly trusted party. The TPA has a better capability of storage and computation. The performance of the TPAs is examined to safeguard their impartiality and fairness.

Moreover, TPAs use their expertise to audit the data and provide the auditing result as the confirmation or the negation of the deceitful role of the CSP. However, the TPA could be malicious, and the CCs will not have enough trust in him. Thus, the CC wants to ensure that the TPA is not playing a malicious role during the auditing process, as explained in this system model. Once, The TPA receives the auditing request from CC; then, he establishes the audit initialization process with the CSP referred to in the message (3) to complete the auditing process. Furthermore, the TPA is also accountable for updating its key with the CC. In response, the TPA sends the key update response message as referred in (5).

- **Cloud Service Provider:** It involves several distributed servers that offer various services to CCs. The CSP has unlimited resources of computation and storage. The CC enjoys cloud services delivered via the Internet. According to our proposed approach, the CSP is assumed as trustworthy and could be considered as deceitful. Thus, the CSP has been given the responsibility to issue the secret key and random number for both the TPA and the CC. Besides, this method of rendering the secret key and random number helps the CC to check the role of the TPA. The CSP is responsible for forwarding the auditing results to TPA done on it referred in (6). The CSP interacts with the TPA and CCs and initially distributes

its identity ' $CSP_{id}$ ' and the TPA and CCs record this information. In response, the CC and the TPA can use  $CSP_{id}$  to calculate the secret key as follows:

For  $i \in \text{Error! Bookmark not defined.}$ , it calculates two hash values in which  $T_i = L_1(CSP_{id}, i), T_i \in O_1$ . The output gives secret key as:  $S_i = vT_i$ .

### **3.2 Proposed LAPP Protocol – Mathematical Model**

#### **3.2.1. Introduction**

There are several security threats to the cloud computing realm that causes damage to the privacy and confidentiality of the data. Hence, complete satisfaction of the CC is of paramount significance. In this section, we present our proposed LAPP model, which provides the mechanisms for preserving the data-privacy for the CC. This mechanism aims to ensure that the TPA should not expose or steal the CC's private data. Our proposed solution provides that the TPA does not disclose the CCs data contents while auditing the out-sourced data on the cloud servers. This concept is significant as it prevents the issues resulting from involving the TPA in the process, whether they are intentional (TPA bribed by other sources) or unintentional (TPA compromised by attackers) [79]. The cloud storage is an easy and flexible platform that allows the CC to store his confidential data from his local computers to the servers on the cloud. Nowadays, many CCs store their sensitive data in the cloud. However, this platform introduces new concerns and security trials. To overcome these concerns, the TPA is proposed to safeguard the sensitive data and restore the confidence of the CCs, but it is also expected that the TPA could be dishonest. The TPA could share confidential information with illegitimate parties for the sake of gaining financial benefits. Therefore, after a comprehensive survey on cloud computing, based on a TPA, homogeneity, dynamicity, and privacy-preserving, we concluded the need for a light-weight (in terms of

processing and communication cost) solution, that allows the CC to have more control on the auditing process when a TPA is involved.

We assume that Cloud Client (CC) transmits the data file ' $D_f$ ' with data contents ' $D_c$ ' given by:

$$D_f = \{D_{c1}, D_{c2}, D_{c3}, \dots, D_{cn}\} \quad (1)$$

Each data content has its physical significance and can be updated by the CC anytime. The CC can decrypt the private data contents.

The CC encrypts the data using his private key:

$$D_f = P_R \langle D_{c1}, D_{c2}, D_{c3}, \dots, D_{cn} \rangle \quad (2)$$

Therefore, private data contents are encrypted given by

$$D_f = \sum_{i=0}^{T_{pr}} E_{key} \{P_R(D_{ci})\} \quad (3)$$

Where:

$E_{key}$ : Encryption key.

$P_R(D_{ci})$ : Private data.

$T_{pr}$ : complete testing process.

In equation (4), the CC uses its private key to encrypt the data contents. The data contents are further divided into sub-data contents ' $Sd_c$ ' which are denoted as:

$$D_c = \{Sd_{c1}, Sd_{c2}, \dots, Sd_{cn}\} \quad (4)$$



By adopting the same concept and for the sake of making the data more secure, the sub-data contents  $Sd_c$  are further fragmented into smaller chunks  $C(Sd_c)$  explained as:

$$(Sd_c) = C(Sd_{c1}), C(Sd_{c2}), \dots, C(Sd_{cn}) \quad (5)$$

For simplicity's sake, we only focus on single data content in our construct, and a constant number of small chunks for each sub-data content. We chose the maximum number of chunks ' $M[C(Sd_c)]$ ' from the list of small chunks as ' $C(Sd_c)n$ .'

Thus,  $C(Sd_c)n < M[C(Sd_c)]$ . Therefore, each sub-data content  $Sd_c$  has a maximum  $M[C(Sd_c)]$  data chunks, by setting  $(Sd_c)k = 0$  for  $(Sd_c) < k < M[C(Sd_c)]$ .

In our case, we are assuming that the size of each data chunk is persistent and equivalent to the security metric ' $\gamma$ .' Therefore, the total of sub-data contents ' $T(Sd_c)$ ' can be calculated by equation (6):

$$T(Sd_c) = S * D_c / \{(Sd_c) \log \gamma\} \quad (6)$$

Where  $k$ : compatible data chunks and  $S$ : data content size.

Hence, the encrypted data contents are given by:

$$Dc = \{[(Sd_c)k] \mid k \in [1, T(Sd_c)], k \in [1, C(Sd_c)] \mid \quad (7)$$

The multiple encryptions are used that leads to higher security. The significance of using multiple encryptions is to address those problems which mostly doesn't exist visibly but can be resolved using multiple encryptions.

Let bilinear map be used so that  $Z_1, Z_2$ , and  $Z_x$  are the multiplicative groups with identical prime order  $e$  and  $p$ . The bilinear creates the relationship between cryptographic groups.

Thus, the bilinear map can be expressed as  $Z_x \leftarrow Z_1 \times Z_2$  and their generators are  $z_1$  and  $z_2 \in Z_p$ .

$$z_1 \in Z_1.$$

$$z_2 \in Z_2.$$

$$e, p \in \mathbb{Z}_p.$$

$$f: Z_1 \times Z_2 \rightarrow Z_x$$

$$\text{where } f(z_1^e, z_2^p) = f(z_1, z_2)^{ep}$$

Let  $f(h): \{0,1\}^* \rightarrow Z_1$  be the secure hash function that is mapped with data contents  $D_c$  for the point  $Z_1$ .

We are using a hash function with a Boolean output. The purpose is to map the data to secure our system further, as by definition hash functions take in its input data with different sizes and will have an output data with a fixed size.

Our Light-weight Accountable Privacy-Preserving Protocol (LAPP) involves the following procedures:

- Key Generation
- Key Update
- Label Generation
- Testing process
- Substantiation process
- Validation process

### *3.2.1. Key Generation*

Since we have subdivided the data into multiple sub-chunks, and to reassemble the data correctly, we are issuing labels for every sub-chunk created. The key generation process involves the security parameter ' $\beta$ ' that is based on two random numbers  $\gamma_1$  and  $\gamma_2$ ; for the secret hash key (for the data)

and secret label key (for the labels) respectively. It can be expressed as  $\gamma_1, \gamma_2 \in \mathbb{Z}_p$ . Thus, the secret label key output 'Sk<sub>o</sub>' can be determined as:

$$Sk_o = z_2 \gamma^2 \in \mathbb{Z}_2 \quad (8)$$

The CSP performs the key-generation process; And the authentication is initiated by the CC, as described in algorithm 1.

**Algorithm 1:** Key-Generation and authentication process

1. **Initialization:** ( $CS_p$ : Cloud service provider;  $S_k$ : Secret key;  $C_c$ : Cloud Client;  $Rn$ : Random number;  $T_{pa}$ : Third-party auditor;  $C_s$ : Cloud Server;  $K_{req}$ : Key Request;  $Vc$ : Valid Client)
2. **Input:** ( $K_{req}$ )
3. **Output:** ( $S_k, Rn$ )
4. Set  $S_k$  &  $Rn \in C_s$
5.  $C_c$  makes  $K_{req}$
6. Check  $CS_p$  into  $C_s$
7. **If**  $C_c = Vc$  then
8.  $CS_p$  releases  $S_k$  &  $Rn$  to  $C_c$  &  $T_{pa}$
9. **Elseif** Denied  $K_{req}$  to  $C_c$
10. **Endif**
11. **End-Elseif**

Algorithm-1 describes the key-generation process when the CC intends for audit. In this algorithm, the CSP is considered as a reliable and trustworthy entity. In step-1, the initialization of variables is given. In steps 2-3, the input and the output processes are defined respectively. In step-4, the secret key and random numbers are stored for each client on the cloud server that is assigned to the CC based on the given request. Step-5 shows the key request process done by the CC when intending to audit. Once the request is received by the CSP; it starts checking the validity of the CC into the cloud server shown in steps 6-7. If the CC is found as a valid client, the CSP

releases the secret key and a random number to the CC and the TPA as described in step 8. If the CC is not registered with a given cloud server, then the request for obtaining the keys is denied.

### *3.2.2. Key update*

Since each generated key requires updates, without the key update, there is a chance of a compromise on the generated key.

**Theorem 1:** The proposed agreement is evident for CCs.

**Proof:** During the keyGen process, CCs can examine the legitimacy of the key updates with the help of open keys and parameters from the CSP. In other words, after the CSP makes the clients' secret keys up to date, using open keys and parameters, the clients can check if the CSP's secured keys have been refreshed. Based on this, the proposed plan supports the key updates' obviousness for customers.

**Theorem 2:** The Proposed agreement is responsible for the period of key updates.

**Proof:** During the key updates, the hash estimation of the encoded key is built by an absolute counter estimation  $T_i$ . Hence, from the above presentation of key updates, the actual counter is a calculation of the prior encoded key and counter. Therefore, in the process of actual key generation, customers can check the present counter against the past counter. If the current counter is equal or less than the prior counter, it can be inferred that the TPA has refreshed the prior key.

### *3.2.3. Label Generation*

Since we have issued labels for all the data sub-chunks, the label generation is implemented in a way that prevents the injection of invalid data in the system and to mitigate any vulnerability of the system that could be introduced through the label generation. This process involves data contents ' $D_c$ ' that encloses the secret hash key ' $\gamma_1$ ' and the secret label key ' $\gamma_2$ '. As ' $\gamma_1$ ' and ' $\gamma_2$ '

are used as inputs. The small chunks ‘C(Sd<sub>c</sub>)’ require random values, as illustrated in equation (11):

$$C(Sd_c) = \{\tau_1, \tau_2, \tau_3, \dots, \tau_{C(Sd_c)}\} \in Z_p \quad (9)$$

Once the random values are chosen, the computing process ‘C<sub>k</sub>’ is initiated as  $C_k = z_1^{tk} \in Z_1$  for all  $k \in [1, C(Sd_c)]$ . Thus, for each data contents (Sd<sub>c</sub>)<sub>k</sub> ( $k \in [1, C(Sd_c)]$ ), the data label dl<sub>j</sub> is determined by:

$$dl_j = \{f(h)(\gamma_1, d_{id} \parallel j)\} * \prod_{k=1}^{C(Sd_c)} C_k(Sd_c)^k \}^{\gamma_2} \quad (10)$$

Labels are a vital part, and data labels require to be updated automatically. Therefore, the complete set of data labels can be obtained from the data contents by substituting the dl<sub>j</sub>.

$$S(dt_j) = [(dl_j)_j \in \{1, C(Sd_c)\}]$$

$$S(dt_j) = \left\{ (f(h)(\gamma_1, d_{id} \parallel j) * \prod_{k=1}^{C(Sd_c)} C_k^{(Sd_c)^k})^{\gamma_2} \right\} j \in [1, C(Sd_c)] \quad (11)$$

Where j: label number for each data content; d<sub>id</sub>: data identifier; S(dt<sub>j</sub>): complete set of data labels.

### 3.2.4. Testing process

This section describes the first step in the overall validation of our method. The idea is to work on a sample of the data contents D<sub>ci</sub>, then generalize the method to the overall data. Hence, few samples of data contents are chosen to build the testing process ‘T<sub>pr</sub>’ and generate random number ‘Rn.’ Thus, it can be written as  $Rn \in Z_p^*$  for selected data contents (D<sub>c</sub>)  $k \{j \in T_{pr}\}$ . We calculate the testing process sample ‘T<sub>ps</sub>’ as  $T_{ps} = \{T_{pr}\}^{Rn}$  by using a random number. Therefore, the complete testing process  $T_{pr}^*$  can be obtained as:

$$T_{pr}^* = \{(j, Rn), j \in T_{pr}, T_{ps}\} \quad (12)$$

### 3.2.5. Substantiation Process

This phase is called “the substantiation process” as it gives more details of the testing with specifics to the data contents ‘ $\mathcal{T}$ ’ and the generated labels ‘ $\Psi$ .’ This process gets the data contents as inputs and then applies to them the complete testing process  $T_{pr}^*$ . Hence the process encompasses the label substantiation ‘ $\Psi$ ’ and the data content substantiation ‘ $\mathcal{T}$ .’

Thus, the label substantiation can be generated by:

$$\Psi = \prod_{k \in T_{pr}} dl_j^{Rn} \quad (13)$$

The small data chunks of all tested data contents  $\{T(D_c)\}$  are first calculated for generating the tested data content substantiation for each  $k \in [1, C(Sd_c)]$  given by:

$$T(D_c) = \sum_{T(D_c)} Rn * (Sd_c)k \quad (14)$$

The data content substantiation process can be obtained as

$$\mathcal{T} = T(D_c) \times \Psi$$

By substituting the values of all tested data contents and label substantiation to obtain the data content substantiation process by equation (17)

$$\mathcal{T} = \left\{ \sum_{T(D_c)} Rn * (Sd_c)k \right\} \times \prod_{k \in T_{pr}} dl_j^{Rn} \quad (15)$$

### 3.2.6. Validation Process

After applying our testing methodology to the labels and the data contents (substantiation phase), in this phase, we are implementing a method to check whether or not the TPA has tampered the data contents. We proceed by applying the complete testing process  $T_{pr}^*$ ,

data content substantiation process  $\mathbb{T}$ , secret hash key  $\gamma_1$ , the secret label key  $Sk_o$ , a sample of the data contents  $D_{ci}$ , as inputs. The identifier hash function  $f(h_i)$  is calculated first as

$$f(h_i) = \{Tpr^* + D_{ci} + \mathbb{T} + \gamma_1(SK_o)\} \quad (16)$$

Based on the given identifier hash function, we need to determine tested hash function  $f(h_t)$  for all tested data contents and calculation of tested hash  $f(h_t)$  given by:

$$f(h_t) = T(D_c) + f(h_i)$$

Substitution of tested data contents  $T(D_c)$  & the identifier hash function  $f(h_i)$

$$f(h_t) = \left[ \left\{ \sum_{T(D_c)} Rn * (Sd_c)k \right\} + \{(j, Rn), j \in Tpr, Tps\} + \left\{ \sum_{T(D_c)} Rn * (Sd_c)k \right\} + \left\{ \sum_{T(D_c)} Rn * (Sd_c)k \right\} \times \prod_{k \in Tpr} dl_j^{Rn} + \gamma_1(z_2 \gamma_2 \in Z_2) \right] \quad (17)$$

Since the hash function, defined earlier,  $f(h)$  has a Boolean output, the validation process could be determined by

{ Data contents not tampered by TPA    if     $\mathbb{T}. \varepsilon\{f(h_t), Sk_o\} = 1$  }

{ Data contents tampered by TPA        if     $\mathbb{T}. \varepsilon\{f(h_t), Sk_o\} = 0$  }

After determining the tampering process done by TPA, the possible validation process can be obtained by:

$$\begin{aligned}
\mathbb{T} \cdot \varepsilon\{f(h_t), Sk_o\} = & \left\{ \sum_{T(D_c)} Rn * (Sd_c)k \right\} \\
& \times \prod_{k \in T_{pr}} dl_j^{Rn} \left\{ \left[ \left\{ \sum_{T(D_c)} Rn * (Sd_c)k \right\} + \{(j, Rn), j \in T_{pr}, T_{ps}\} + \left\{ \sum_{T(D_c)} Rn * (Sd_c)k \right\} \right. \right. \\
& \left. \left. + \left\{ \sum_{T(D_c)} Rn * (Sd_c)k \right\} \times \prod_{k \in T_{pr}} dl_j^{Rn} + \gamma 1(z_2 \gamma_2 \in Z_2) \right], z_2 \gamma_2 \in Z_2 \right\} \quad (18)
\end{aligned}$$

Where  $\varepsilon$  is a small positive value.

In this section, the validation of the malicious role of the TPA is described in algorithm 2.

Algorithm 2: Validation of the malicious role of TPA

1. **Initialization:**  $\{CS_p$ : Cloud service provider;  $S_k$ : Secret key;  $C_c$ : Cloud Client;  $Rn$ : Random number;  $A_{req}$ : Audit Request;  $T_{pa}$ : Third party auditor;  $C_s$ : Cloud Server;  $K_{ureq}$ : Key Update Request;  $K_{ures}$ : Key Update Response;  $TP_l$ : Third party auditor legitimate;  $TP_i$ : Third party auditor illegitimate}
2. **Input:**  $\{A_{req}\}$
3. **Output:**  $\{TP_l, TP_i\}$
4.  $C_c \rightarrow A_{req}$  to  $T_{pa}$
5. Initiate  $T_{pa} \rightarrow CS_p$
6. Set  $K_{ureq} \rightarrow T_{pa}$
7.  $T_{pa} \rightarrow K_{ures}$  to  $C_c$
8.  $C_c$  Checks if  $K_{ures} = S_k \& Rn$  then
9.  $T_{pa} = TP_l$
10. *Elseif*  $T_{pa} = TP_i$
11. *Endif*
12. *End Elseif*

Algorithm-2 determines the malicious activity of TPA. In step-1, the initialization process is described. Steps 2-3 show the input and output processes, respectively. In steps-4-5, the CC initiates the request for the auditing process to the TPA. Once, the TPA receives the solicitation of auditing; then it starts the auditing process.



On the other hand, the CC does not have trust in the TPA, so it wants to send a key update request to ensure that TPA has similar keys shown in step-6. Once, the TPA receives the key update request; if it is a legitimate TPA, then it sends the key update response to the CC as described in step-7. When the CC receives the key update response from the TPA, then it checks whether the response with a secret key and a random number matches, then the TPA is declared as legitimate; Otherwise, the TPA is considered as illegitimate (please refer to steps 8-10).

#### 4. Privacy-Preserving Polynomial Model Generation

In this section, we use the Chebyshev polynomials to demonstrate the effectiveness of our introduced method (LAPP) mathematically.

Consider a dataset  $\{(x_i, y_i) | 1 \leq i \leq n\}$ , and let

$$\hat{f}(x) = a_1\varphi_1(x) + a_2\varphi_2(x) + \cdots + a_m\varphi_m(x) \quad (19)$$

where,  $a_1, a_2, \dots, a_m$  are coefficients and  $\varphi_1(x), \varphi_2(x), \dots, \varphi_m(x)$  are Chebyshev polynomials of the first kind,

$$\varphi_1(x) = T_0(x) = 1$$

$$\varphi_2(x) = T_1(x) = x$$

$$\varphi_n(x) = T_{n+1}(x) = 2xT_n(x) - T_{n-1}(x)$$

Assume that the data  $\{x_i\}$ , are chosen from an interval  $[\alpha, \beta]$ . The Chebyshev polynomials can be modified as:

$$\varphi_k(x) = T_{k-1}\left(\frac{2x - \alpha - \beta}{\beta - \alpha}\right) \quad (20)$$

The approximated function  $\hat{f}$  of degree  $(m - 1)$  can be given by Equation (21), where the degree of  $(\varphi_k)$  is  $k - 1$ . We will assume the interval  $[\alpha, \beta] = [0, 1]$  and construct the model accordingly. According to Equation (22) when  $[\alpha, \beta] = [0, 1]$ , we get Equation (22)

$$\varphi_k(x) = T_{k-1}\left(\frac{2x - \alpha - \beta}{\beta - \alpha}\right) = T_{k-1}(2x - 1) \quad (21)$$

We can obtain the following equations for  $m = 4$ .

$$\varphi_1(x) = T_0(2x - 1) = 1$$

$$\varphi_2(x) = T_1(2x - 1) = 2x - 1$$

$$\varphi_3(x) = T_2(2x - 1) = 8x^2 - 8x + 1$$

$$\varphi_4(x) = T_3(2x - 1) = 32x^3 - 48x^2 + 18x - 1$$

To determine the  $\hat{f}(x)$  when  $m = 4$ .

$$\hat{f}(x) = a_1\varphi_1(x) + a_2\varphi_2(x) + a_3\varphi_3(x) + a_4\varphi_4(x) \quad (22)$$

$$\hat{f}(x) = a_1(1) + a_2(2x - 1) + a_3(8x^2 - 8x + 1) + a_4(32x^3 - 48x^2 + 18x - 1)$$

Let the actual input be  $y_i$ , where  $i = 1$  to  $n$ . The error of the approximated input can be determined by Equation (25)

$$e_i = \hat{f}(x_i) - y_i \quad (23)$$

We need to determine the values of  $a_1, a_2, a_3$ , and  $a_4$ , we use the root mean square error.

$$E = \sqrt{\frac{1}{n} \sum_{i=1}^n [\hat{f}(x_i) - y_i]^2}$$

Let's take the least-squares fitting of  $\hat{f}(x_i)$  of the class of functions  $C$  which minimizes  $E$  as

$$\widehat{f^*}(x)$$

We can obtain  $\widehat{f^*}(x)$  by minimizing  $E$ . Thus, we seek to minimize  $M(a_1, a_2, a_3, a_4)$  which is given in Equation (26)

$$M(a_1, a_2, a_3, a_4) = \sum_{i=1}^n [a_1 + a_2(2x - 1) + a_3(8x^2 - 8x + 1) + a_4(32x^3 - 48x^2 + 18x - 1) - y_i]^2 \quad (24)$$

The values of  $a_1, a_2, a_3$ , and  $a_4$  that minimize  $M(a_1, a_2, a_3, a_4)$  will satisfy the expressions given from equations 27-30.

$$\frac{\partial M(a_1, a_2, a_3, a_4)}{\partial a_1} = \frac{\partial (\sum_{i=1}^n [a_1 + a_2(2x - 1) + a_3(8x^2 - 8x + 1) + a_4(32x^3 - 48x^2 + 18x - 1) - y_i]^2)}{\partial a_1} = 0 \quad (25)$$

$$\frac{\partial M(a_1, a_2, a_3, a_4)}{\partial a_2} = \frac{\partial (\sum_{i=1}^n [a_1 + a_2(2x - 1) + a_3(8x^2 - 8x + 1) + a_4(32x^3 - 48x^2 + 18x - 1) - y_i]^2)}{\partial a_2} = 0 \quad (26)$$

$$\frac{\partial M(a_1, a_2, a_3, a_4)}{\partial a_3} = \frac{\partial (\sum_{i=1}^n [a_1 + a_2(2x - 1) + a_3(8x^2 - 8x + 1) + a_4(32x^3 - 48x^2 + 18x - 1) - y_i]^2)}{\partial a_3} = 0 \quad (27)$$

$$\frac{\partial M(a_1, a_2, a_3, a_4)}{\partial a_4} = \frac{\partial (\sum_{i=1}^n [a_1 + a_2(2x - 1) + a_3(8x^2 - 8x + 1) + a_4(32x^3 - 48x^2 + 18x - 1) - y_i]^2)}{\partial a_4} = 0 \quad (28)$$

The summary of the used notations is given in Table 3.1.

*Table 3-1 Naming Convention*

Symbol/Function	Details
Df	data file
Dci	data contents
Sdci	sub-data contents
C(Sdci)	smaller chunks of sub-data contents
M[C(Sdc)]	the maximum of data chunks
T(Sdc)	total sub data contents
$\gamma$	security metric
k	compatible data chunks
S	data content size
f(h)	secure hash function
z1 and z2	key generators
$\beta$	security parameter consisting of two random numbers $\gamma_1$ and $\gamma_2$
$\gamma_1$	secret hash key
$\gamma_2$	secret label key
Sk0	secret label key output
$\tau_1$	random number
d <sub>id</sub>	data identifier
dlj	data label
S(dtj)	complete set of data labels
Ck	computing process
A(Dc)	data content abstract information
Tpr*	complete testing process
$\Psi$	the label substantiation
Rn	random number
$\mathcal{T}$	data content substantiation
$\varepsilon$	small positive value

### 3.2.7 Mathematical Model Diagram

Figure 3.2 recapitulates the main steps of our proposed model.

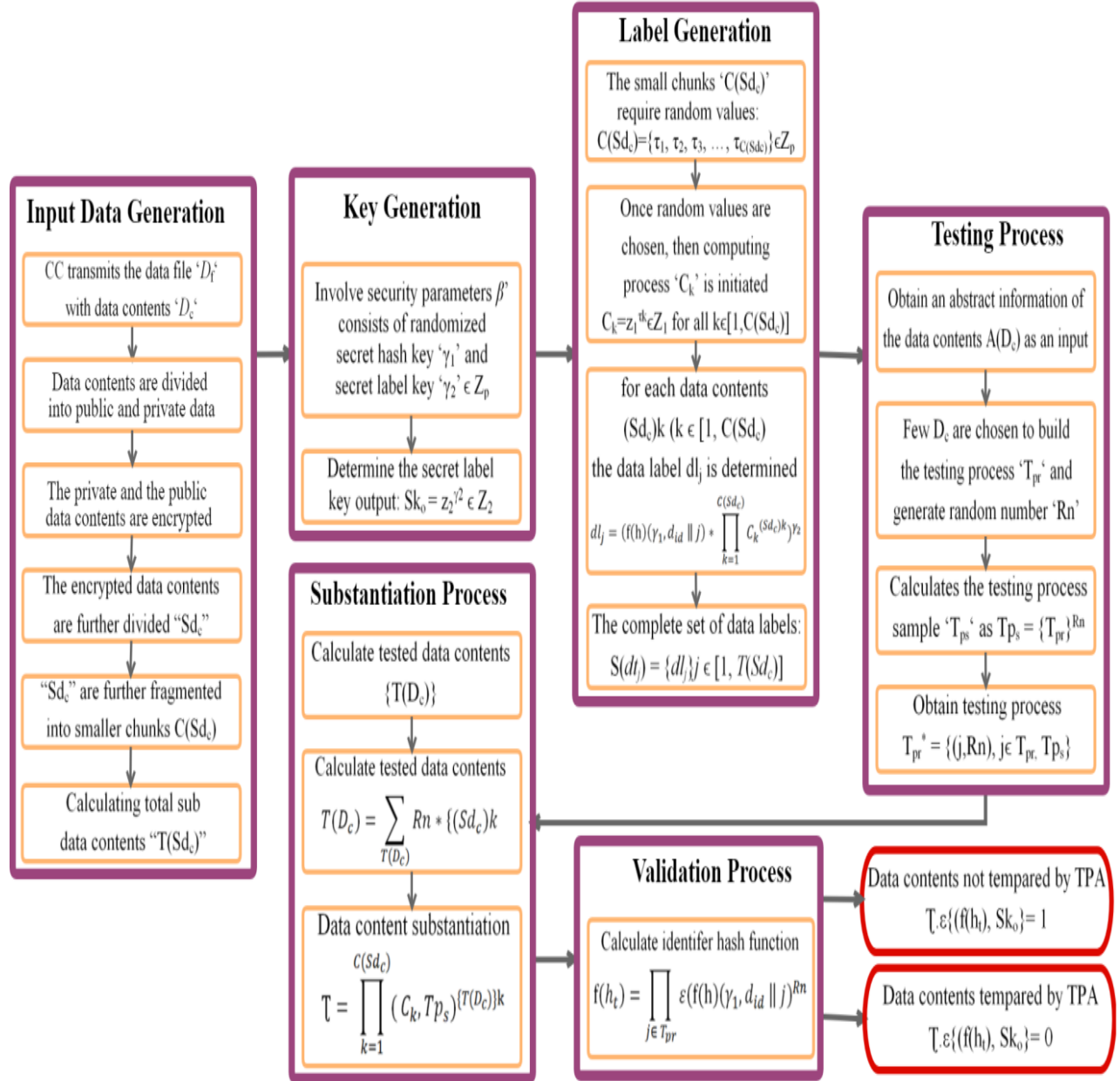


Figure 3.2. Mathematical Model Diagram

## CHAPTER 4: ALGORITHMS

Our proposed solution is based on three main algorithms that will be detailed below:

- Key Validation process to avoid the malicious role of the TPA.
- The key-extraction process of three stakeholders.
- Detecting the malicious activity of third party auditor and Cloud service provider.

Our assumptions are as follows: these proposed algorithms will not serve to replace the existing security algorithms used by the cloud, but rather serve as an extra layer to give additional tools to the CC to be able to have a way to verify on the auditing process.

### 4.1 Key Validation process to avoid the malicious role of TPA

**Algorithm 3:** Key Validation process to prevent the malicious role of the TPA

1. Initialization: ( $g^k$ : Guarantee of the key;  $s_k$ : Secret key;  $A_{test}$ : Auditor testing process;  $A_{res}$ : Service provider response; CC: Cloud Client;  $p$ : prime number,  $R_n$ : Random number, and  $V$ : validation) Input:  $D_c$
2. Input: ( $g^k$ ;  $s_k$ ;  $A_{test}$ ;  $R_n$ ;  $p$ )
3. Output: ( $A_{res}$ ;  $CSP_{res}$ ;  $V$ )
4. Pick  $R_n$  such that  $1 < R_n < p$
5. Compute  $g^{R_n}$  // Group of random numbers
6. CC initiates  $A_{test}$
7. Set  $A_{test} = g^{R_n}$
8. CC computes  $CSP_{res} = g^{R_n.K}$
9.  $CSP_{res} \rightarrow CC: g^{R_n.K}$
10. CC computes  $A_{test} = (g^k)^{R_n}$
11. CC checks if  $A_{test} = g^{R_n.K}$  then
12. CC confirms successful  $V$
13. else if CC determines the malicious activity of TPA
14. End if
15. End else if

The goal of this algorithm is to validate the key presented by the TPA when initiating the audit process:

- From lines 1-3, we explain the initialization, input, and output processes, respectively.
- In line 4, random number is picked that should be greater than 1 and less than a prime number.
- In line 5, we compute the group of random numbers.
- From lines 6-7, Cloud client starts the testing process for the auditor, and we set the group of random numbers for the testing process.
- In line 8, the cloud client computes the service provider's response against a group of the secret keys and random numbers; as a group of a random number including secret keys must be matched with the list of cloud client's random numbers.
- In line 9, the cloud service provider sends its response to the cloud client to confirm the group of secret keys including the random numbers.
- On line 10, cloud client computes the testing process that should be encrypted using secret keys inclusive random numbers.
- In lines 11-13, the cloud client checks if the auditing testing process meets the criteria of the random number and secret keys issued by the cloud service provider are same as third party auditor (TPA) presents for auditing then, cloud client confirms the successful key validation process. If the criteria are not met, then the cloud client determines the existence of a TPA's malicious activity.

## **4. 2 The key-extraction process of three stakeholders**

**Algorithm 4:** Key-Extraction process of three stakeholders

1. Initialization: ( $g^{R_s}$  : Group of Random shared key; CSP: Cloud service provider;  $S_k$ : Secret key;  $B_k$ : Blinded key; CC: Cloud Client;  $p$ : prime number;  $R_s$ : Random shared key; TPA: Third Party Auditor; and  $T$ : trusted party)
2. Input: ( $g^{R_s}$ ; CSP;  $S_k$ ;  $T$ )
3. Output: (TPA;  $B_k$ )
4. CC & CSP use  $R_s$
5. Set  $1 < R_s < p$  && TPA knows  $g^{R_s}$  &&  $g^{R_s} \in T$
6. CSP  $\rightarrow$  TPA :  $B_k = R_s + S_k \bmod p$
7. TPA examines  $g^{B_k}$  if  $g^{R_s} g^{S_k} = g^{R_s + S_k \bmod p}$  then
8. Set  $g^{B_k} = g^{R_s + S_k \bmod p}$
9. Elseif  $g^{R_s} g^{S_k} \neq g^{R_s + S_k \bmod p}$  then
10. Set CSP  $\nexists g^{B_k}$
11. Endif
12. End else-if
13. TPA  $\rightarrow$  CC  $\in B_k$
14. CC computes  $B_k - R_s = S_k \bmod p$  if CC =  $B_k$  then
15. Set TPA = CC
16. endif

The goal of this algorithm is to ensure the three stakeholders (Cloud Client, Cloud Service Provider, and Third-Party Auditor) are using the same keys as issued by the trusted party:

- From lines 1-3, we explain the used parameter-initialization, input, and output processes respectively.
- In line 4, cloud service providers and cloud clients use the secret random shared key.
- In the line 5, we set the value of the secret random shared key more than one and less than the prime number. Furthermore, the trusted party is responsible for issuing the group of random shared keys for the three stakeholders.
- In line 6, the cloud service provider issues the blinded key to the third-party auditor.
- In lines 7-10, third party auditor examines the group of blinded-key if they are same that are already available with the cloud service provider. If the group of secret keys, including random



shared keys, is similar to that of blinded key, then it is proved that both entities (cloud service provider and third-party auditor) are possessing the same keys for authentication. If a group of secret keys including random shared keys is not identical with that of blinded-key, then it is proved that cloud service provider does not have valid keys that should match with a group of blinded keys.

- In line 11-13, Third-party auditor also checks the blinded-key with cloud client. Hence, the cloud client computes the key, and if the key of the cloud client matches the key of the third-party auditor, then it is declared that both parties have the same keys.

#### 4. 3 Detecting the malicious activity of TPA and CSP

**Algorithm 5:** Detecting the malicious activity of third-party auditor and Cloud service provider

1. Initialization: ( $f(h)$ ): Hash function; CSP: Cloud service provider;  $K_{en}$ : Encrypted key;  $D_c$ : Data contents; CC: Cloud client; TPA: Third party auditor;  $O(D_s)$ : Original data service; Ma: Malicious)
2. Input: ( $D_c$ )
3. Output: (Ma)
4. CSP assigns  $K_{en}$  to TPA & CC
5. TPA audits  $\{f(h) K_{en} (D_c)\} \in CC$
6. If  $\{f(h) K_{en} (D_c)\} \in CC = O(D_s)$  then
7. CSP  $\in h$  otherwise Ma
8. Endif
9. CC  $\rightarrow$  TPA:  $\{f(h) K_{en} (D_c)\}$
10. If  $\{f(h) K_{en} (D_c)\} = O(D_s)$  then
11. TPA  $\in h$  otherwise Ma
12. Endif

This algorithm aims to detect the malicious activities of the cloud service provider as well as the third-party auditor as he/she audits the data contents of the cloud client.

This algorithm consists of two phases:

- In the first phase, the TPA audits if the CSP is malicious or not.
- In the second phase, the CC audits whether the TPA is malicious or not.

The algorithm steps are as follows:

- From lines 1-3, we describe the parameter-initialization, input, and output processes, respectively.
- In line 4, the cloud service provider assigns the encrypted key to the third-party auditor and the cloud client. As, the third-party auditor uses this key to audit the encrypted contents of the cloud data user, while the cloud client uses the key to access the cloud and detect the malicious activity of the third-party auditor when auditing the data contents.
- In lines 5-7, the third-party auditor checks the services provided to cloud client from the cloud service provider. The cloud client presents its encrypted data contents to inspect and determine the role of the cloud service provider. If the data contents submitted by the cloud client match with provided service from the cloud service provider, then the cloud service provider is considered as honest otherwise the cloud service provider is regarded as malicious (dishonest).
- From lines 9-11, the role of the third-party auditor has checked whether it is honest or doing a malicious activity with the data contents.

## CHAPTER 5: SIMULATION SETUP AND EXPERIMENTAL RESULTS

To confirm the performance of the proposed Light-weight Accountable Privacy-Preserving (LAPP) protocol, we have developed the LAPP using the C++ programming language and integrated into the GreenCloud simulator. The GreenCloud is run on the IBM z13 to obtain quick and realistic outcomes. We generated several scenarios that were almost identical to the real environment. The used parameters are described in Table 5.1 and are for testing purposes.

### 5.1 Simulation Parameters

*Table 5.1 Simulation Parameters*

Parameters	Details
Number of chassis switches at L4	1920
Line cards at L4	1630
Ports at L4	72
Number of racks at L4	16
Number of chassis switches at L3	432
Line cards at L3	164
Ports at L3	48
Number of racks at L3	128
Used virtual machines	1800
Number of Servers	64
Maximum number of Cloud Service Users	18000
Hosts in each rack	132
Each Host supports	16 processors
Memory with each processor	256 GB
Storage Memory	512 GB

Parameters	Details
Virtual Disk Memory	430 GB
Bandwidth for L4	256 GB/Sec
Bandwidth for L3	128 GB/Sec
Bandwidth for L2	64 GB/Sec
Bandwidth for L1	16 GB/Sec
Queue delay	0.005 Seconds
Burst time	0.0056 Seconds
Idle time	0.0032 Seconds
Packet Size	1260

## 5.2 Simulations

We have elaborated on all simulations with malicious attempts rates at 1%, 2% and 5% malicious TPA activities [80]. All simulation results showed a superiority in performance for our proposed LAPP. In our simulation, we have proceeded with the following measurements:

- Communication Cost vs. Block Size
- Auditing Time per Task vs. Fraction of Invalid Responses
- Reliable Auditing Detection vs. Number of Cloud Auditing Users
- Computation time on Auditing vs. Number of Challenged Data Blocks
- Accuracy vs. Number of Malicious Attempts
- Time complexity vs. Input Files
- Processing time vs. Volume of Data
- Average Auditing Time of each Service User vs. Number of Clients

For each simulation, we will present four graphs corresponding to 0%, 1%, 2% and 5% malicious rates, and we will give the interpretation of the results for the 2% rate ones.

### 5.2.1 Communication Cost Vs. Block Size

In this simulation, as depicted in figures 5.1 – 5.4, we measure the communication cost while increasing the audited data's block size from 0 to 900 KB. At 900 KB, the simulation result shows 830 KB while other methods show a fluctuation between 1100 KB and 1250 KB.

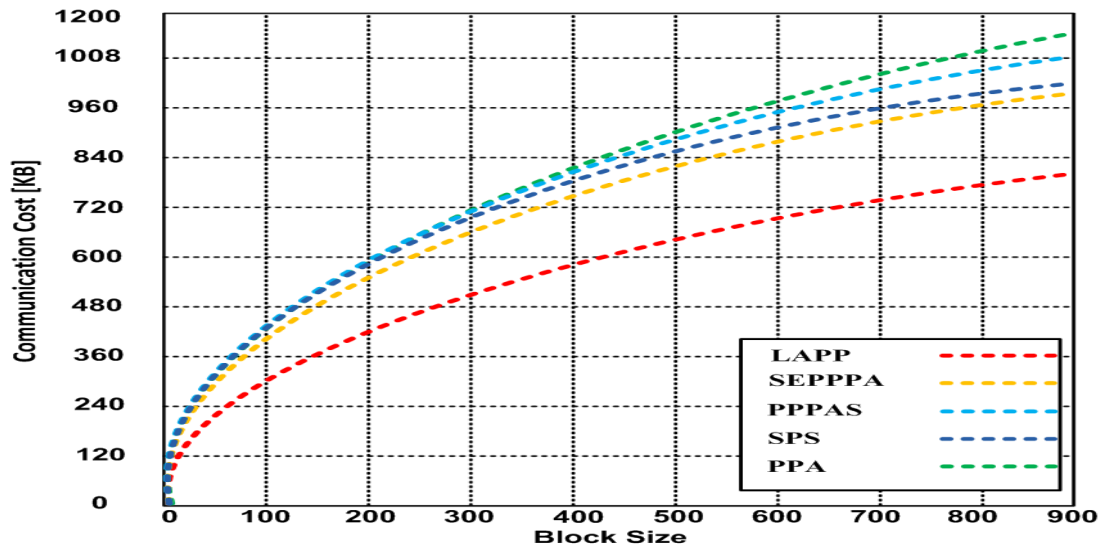


Figure 5.1. Communication Cost Vs. Block Size (0% Ma)

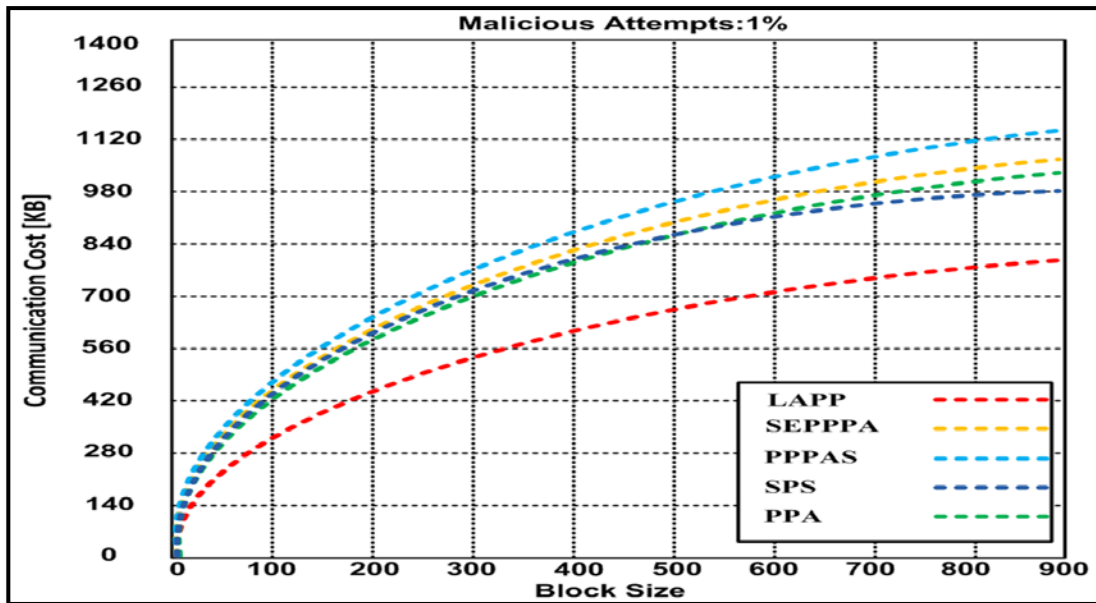


Figure 5.2. Communication Cost Vs. Block Size (1% Ma)

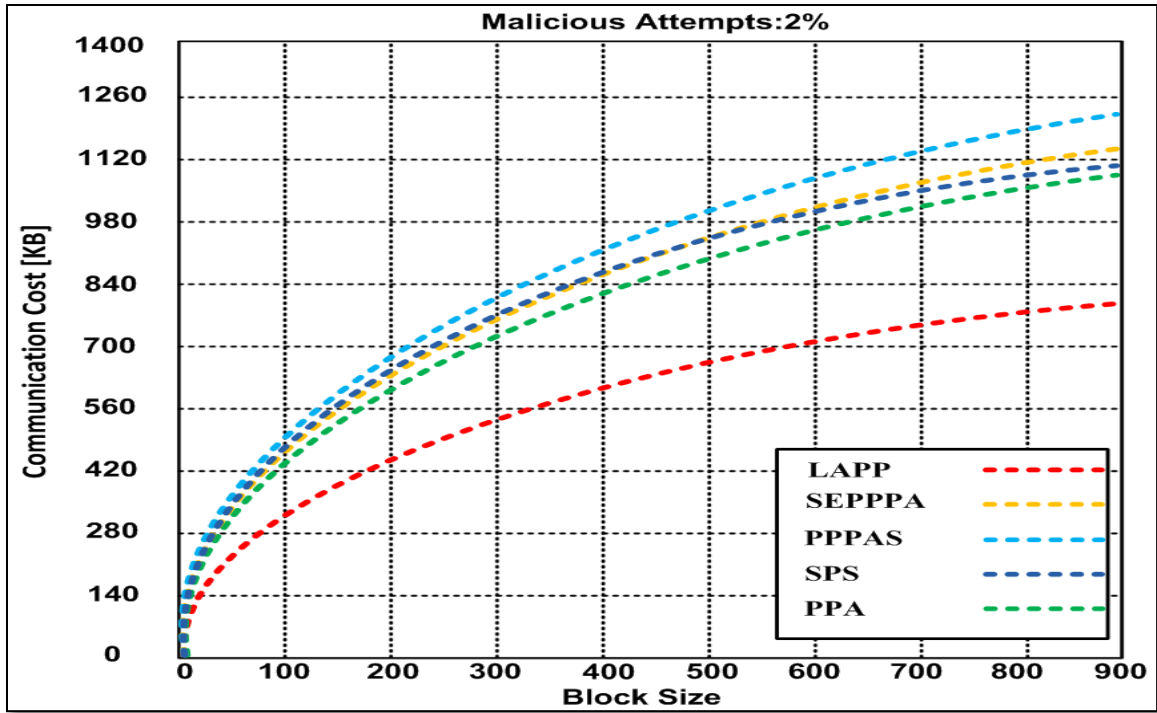


Figure 5.3. Communication Cost Vs. Block Size (2% Ma)

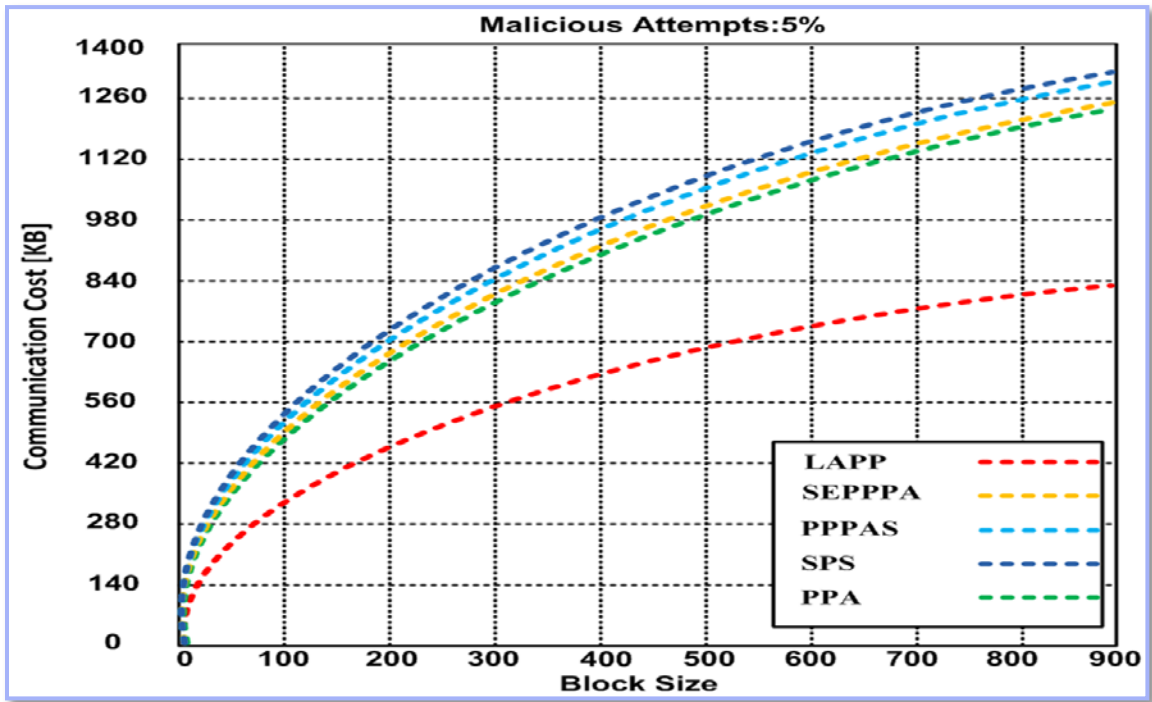


Figure 5.4. Communication Cost Vs. Block Size (5% Ma)

### 5.2.2. Auditing Time per Task Vs. Fraction of Invalid Responses

In this simulation, as depicted in figures 5.5 – 5.8, we measure the auditing time per task in milliseconds in comparison with the fraction of invalid responses; increasing by five on each occurrence. When the fraction of invalid responses reaches 45, the simulation showed a result of 560 ms for LAPP, while all other methods show a fluctuation between 590 ms and 645 ms.

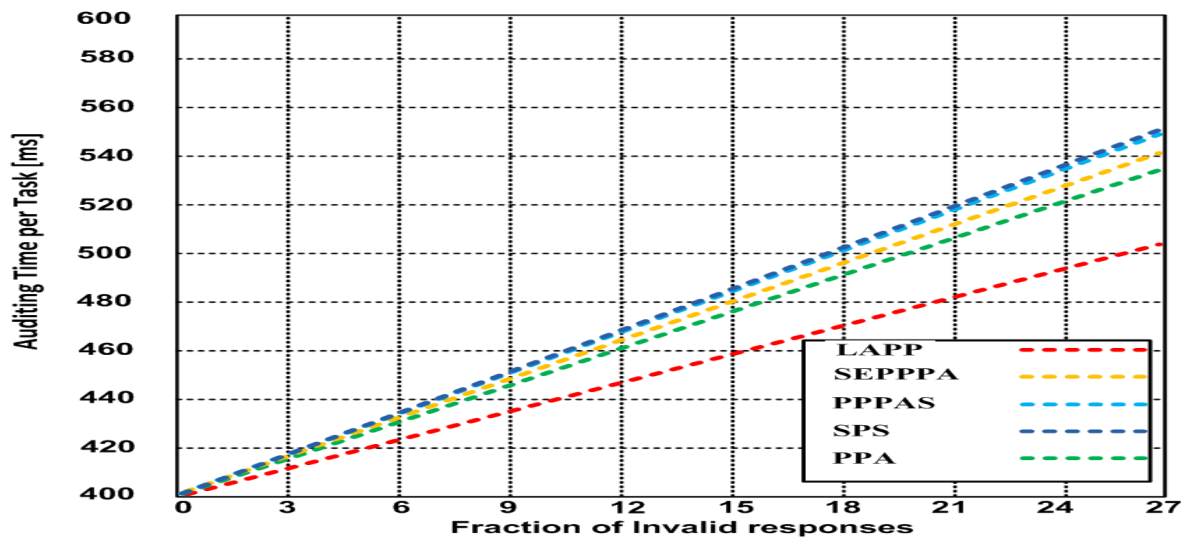


Figure 5.5. Auditing Time per Task Vs. the Fraction of Invalid Responses (0% Ma)

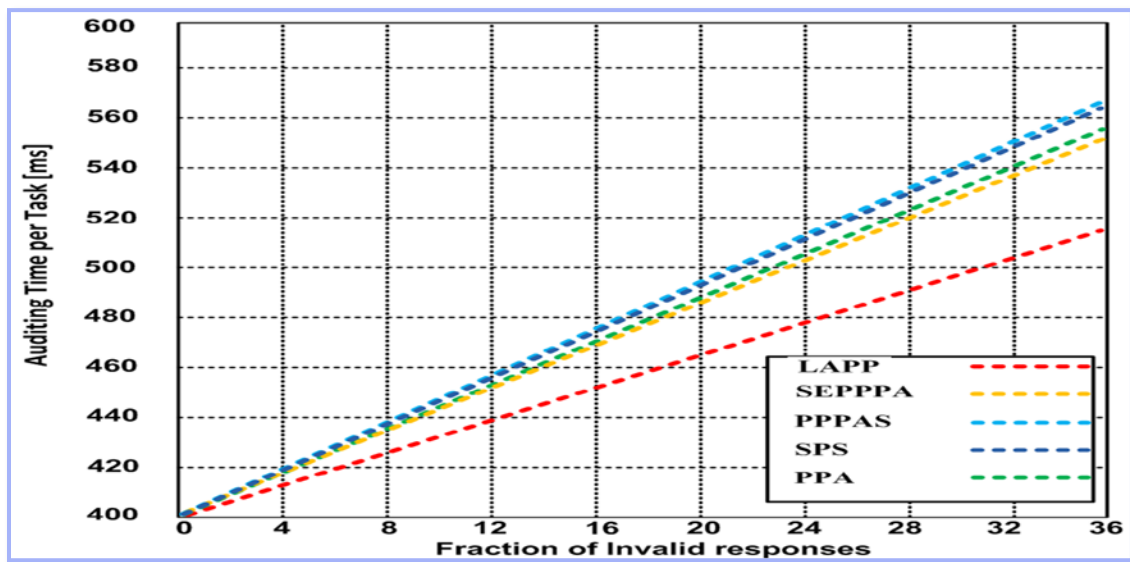


Figure 5.6. Auditing Time per Task Vs. the Fraction of Invalid Responses (1% Ma)

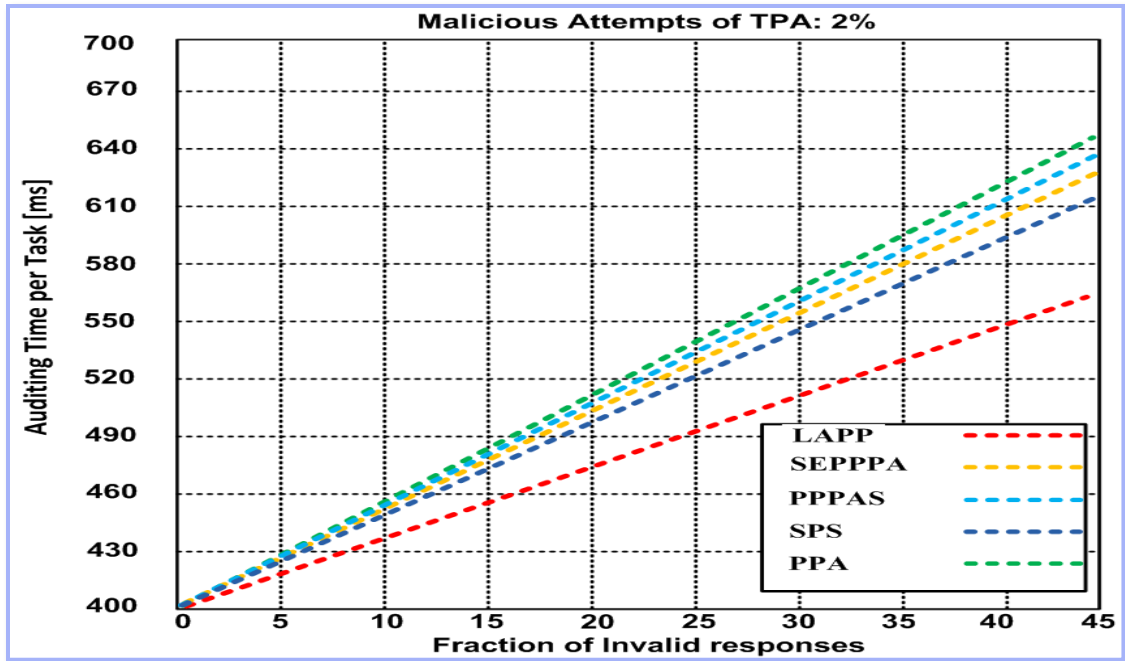


Figure 5.7. Auditing Time per Task Vs. the Fraction of Invalid Responses (2% Ma)

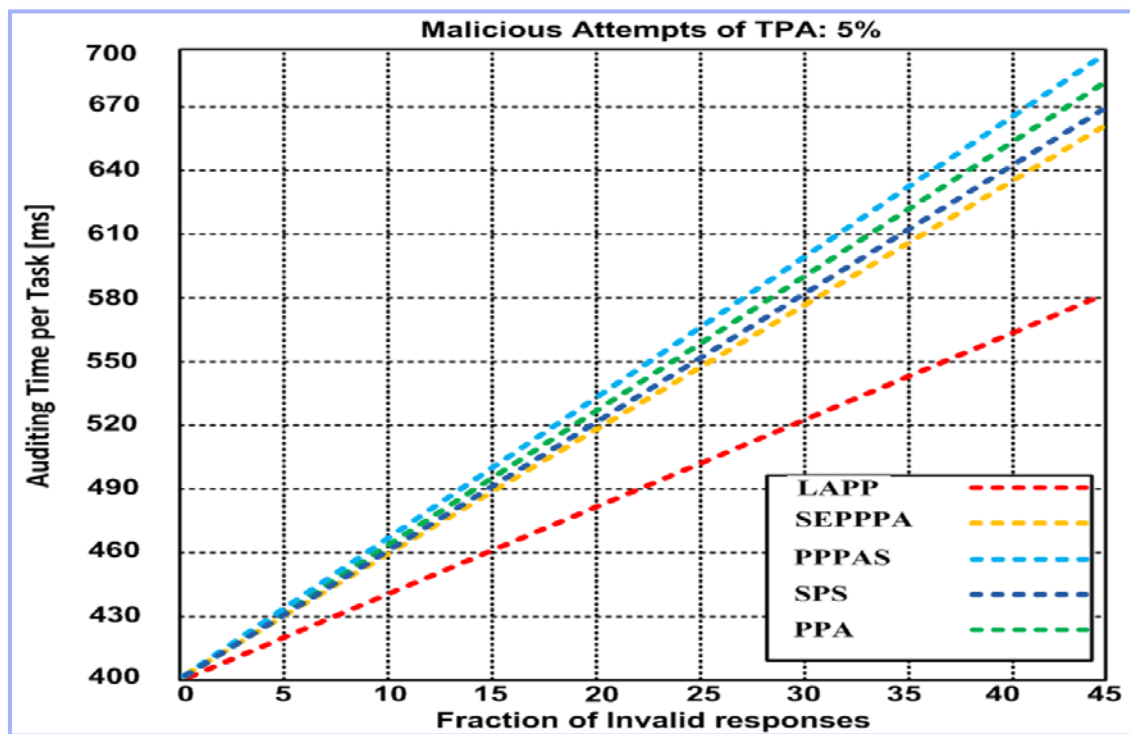


Figure 5.8. Auditing Time per Task Vs. the Fraction of Invalid Responses (5% Ma)



### 5.2.3 Reliable Auditing Detection Vs. Number of Cloud Auditing Users

As depicted in figures 5.9 – 5.12, we measure the percentage of reliable auditing detection (verifying the effectiveness and security of the CSP's controls and methods employed on the CC's data) with some cloud auditing users. We took the measurements by increasing the number of cloud auditing users by 10,000 each time. At a malicious rate of 2%, The LAPP is showing around 99.75% reliability measurement while showing a fluctuation of the other methods we have compared our results to, between 99.58% and 99.46%.

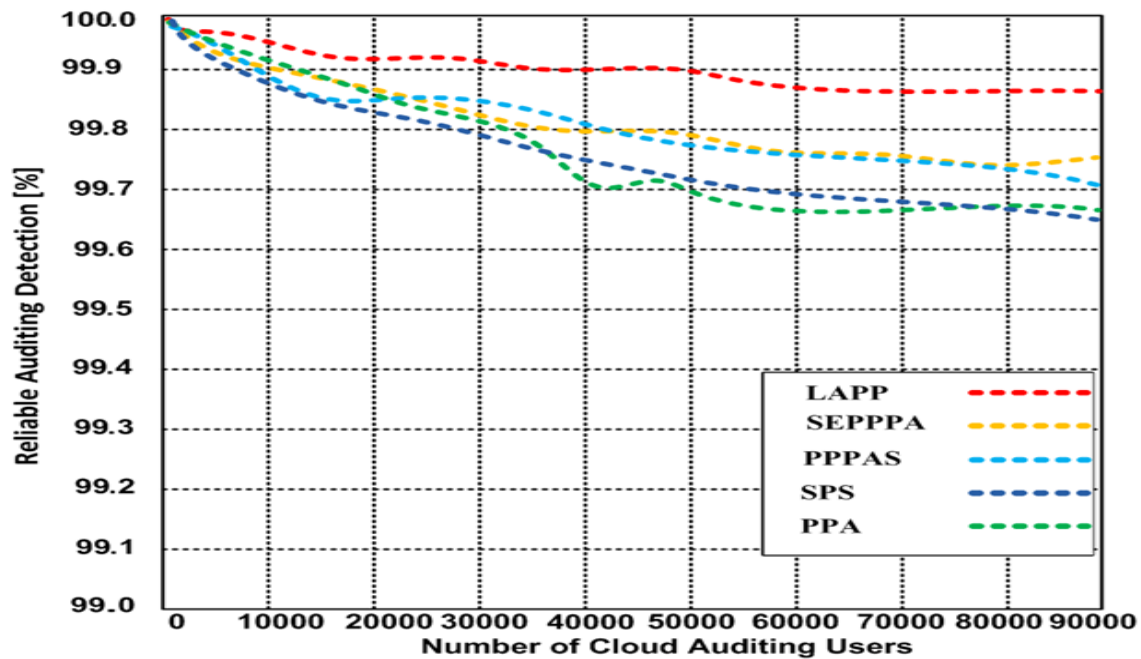


Figure 5.9. Reliable Auditing Detection Vs. Number of Cloud Auditing Users (0% Ma)

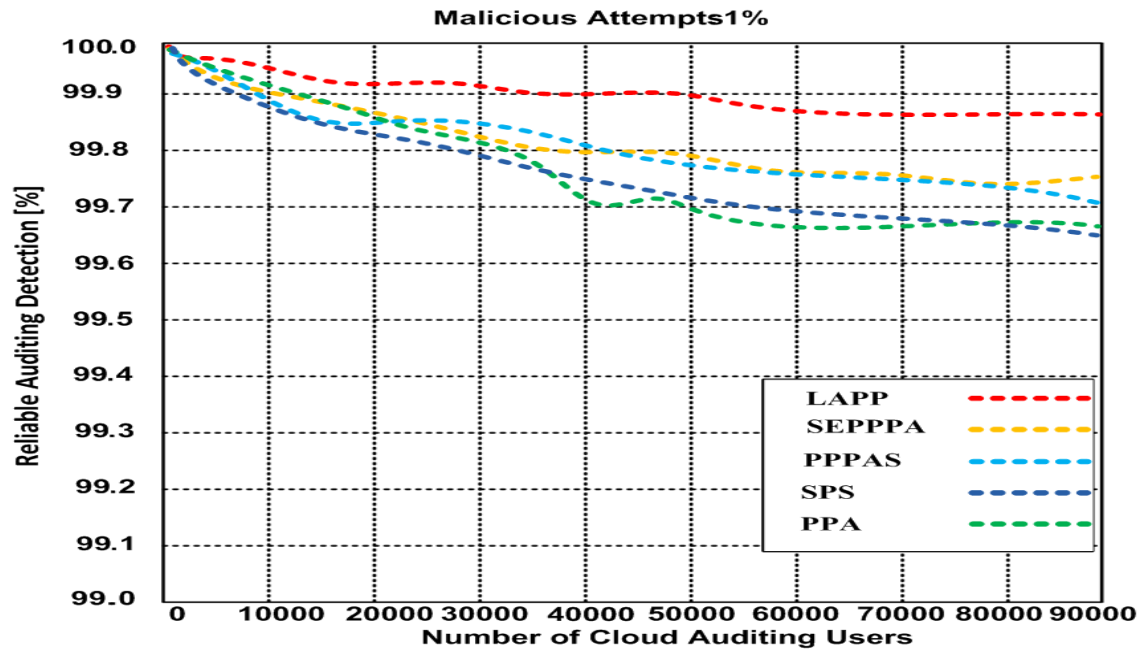


Figure 5.10. Reliable Auditing Detection Vs. Number of Cloud Auditing Users (1% Ma)

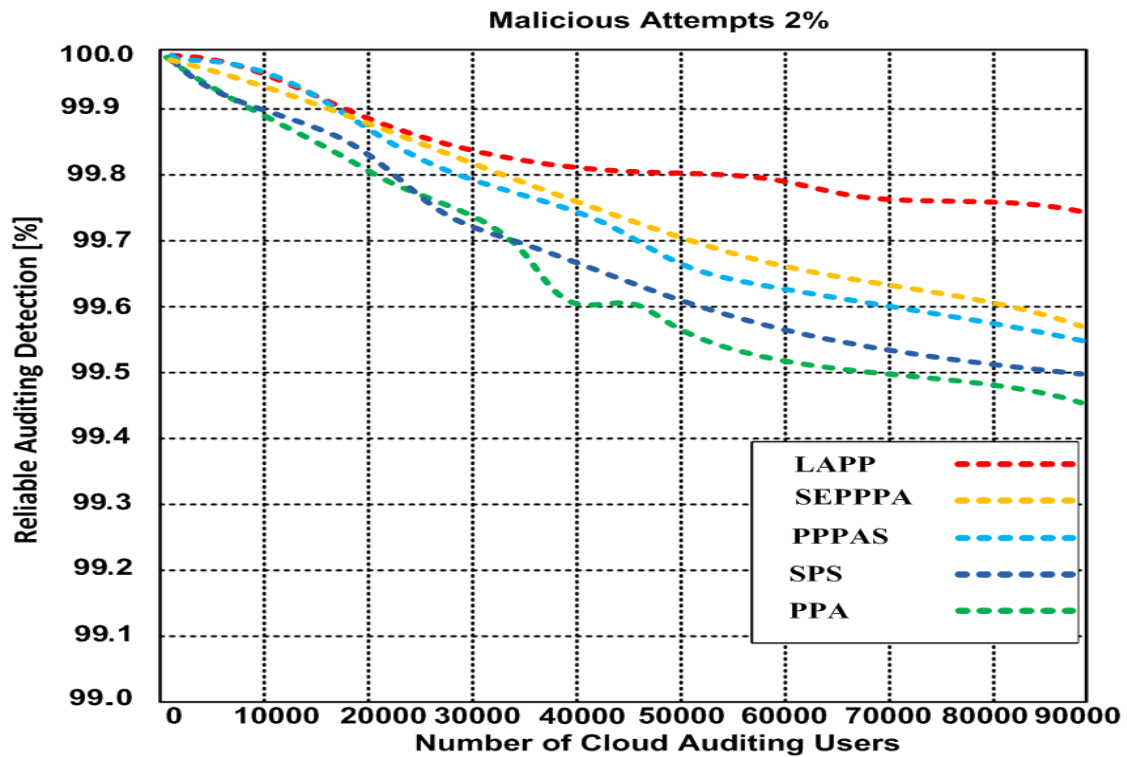


Figure 5.11. Reliable Auditing Detection Vs. Number of Cloud Auditing Users (2% Ma)

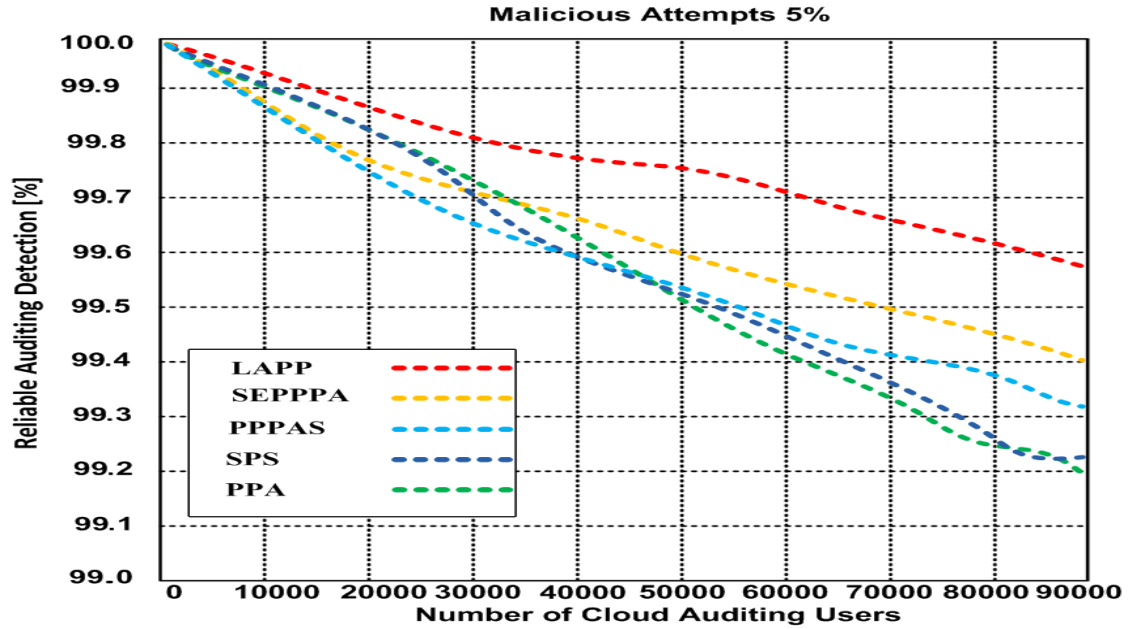


Figure 5.12. Reliable Auditing Detection Vs. Number of Cloud Auditing Users (5% Ma)

#### 5.2.4 Computation Time on Auditing Vs. the Number of Challenged Data Blocks

In this experiment, as depicted in figures 5.13 – 5.16, we measure the computation time on auditing in seconds versus the number of challenged blocks. We proceed from measurement to measurement by increasing the number of challenged blocks by 500. Our results show that at 4500 challenged blocks the computation time on auditing for LAPP is about 2.25 seconds. While for the other methods, it varies between 2.9 and 3.25 seconds.

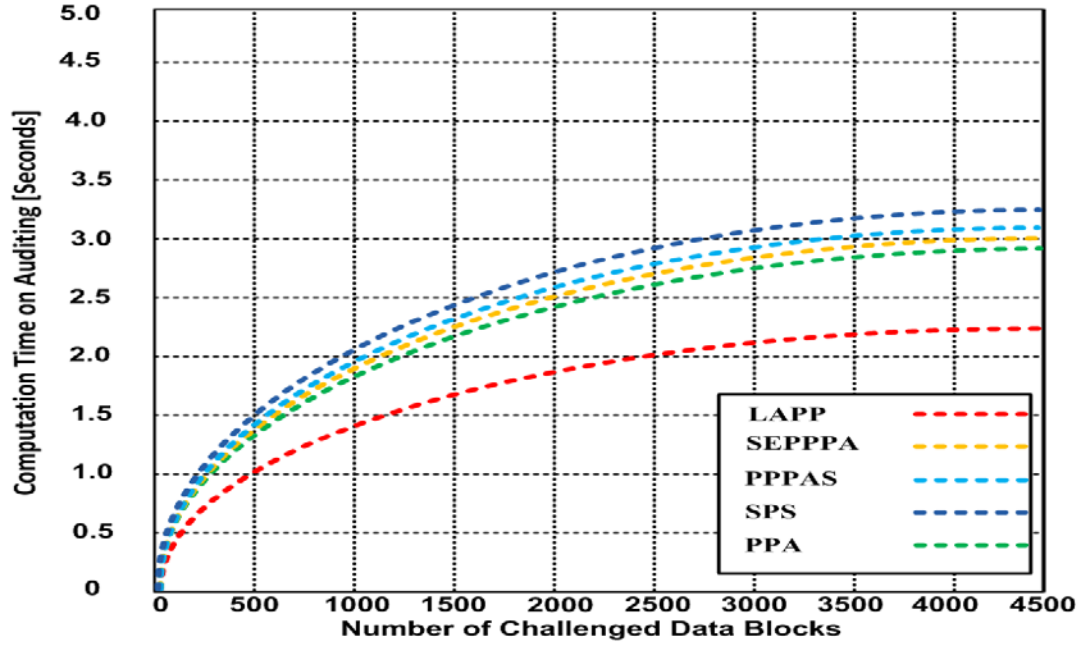


Figure 5.13. Computation time on Auditing (Number of Challenged Blocks) (0% Ma)

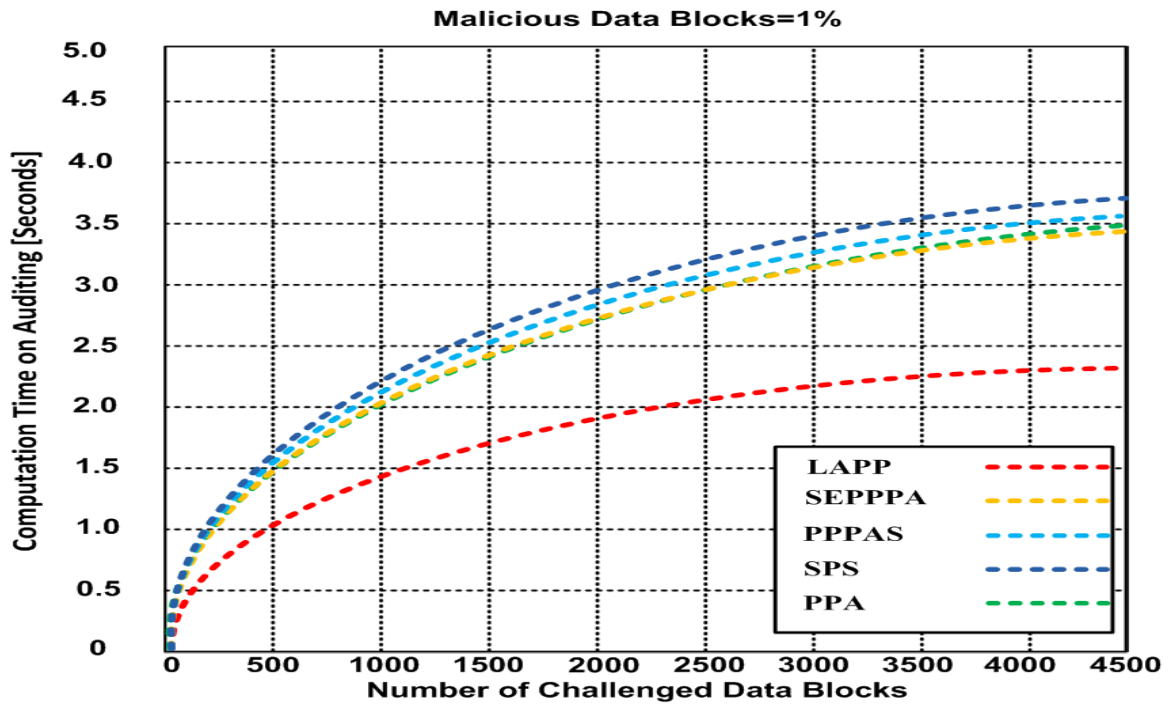


Figure 5.14. Computation time on Auditing (Number of Challenged Blocks) (1% Ma)

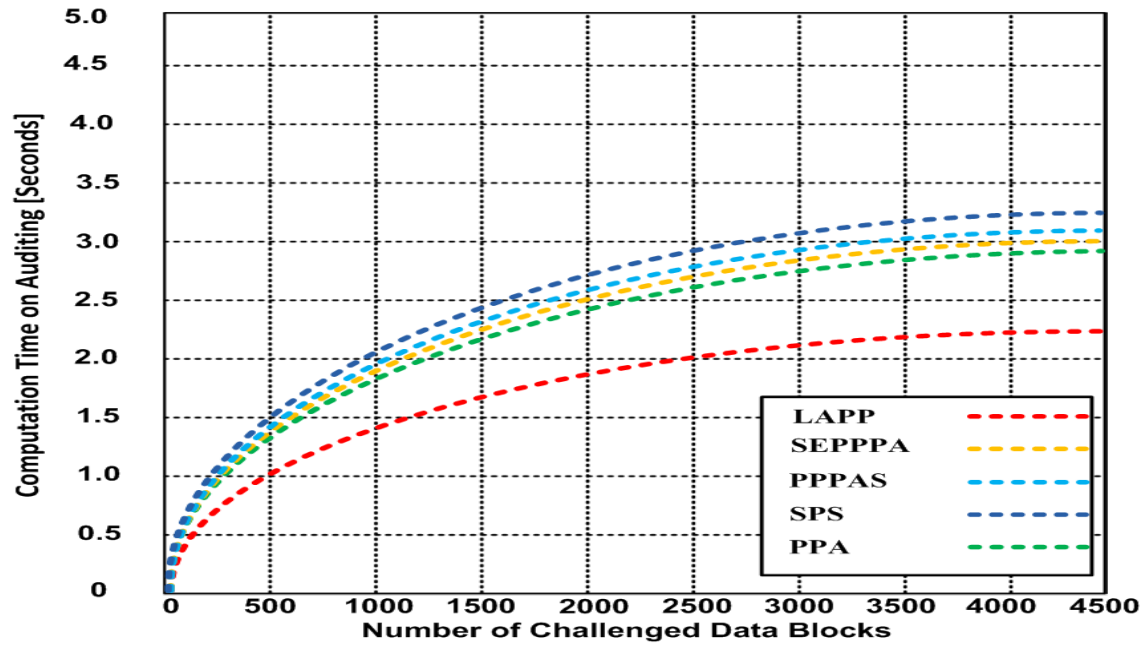


Figure 5.15. Computation time on Auditing (Number of Challenged Blocks) (2% Ma)

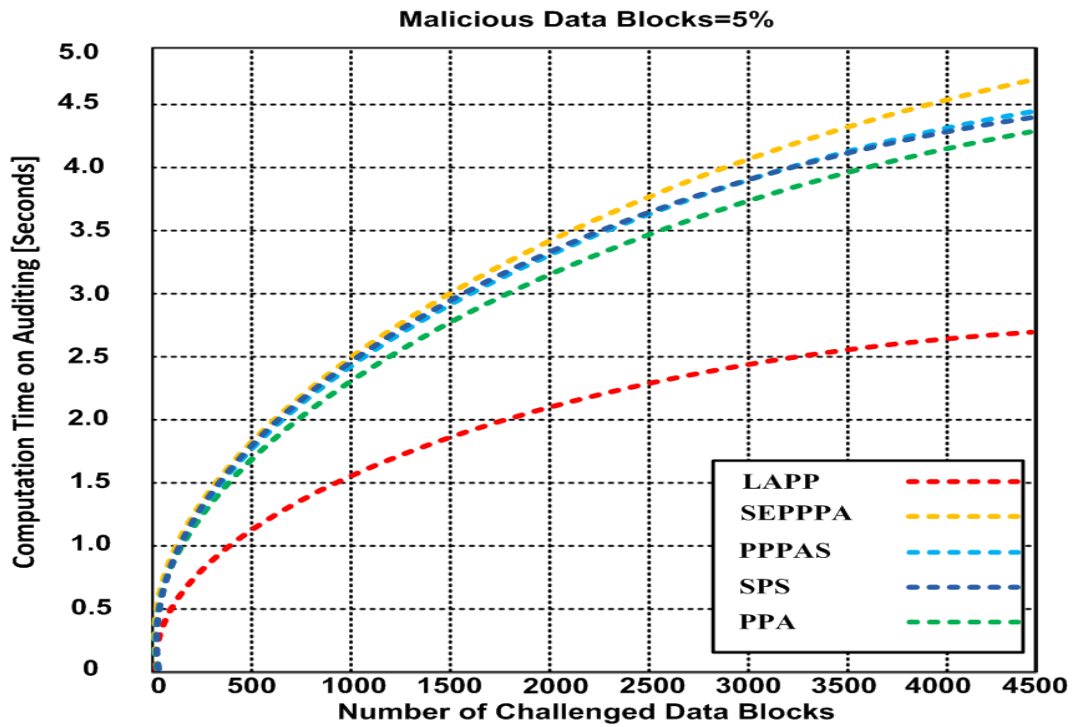


Figure 5.16. Computation time on Auditing (Number of Challenged Blocks) (5% Ma)

### 5.2.5 Accuracy Vs. the Number of Malicious Attempts

In this simulation, as depicted in figures 5.17 – 5.20, we measure the accuracy in percentage versus the number of malicious attempts, by increasing the number of malicious attempts by 3 for every measurement. Our results show that LAPP's accuracy at 99.98% at 27 malicious attempts while the other methods decrease between 99.46% and 99.58%.

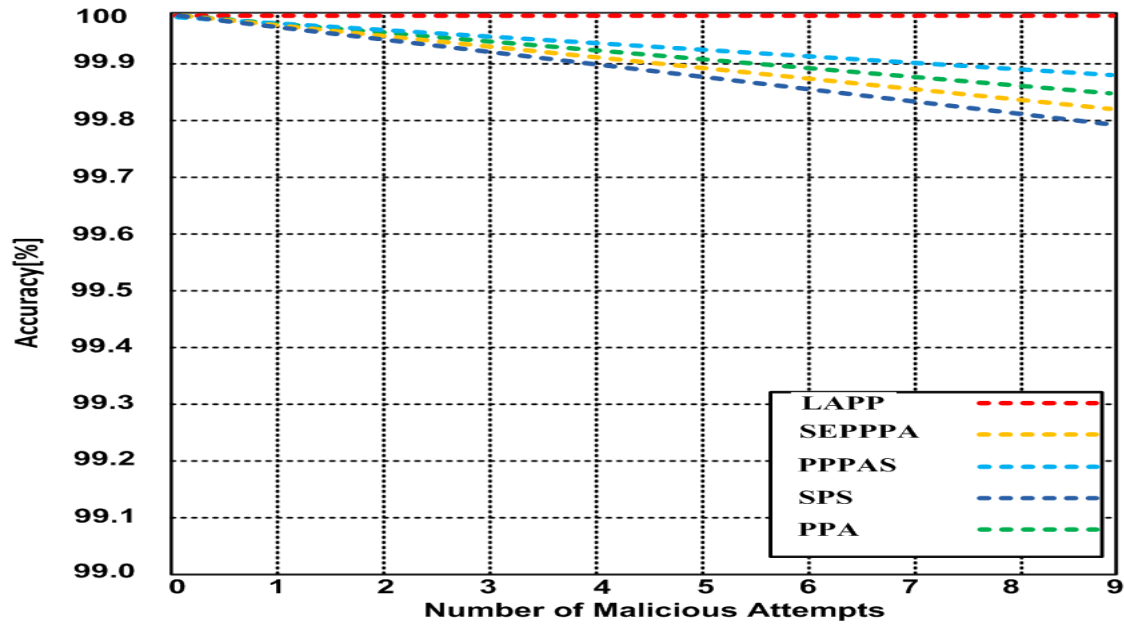


Figure 5.17. Accuracy (Number of Malicious Attempts) (0% Ma)

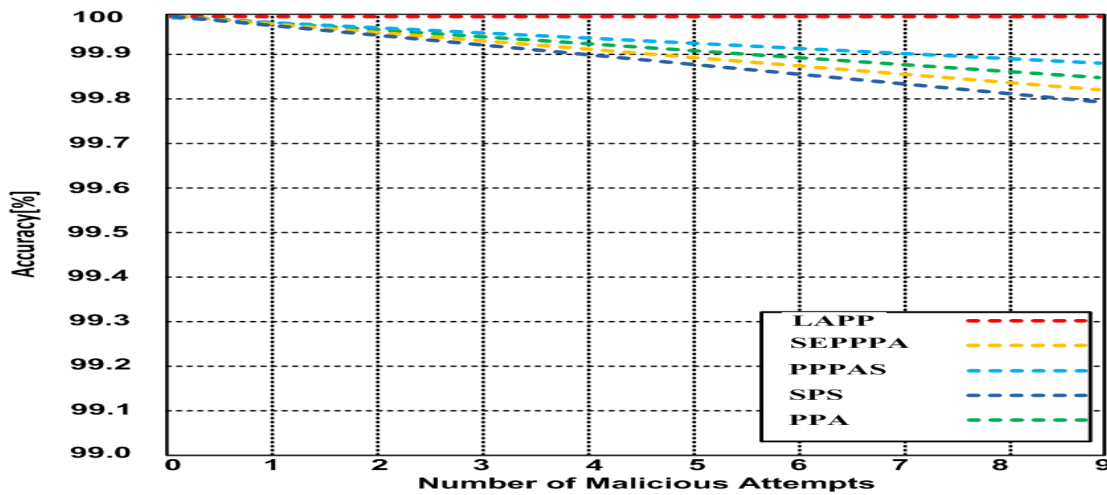


Figure 5.18. Accuracy (Number of Malicious Attempts) (1% Ma)

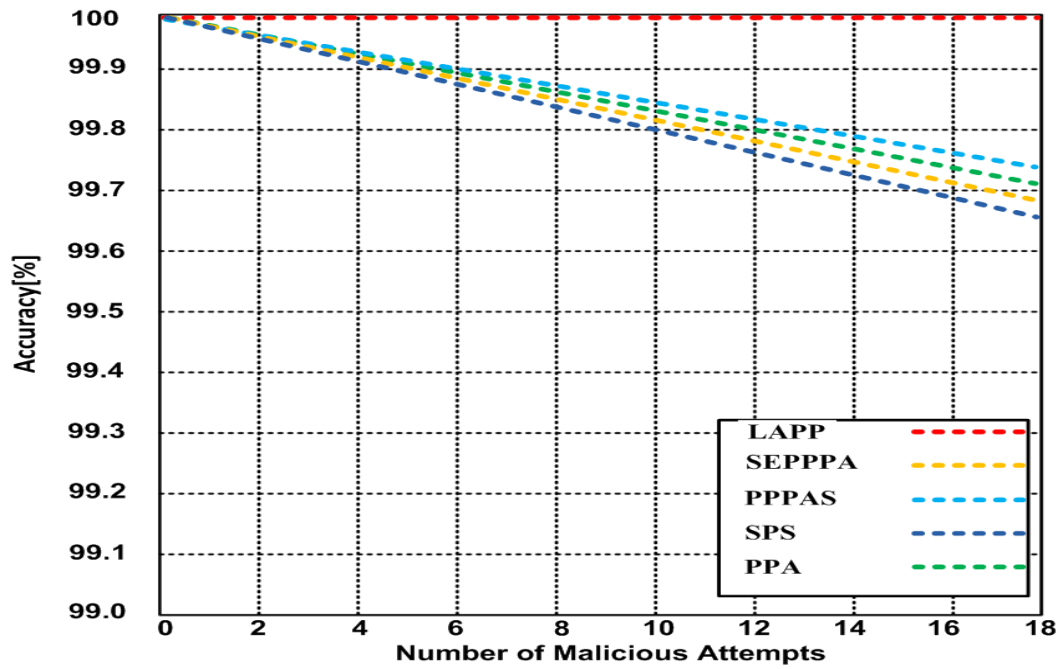


Figure 5.19. Accuracy (Number of Malicious Attempts) (2% Ma)

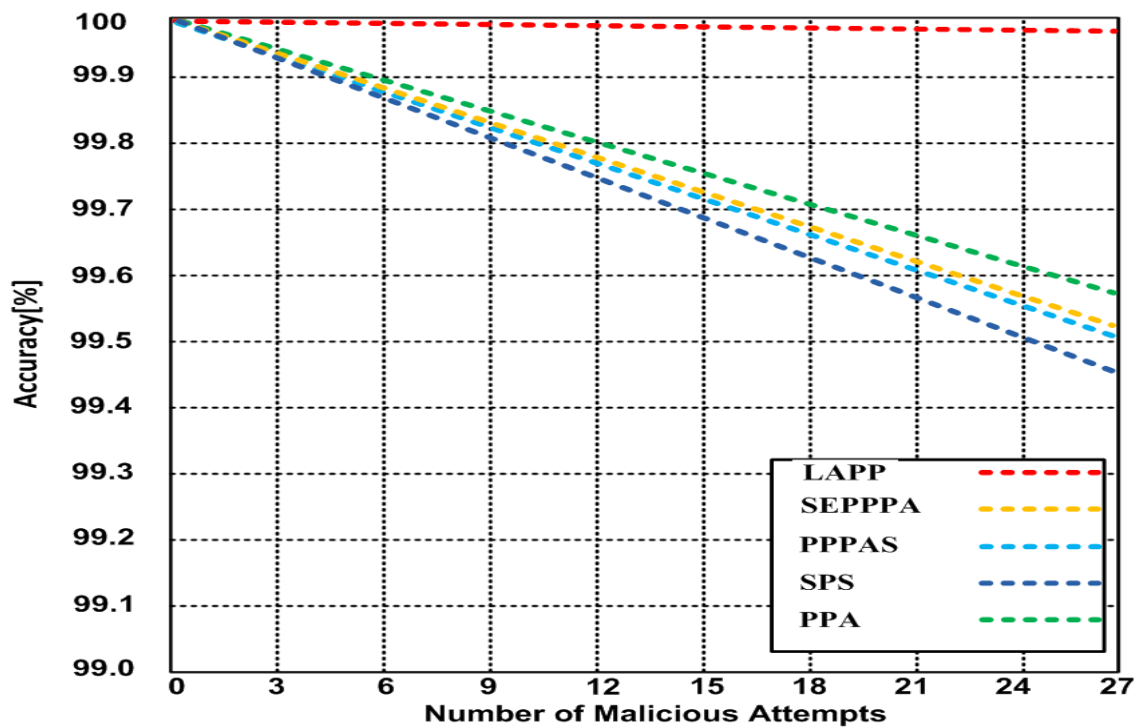


Figure 5.20. Accuracy (Number of Malicious Attempts) (5% Ma)



## 5.2.6 Time complexity Vs. Input Files

In this experiment, as depicted in figures 5.21 – 5.24, we measure the time complexity in seconds versus the size of input files in KB. We proceed from measurement to measurement by increasing the size of the input files by 4 KB. The order of complexity is depicted in terms of the big-O notation. The simulation results show a linear increase for LAPP, logarithmic increase for SEPPA, and PPA; and quadratic time complexity for PPPAS and SPS.

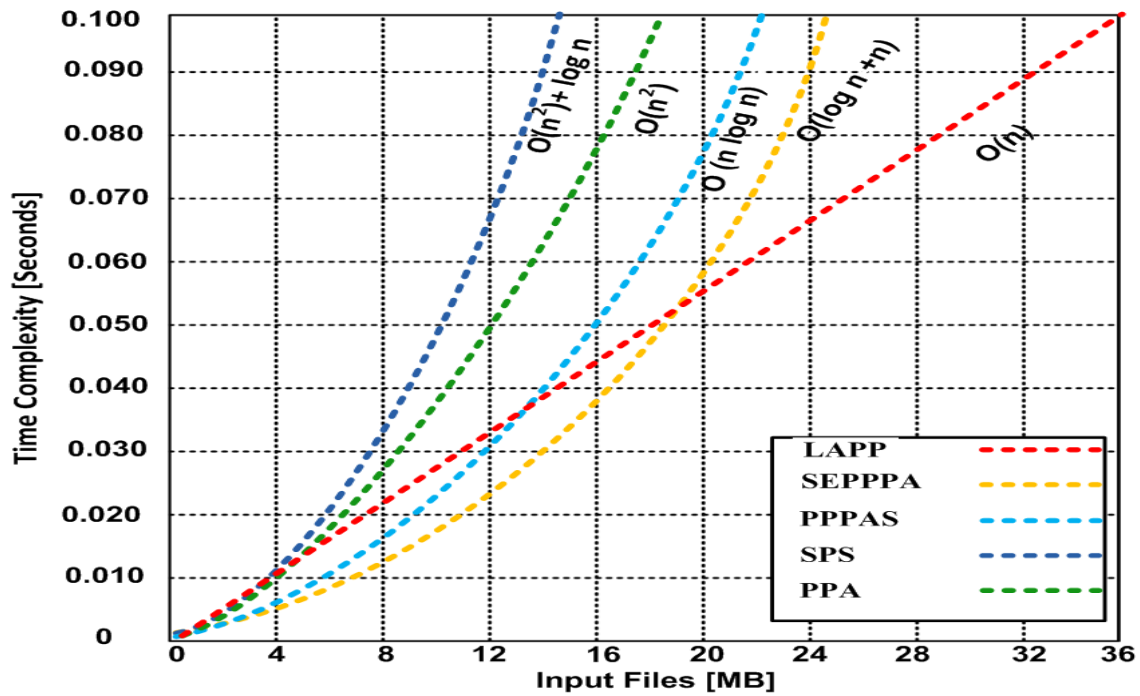


Figure 5.21. Time Complexity (Input Files) (0% Ma)



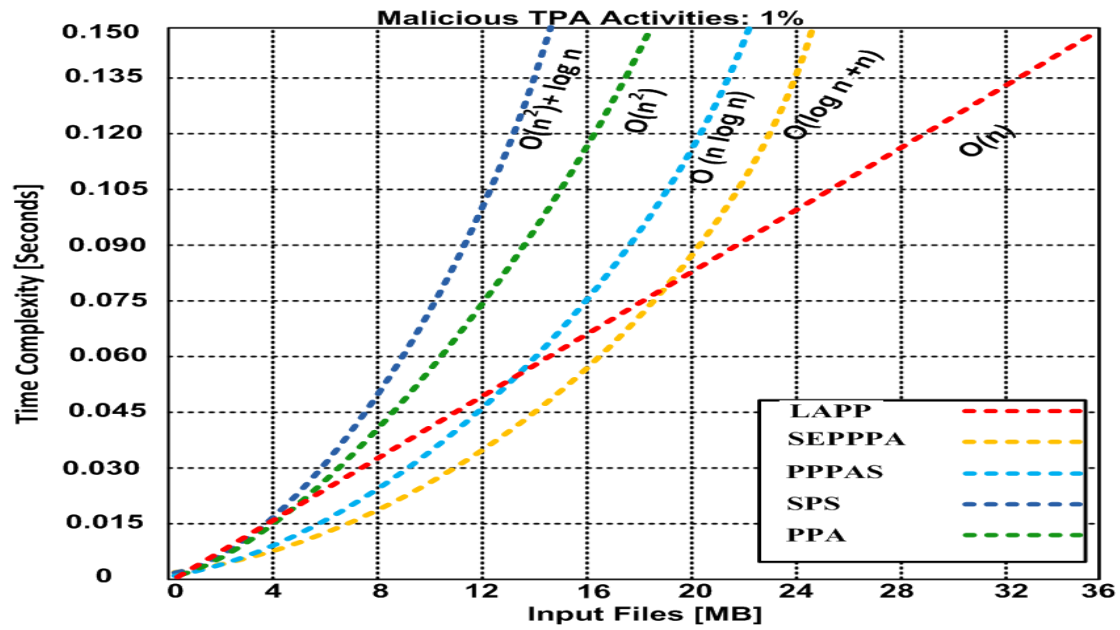


Figure 5.22. Time Complexity (Input Files) (1% Ma)

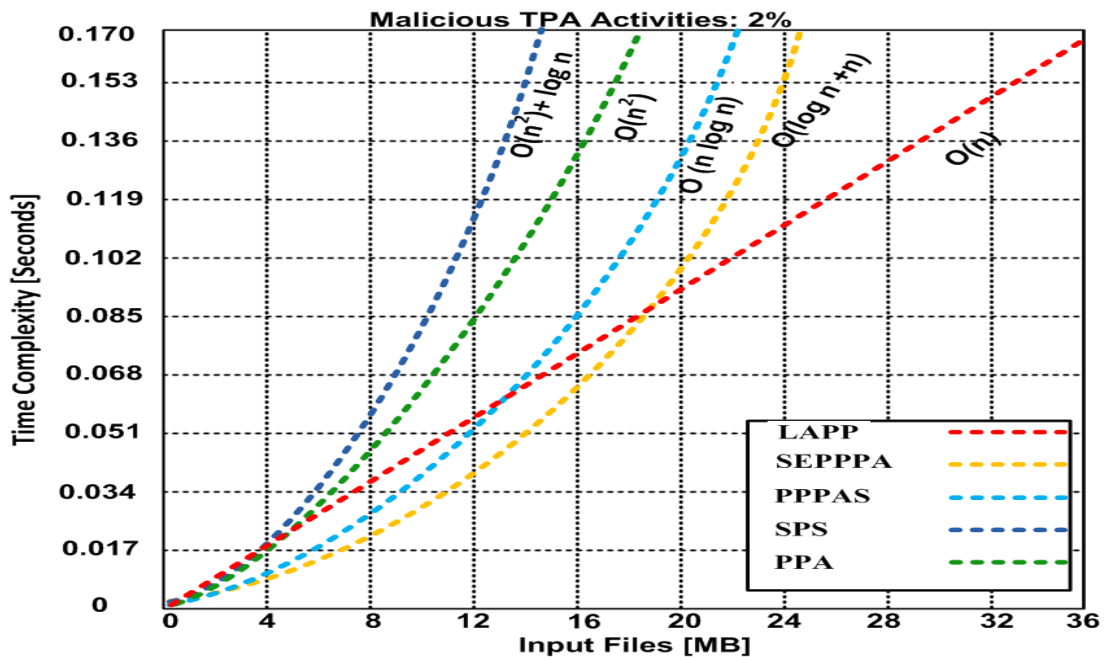


Figure 5.23. Time Complexity (Input Files) (2% Ma)

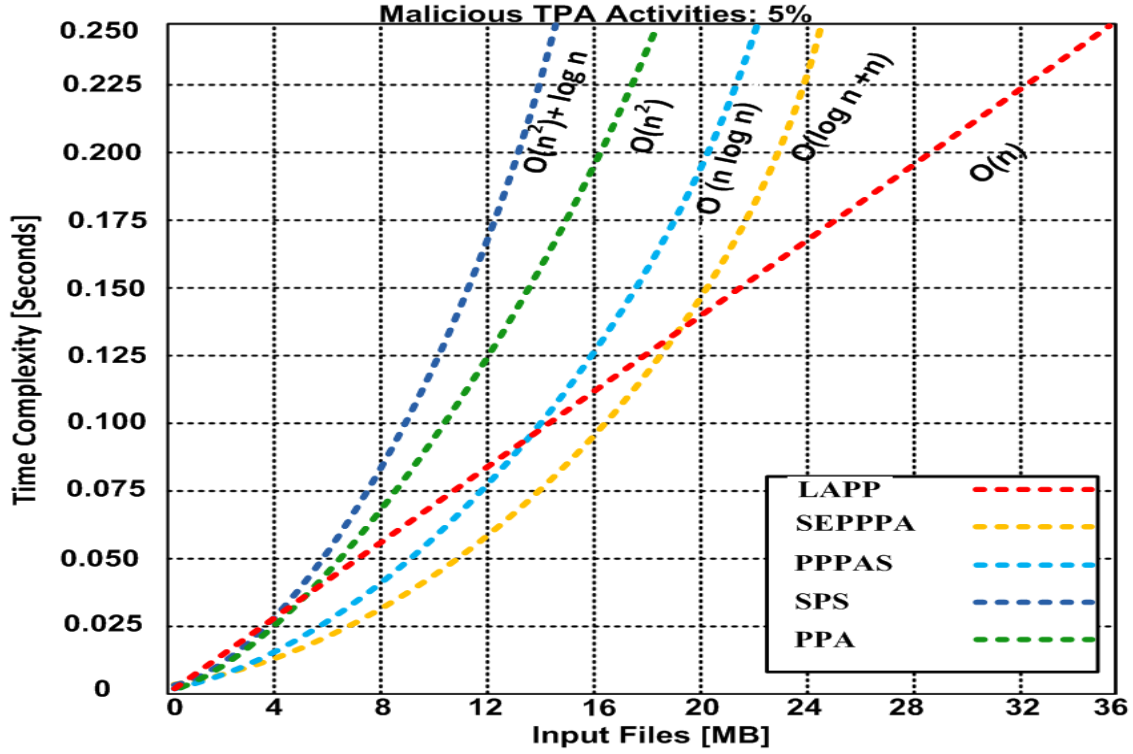


Figure 5.24. Time Complexity (Input Files) (5% Ma)

### 5.2.7 Time Vs. Total Number of Data

Figures 5.25 – 5.28 illustrate the change in the needed processing time versus the volume of the processed data, adding 5 MB of data at a time. The simulation results show that LAPP requires less processing time than the method it has been compared against, as we progress in time.

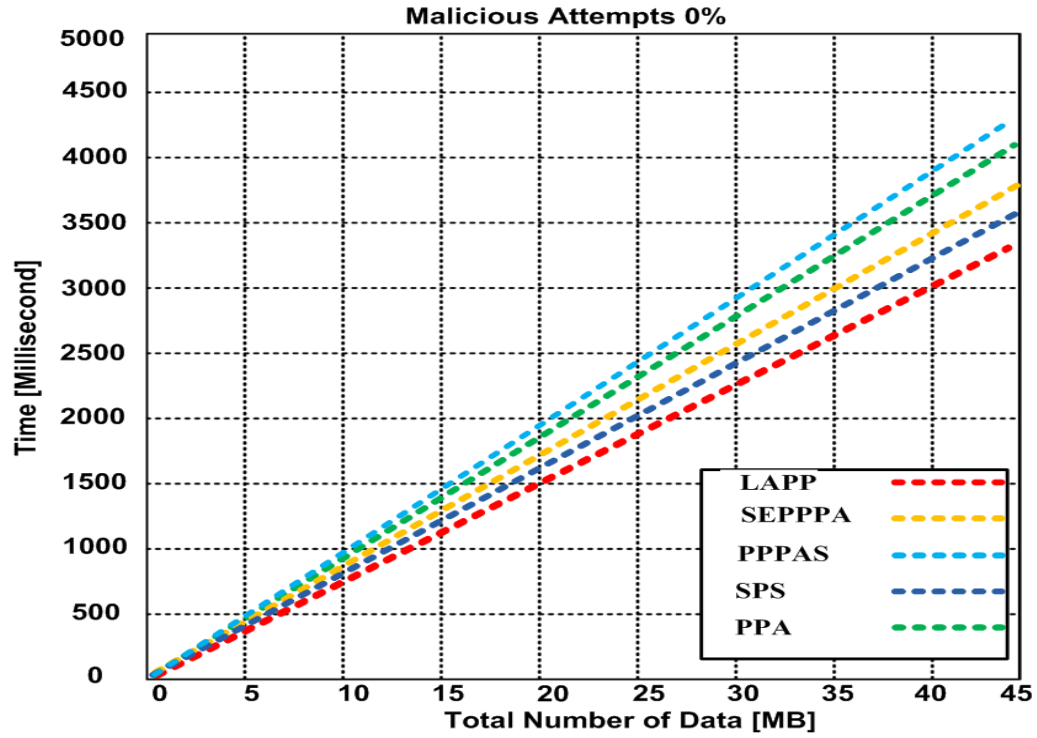


Figure 4. Time (Total Number of Data) (0% Ma)

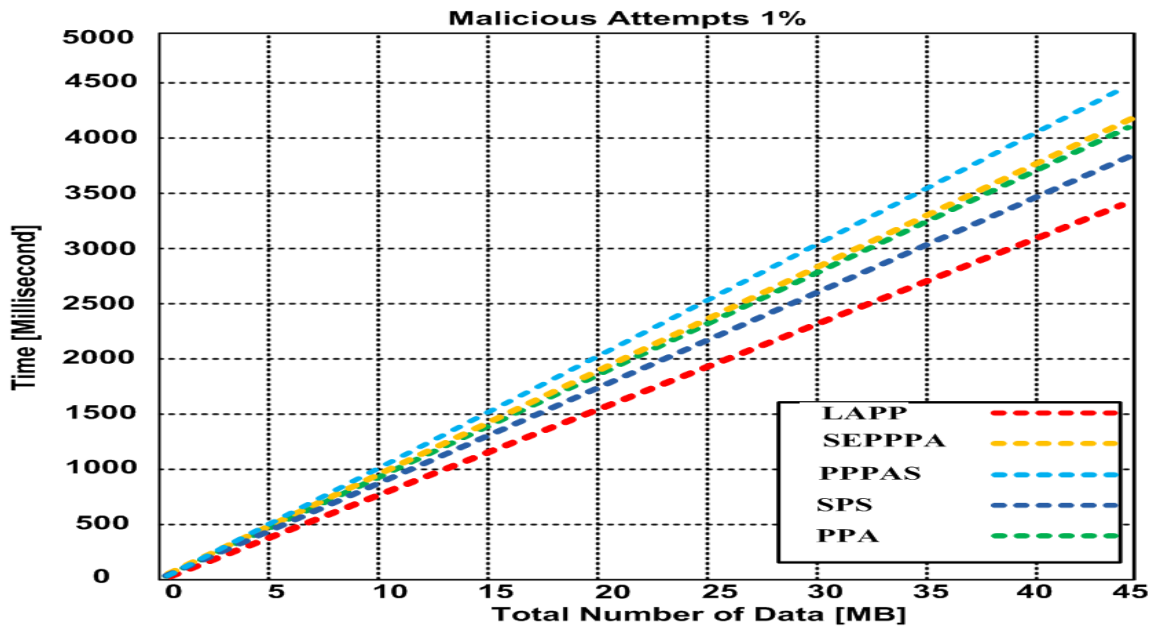


Figure 5.26. Time (Total Number of Data) (1% Ma)

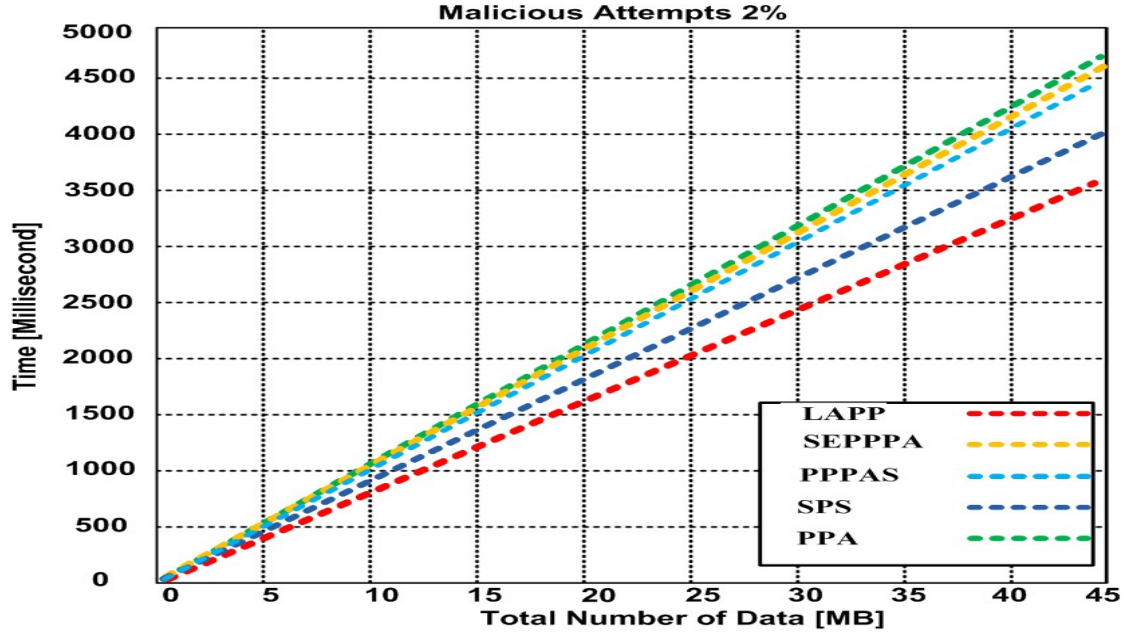


Figure 5.27. Time (Total Number of Data) (2% Ma)

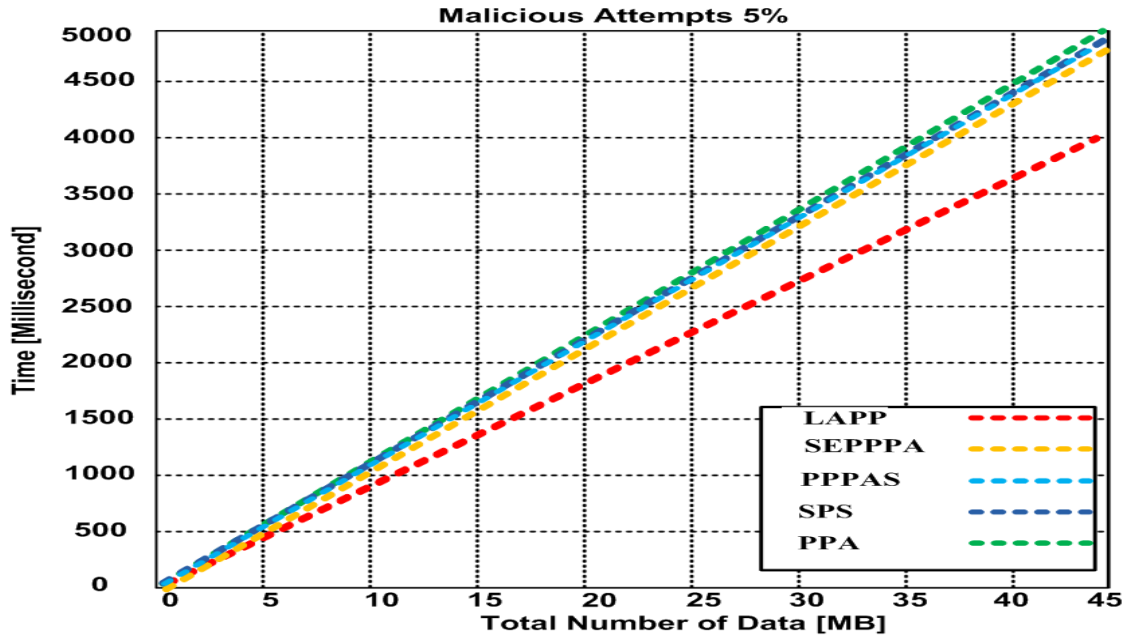


Figure 5.28. Time (Total Number of Data) (5% Ma)

### 5.2.8 The Average Auditing Time Vs. the Number of Clients

Figures 5.29 – 5.32 illustrate the advancement in the average auditing time for each CC versus the number of the CCs in the system, adding 2000 CCs at a time. The simulation

results show that a drop in the average auditing time when we have 8000 users then kept around the same average while increasing the number of users up to 18000 showing superiority to the other methods it has been compared to.

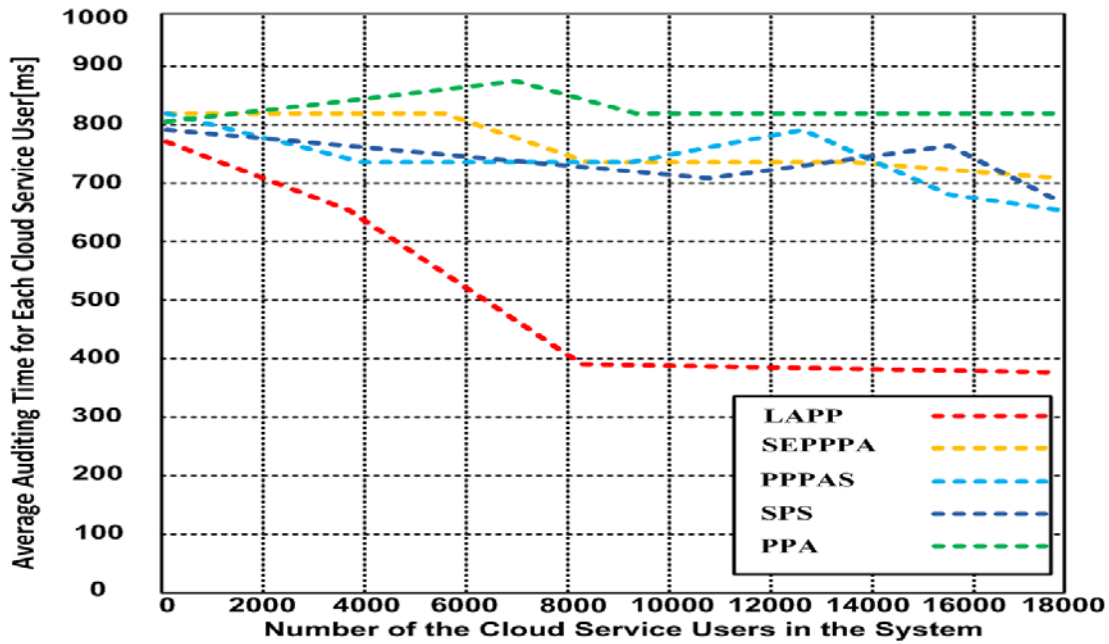


Figure 5.29. The Average Auditing Time (Number of Clients) (0% Ma)

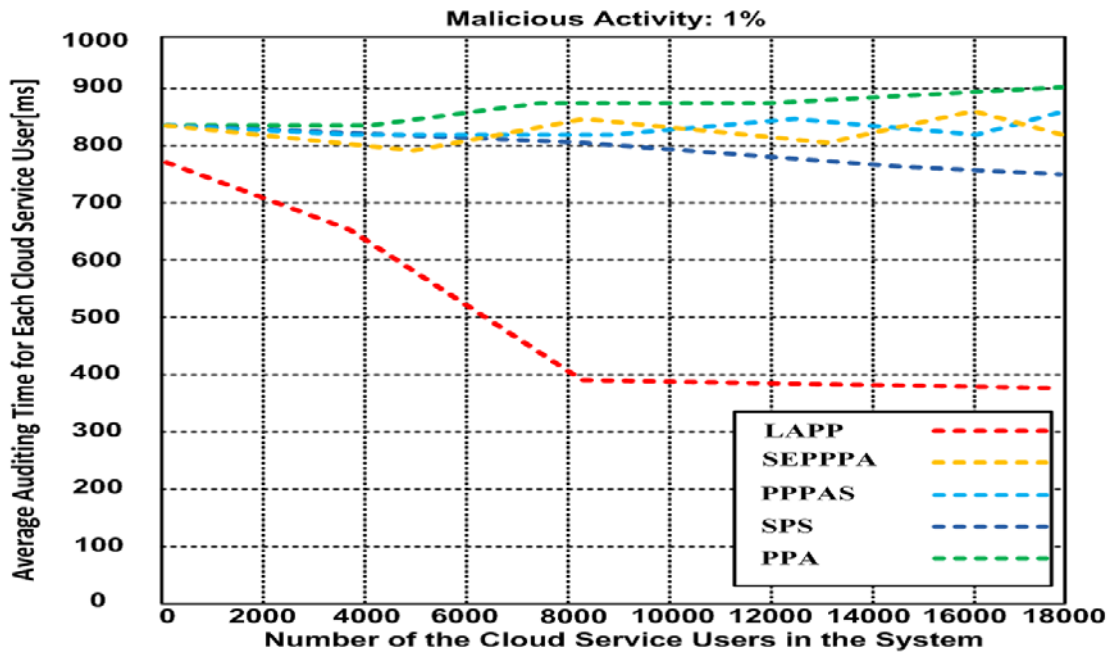


Figure 5.30. The Average Auditing Time (Number of Clients) (1% Ma)

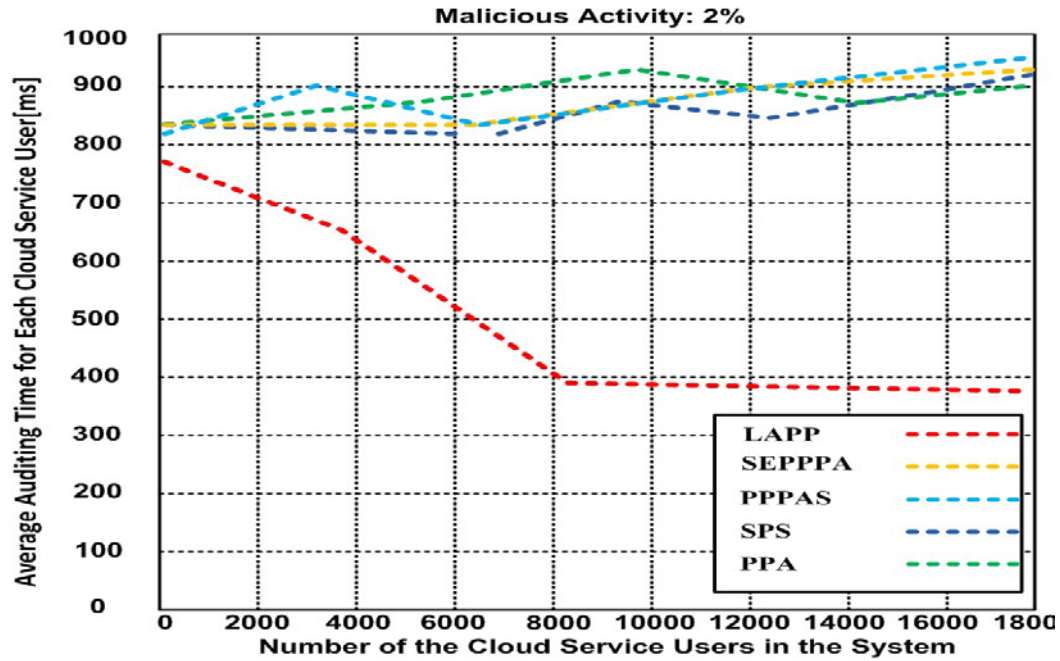


Figure 5.31. The Average Auditing Time (Number of Clients) (2% Ma)

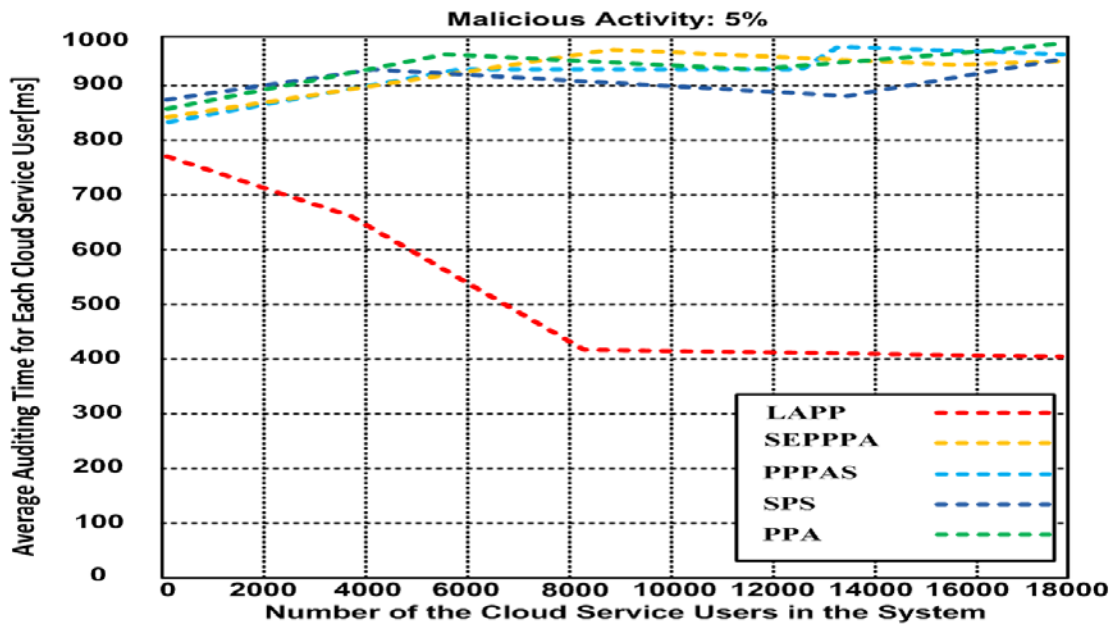


Figure 5.32. The Average Auditing Time (Number of Clients) (5% Ma)

### 5.3 Interpretation of the Results

Based on our newly introduced mathematical model and algorithms, we have proposed a novel model consisting of a secure validation of the keys between the tree stakeholders (CSP, CC, and TPA) allowing us to have an on the fly auditing as we are processing the data.

In the first simulation, we have measured the communication cost depending on the block size up to 900 KB, increasing the block size by 100 each time. Our results showed a noticeable lower computation cost compared to the other methods to which we have compared our results.

In our second simulation, we have measured the auditing time per task, mainly auditing the CSP services and policy violations [2], compared to the fraction of invalid responses (the percentage of the invalid responses compared to the valid ones), and the results showed a shorter average of 50 ms to 80 ms depending on the methods we have compared our results to.

In our third simulation, we have measured the percentage of reliable auditing detection for up to 90000 cloud auditing users, incrementing the number of users by 10000 for each measurement. The simulation results showed a better reliability measurement for LAPP compared to the other methods that we have involved in our simulations.

In our fourth simulation, we measure the computation time on auditing in seconds versus the number of challenged blocks. We proceed from measurement to measurement by increasing the number of challenged blocks by 500. Our results also showed better computation time on auditing for our introduced method.

In our fifth simulation, we measure the accuracy in percentage versus the number of malicious attempts, by increasing the number of malicious attempts by three for every measurement. Our results showed a better accuracy for LAPP.

In our sixth simulation, we measure the time complexity in seconds versus the size of input



files in KB. We proceed from measurement to measurement by increasing the size of the input files by 4 KB. We depict the order of complexity in terms of the big-O notation. The simulation results show a linear increase for LAPP, logarithmic increase for SEPPA, and PPA; and quadratic time complexity for PPPAS and SPS.

In our seventh simulation, we measure the change in the needed processing time versus the volume of the processed data, adding 5 MB of data at a time. The simulation results show that LAPP requires less processing time than the methods it has been compared against, as we progress in time.

In our last simulation, we measure the advancement in the average auditing time for each CC versus the number of the CCs in the system, adding 2000 CCs at a time. The simulation results show a drop in the average auditing time when we have 8000 users then kept around the same average while increasing the number of users up to 18000, showing superiority to the other methods to which we compare LAPP.

These results show the efficiency of our proposed method (LAPP) as well as the optimization in reducing the processing and communication overheads as well as a better quality of service.



## CHAPTER 6: CONCLUSION

In this dissertation, we have studied cloud security based on a Third-Party Auditor (TPA). The TPA's role in cloud computing is to assure the auditing function on behalf of the Cloud Client (CC) and to validate the security of the connection while guaranteeing the integrity of the data. Nonetheless, there are some issues primarily related to trust that could emerge from involving a TPA. Many research papers addressing security using a TPA and proposing solutions have been elaborated and published.

Our study reveals the significant impact brought in by the TPA's adoption in securing cloud computing, as adopting a solution with a TPA can come with trust concerns, extra overhead, security, and data manipulation breaches.

In this dissertation, we have oriented our efforts towards developing a more simplistic lightweight, and secure solution that will help alleviate the gap in the CC's decision and to increase the trust in adopting a resolution based on a TPA. Allowing the CC to audit the TPA and the CSP for malicious activities. Hence, we have introduced the Light-weight Accountable Privacy-Preserving protocol (LAPP).

LAPP's primary function is to assure a smooth "auditing of the auditor" process. We have obtained our simulation results with the introduction of 0%, 1%, 2%, and 5% malicious TPA activities. The results demonstrated the superiority of the LAPP method versus other protocols against which it was compared.

LAPP allows the CC to audit the TPA and the CSP for malicious activities, which increases the confidence and the willingness of more companies to embrace the cloud realm.

LAPP is also positioned to increase trust and grant more control to the CC to timely detect issues, and to take practical actions by introducing a malicious-detection algorithm that enables both parties to keep a check and balance on each other.

Our simulation results demonstrated LAPP assurance to the following:

- Expending a minimum overhead while successfully issuing the secret key to the three stakeholders (CSP, CC, and TPA).
- Determining and avoiding the malicious role of the TPA, if any, with a lightweight and straightforward algorithm.
- Guaranteeing a more secure communication at a minimum communication-cost.
- Accurately detecting malicious activities.
- Improving the Quality-of-Service (QoS) provision, mainly determined through our time complexity simulation results.

In our future work, we are planning to apply LAPP on different sizes of networks to see its effect on small to medium systems.

## REFERENCES

- [1] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, "Cloud computing: Distributed internet computing for IT and scientific research," *IEEE Internet computing*, 2009, vol. 13.
- [2] A. Razaque and S. S. Rizvi, "Privacy preserving model: a new scheme for auditing cloud stakeholders," *Journal of Cloud Computing*, 2017, vol. 6, p. 7.
- [3] P. Mell and T. Grance, "The NIST definition of cloud computing," 2011, Special Publication 800 -145, pp 2-3 .
- [4] M. Zhao, Y. Ding, Y. Wang, H. Wang, B. Wang, and L. Liu, "A Privacy-Preserving TPA-aided Remote Data Integrity Auditing Scheme in Clouds," in *International Conference of Pioneering Computer Scientists, Engineers and Educators*, 2019, pp. 334-345.
- [5] K. Gai, M. Qiu, and H. Zhao, "Privacy-preserving data encryption strategy for big data in mobile cloud computing," *IEEE Transactions on Big Data*, 2017.
- [6] B. Chen, R. Curtmola, G. Ateniese, and R. Burns, "Remote data checking for network coding-based distributed storage systems," in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*, 2010, pp. 31-42.
- [7] J. Kaur and J. Singh, "Monitoring Data Integrity while using TPA in Cloud Environment," *Global Journal of Computer Science and Technology*, 2013, vol. 13.
- [8] G. B. Bethel and S. A. Reddy, "Implementing a Role Based Self Contained Data Protection Scheme in Cloud Computing," in *International Conference on Inventive Computation Technologies*, 2019, pp. 653-660.
- [9] B. Grobauer, T. Walloschek, and E. Stöcker, "Understanding cloud computing vulnerabilities," *Security & privacy, IEEE*, 2011, vol. 9, pp. 50-57.

- [10] F. Sabahi, "Cloud computing security threats and responses," in *Communication Software and Networks (ICCSN), 2011 IEEE 3rd International Conference on*, 2011, pp. 245-249.
- [11] K. S. Wilson and M. A. Kiy, "Some fundamental cybersecurity concepts," *IEEE Access*, 2014, vol. 2, pp. 116-124.
- [12] T. H. Noor, Q. Z. Sheng, L. Yao, S. Dustdar, and A. H. Ngu, "CloudArmor: Supporting reputation-based trust management for cloud services," *IEEE transactions on parallel and distributed systems*, 2016, vol. 27, pp. 367-380.
- [13] H. Zhao, X. Yao, and X. Zheng, "Privacy-preserving TPA Auditing Scheme Based on Skip List for Cloud Storage," *IJ Network Security*, 2019, vol. 21, pp. 451-461.
- [14] Y. L. Sun, Z. Han, W. Yu, and K. R. Liu, "A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks," in *INFOCOM*, 2006, pp. 1-13.
- [15] D. Wang, T. Muller, Y. Liu, and J. Zhang, "Towards robust and effective trust management for security: A survey," in *2014 IEEE 13th International Conference on Trust, Security and Privacy in Computing and Communications*, 2014, pp. 511-518.
- [16] S. Pavithra, S. Ramya, and S. Prathibha, "A Survey On Cloud Security Issues And Blockchain," in *2019 3rd International Conference on Computing and Communications Technologies (ICCCCT)*, 2019, pp. 136-140.
- [17] C. Wang, K. Ren, S. Yu, and K. M. R. Urs, "Achieving usable and privacy-assured similarity search over outsourced cloud data," in *INFOCOM, 2012 Proceedings IEEE*, 2012, pp. 451-459.
- [18] L. A. Martucci, A. Zuccato, B. Smeets, S. M. Habib, T. Johansson, and N. Shahmehri, "Privacy, security and trust in cloud computing: the perspective of the telecommunication

- industry," in *Ubiquitous Intelligence & Computing and 9th International Conference on Autonomic & Trusted Computing (UIC/ATC), 2012 9th International Conference on*, 2012, pp. 627-632.
- [19] K. Popović and Ž. Hocenski, "Cloud computing security issues and challenges," in *Proceedings of the 33rd International Convention. IEEEExplore, Opatija*, 2010, pp. 344-349.
  - [20] J. A. Bowen, "Cloud computing: Issues in data privacy/security and commercial considerations," *COMPUTER AND INTERNET LAWYER*, 2011, vol. 28, pp. 1-8.
  - [21] M. D. Ryan, "Cloud computing security: The scientific challenge, and a survey of solutions," *Journal of Systems and Software*, 2013, vol. 86, pp. 2263-2268.
  - [22] L. Wei, H. Zhu, Z. Cao, X. Dong, W. Jia, Y. Chen, *et al.*, "Security and privacy for storage and computation in cloud computing," *Information Sciences*, 2014, vol. 258, pp. 371-386.
  - [23] O. Kocabas and T. Soyata, "Towards privacy-preserving medical cloud computing using homomorphic encryption," in *Virtual and Mobile Healthcare: Breakthroughs in Research and Practice*, ed: IGI Global, 2020, pp. 93-125.
  - [24] M. Jouini and L. B. A. Rabai, "A security framework for secure cloud computing environments," in *Cloud security: Concepts, methodologies, tools, and applications*, ed: IGI Global, 2019, pp. 249-263.
  - [25] Z. A. Hussien, H. Jin, Z. A. Abduljabbar, A. A. Yassin, M. A. Hussain, S. H. Abbdal, *et al.*, "Public auditing for secure data storage in cloud through a third party auditor using modern ciphertext," in *Information Assurance and Security (IAS), 2015 11th International Conference on*, 2015, pp. 73-78.

- [26] Q. W. Cong Wang, and Kui Ren, Wenjing Lou, "Privacy-Preserving Public Auditing for Data Storage Security in Cloud Computing," 2010, INFOCOM, 2010 Proceedings IEEE, pp. 1-9.
- [27] B. Wang, B. Li, and H. Li, "Panda - Public Auditing for Shared Data with Efficient User Revocation in the Cloud," 2015, IEEE Transactions on services computing, vol. 8, pp. 92-106.
- [28] S.Pavithra, "Secure Data Storage in Cloud using Code Regeneration and Public Audition," 2016, International Journal of Emerging Technology in Computer Science & Electronics (IJETCSE), vol. 20, Issue 2.
- [29] S. G. Worku, C. Xu, J. Zhao, and X. He, "Secure and efficient privacy-preserving public auditing scheme for cloud storage," *Computers & Electrical Engineering*, 2014, vol. 40, pp. 1703-1713.
- [30] B. P. Gajendra, V. Singh, and V. Kumar, "Achieving cloud security using TPA." 2016 International Conference on Computing, Communication and Automation (ICCCA), pp 1304-1309.
- [31] K. Yang, and X. Jia, " An efficient and secure dynamic auditing protocol for data storage in cloud computing," 2012, IEEE transactions on parallel and distributed systems, vol.24, pp. 1717-1726.
- [32] F. Moghaddam, O. Karimi, and M. Alrashdan, "A Comparative Study of Applying Real-Time Encryption in Cloud Computing Environments," 2013 IEEE 2nd International Conference on Cloud Networking (CloudNet), pp. 185-189.

- [33] W. B. Lee and C. C.Chang, "Efficient group signature scheme based on the discrete logarithm", 1998, *IEEE Proceedings-Computers and Digital Techniques*, vol. 145, pp. 15-18.
- [34] B. Wang, H. Li, and M. Li, "Privacy-preserving public auditing for shared cloud data supporting group dynamics," in *2013 IEEE International Conference on Communications (ICC)*, 2013, pp. 1946-1950.
- [35] Q. Wang, C. Wang, K. Ren, W. Lou, and J. Li, "Enabling public auditability and data dynamics for storage security in cloud computing," *IEEE transactions on parallel and distributed systems*, 2011, vol. 22, pp. 847-859.
- [36] S. Rizvi, A. Razaque, and K. Cover, "Third-Party Auditor (TPA): A Potential Solution for Securing a Cloud Environment," in *Cyber Security and Cloud Computing (CSCloud), 2015 IEEE 2nd International Conference on*, 2015, pp. 31-36.
- [37] S. Rizvi, A. Razaque, and K. Cover, "Cloud Data Integrity Using a Designated Public Verifier," in *High Performance Computing and Communications (HPCC), 2015 IEEE 7th International Symposium on Cyberspace Safety and Security (CSS), 2015 IEEE 12th International Conferen on Embedded Software and Systems (ICESS), 2015 IEEE 17th International Conference on*, 2015, pp. 1361-1366.
- [38] Y. Ren, Z. Yang, J. Wang, and L. Fang, "Attributed Based Provable Data Possession in Public Cloud Storage," in *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2014 Tenth International Conference on*, 2014, pp. 710-713.
- [39] Z. Hao and N. Yu, "A multiple-replica remote data possession checking protocol with public verifiability," in *Data, Privacy and E-Commerce (ISDPE), 2010 Second International Symposium on*, 2010, pp. 84-89.

- [40] C. C. Erway, A. Küpçü, C. Papamanthou, and R. Tamassia, "Dynamic provable data possession," *ACM Transactions on Information and System Security (TISSEC)*, 2015, vol. 17, p. 15.
- [41] D. Shrinivas, "Privacy-preserving public auditing in cloud storage security," *International Journal of computer science nad Information Technologies*, vol. 2, 2011, pp. 2691-2693.
- [42] C. Wang, Q. Wang, K. Ren, and W. Lou, "Privacy-preserving public auditing for data storage security in cloud computing," in *Infocom, 2010 proceedings ieee*, 2010, pp. 1-9.
- [43] M. Venkatesh, M. Sumalatha, and C. SelvaKumar, "Improving public auditability, data possession in data storage security for cloud computing," in *Recent Trends In Information Technology (ICRTIT), 2012 International Conference on*, 2012, pp. 463-467.
- [44] Z. Jianhong and C. Hua, "Secuirty storage in the cloud computing: a rsa-based assumption data integrity check without original data," in *Educational and Information Technology (ICEIT), 2010 International Conference on*, 2010, pp. V2-143-V2-147.
- [45] P. Li, J. Li, Z. Huang, C.-Z. Gao, W.-B. Chen, and K. Chen, "Privacy-preserving outsourced classification in cloud computing," *Cluster Computing*, 2018, vol. 21, pp. 277-286.
- [46] S. Han and J. Xing, "Ensuring data storage security through a novel third party auditor scheme in cloud computing," in *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, 2011, pp. 264-268.
- [47] B. Lee, K. Dewi, and M. F. Wajdi, "Data security in cloud computing using AES under HEROKU cloud," 2018, 27th Wireless and Optical Communication Conference (WOCC), pp. 1-5.



- [48] J. C. Benaloh, "Secret sharing homomorphisms: Keeping shares of a secret secret," in *Conference on the Theory and Application of Cryptographic Techniques*, 1986, pp. 251-260.
- [49] R. Kumaresan, A. Patra, and C. P. Rangan, "The round complexity of verifiable secret sharing: The statistical case," in *International Conference on the Theory and Application of Cryptology and Information Security*, 2010, pp. 431-447.
- [50] C.-N. Yang and J.-B. Lai, "Protecting data privacy and security for cloud computing based on secret sharing," in *Biometrics and Security Technologies (ISBAST), 2013 International Symposium on*, 2013, pp. 259-266.
- [51] R. Du, L. Deng, J. Chen, K. He, and M. Zheng, "Proofs of Ownership and Retrievalability in Cloud Storage," in *Trust, Security and Privacy in Computing and Communications (TrustCom), 2014 IEEE 13th International Conference on*, 2014, pp. 328-335.
- [52] H. Shacham and B. Waters, "Compact proofs of retrievalability," in *International Conference on the Theory and Application of Cryptology and Information Security*, 2008, pp. 90-107.
- [53] J. Yuan and S. Yu, "Proofs of retrievalability with public verifiability and constant communication cost in cloud," in *Proceedings of the 2013 international workshop on Security in cloud computing*, 2013, pp. 19-26.
- [54] Q. Zheng and S. Xu, "Secure and efficient proof of storage with deduplication," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*, 2012, pp. 1-12.

- [55] J. Li, X. Tan, X. Chen, D. S. Wong, and F. Xhafa, "Opor: Enabling proof of retrievability in cloud computing with resource-constrained devices," *Cloud Computing, IEEE Transactions on*, 2015, vol. 3, pp. 195-205.
- [56] H. Xiong, X. Zhang, D. Yao, X. Wu, and Y. Wen, "Towards end-to-end secure content storage and delivery with public cloud," in *Proceedings of the second ACM conference on Data and Application Security and Privacy*, 2012, pp. 257-266.
- [57] Q. Zheng and S. Xu, "Fair and dynamic proofs of retrievability," in *Proceedings of the first ACM conference on Data and application security and privacy*, 2011, pp. 237-248.
- [58] Y. Zhang, C. Xu, S. Yu, H. Li, and X. Zhang, "SCLPV: Secure Certificateless Public Verification for Cloud-Based Cyber-Physical-Social Systems Against Malicious Auditors," *IEEE Transactions on Computational Social Systems*, 2015, vol. 2, pp. 159-170.
- [59] H. Li, X. Lin, H. Yang, X. Liang, R. Lu, and X. Shen, "EPPDR: an efficient privacy-preserving demand response scheme with adaptive key evolution in smart grid," *IEEE Transactions on Parallel and Distributed Systems*, 2014, vol. 25, pp. 2053-2064.
- [60] V. Kumar, and G. Venkatesa Poornima, " Ensuring Data Integrity in Cloud Computing," 2012, vol. 5, pp. 12-15.
- [61] R. Singh, S. Kumar, and S. K. Agrahari, "Ensuring Data Storage Security in Cloud Computing," *IOSR Journal of Engineering*, 2012, vol. 2, p. 12.
- [62] C. Ke, Z. Huang, and M. Tang, "Supporting negotiation mechanism privacy authority method in cloud computing," *Knowledge-Based Systems*, 2013, vol. 51, pp. 48-59.
- [63] P. Syam Kumar, R. Subramanian, and D. Thamizh Selvam, "Ensuring data storage security in cloud computing using Sobol Sequence," in *Parallel Distributed and Grid Computing (PDGC), 2010 1st International Conference on*, 2010, pp. 217-222.

- [64] B. Gilburd, A. Schuster, and R. Wolff, "k-TTP: a new privacy model for large-scale distributed environments," in *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 563-568.
- [65] K. Huang, M. Xian, S. Fu, and J. Liu, "Securing the cloud storage audit service: defending against frame and collude attacks of third party auditor," *Communications, IET*, 2014, vol. 8, pp. 2106-2113.
- [66] J. Xu, "Auditing the Auditor: Secure Delegation of Auditing Operation over Cloud Storage," *IACR Cryptology EPrint Archive*, 2011, vol. 2011, p. 304.
- [67] F. Sebé, J. Domingo-Ferrer, A. Martinez-Balleste, Y. Deswarte, and J.-J. Quisquater, "Efficient remote data possession checking in critical information infrastructures," *IEEE Transactions on Knowledge and Data Engineering*, 2008, vol. 20, pp. 1034-1038.
- [68] Y. Yu, L. Niu, G. Yang, Y. Mu, and W. Susilo, "On the security of auditing mechanisms for secure cloud storage," *Future Generation Computer Systems*, 2014, vol. 30, pp. 127-132.
- [69] B. Wang, B. Li, and H. Li, "Knox: privacy-preserving auditing for shared data with large groups in the cloud," in *International Conference on Applied Cryptography and Network Security*, 2012, pp. 507-525.
- [70] K. Yang and X. Jia, "TSAS: Third-Party Storage Auditing Service," in *Security for Cloud Storage Systems*, ed: Springer, 2014, pp. 7-37.
- [71] Y. Zhu, H. Wang, Z. Hu, G.-J. Ahn, H. Hu, and S. S. Yau, "Dynamic audit services for integrity verification of outsourced storages in clouds," in *Proceedings of the 2011 ACM Symposium on Applied Computing*, 2011, pp. 1550-1557.

- [72] S. Rizvi, K. Karpinski, B. Kelly, and T. Walker, "Utilizing Third Party Auditing to Manage Trust in the Cloud," *Procedia Computer Science*, 2015, vol. 61, pp. 191-197.
- [73] S. Rizvi, K. Cover, and C. Gates, "A Trusted Third-party (TTP) based Encryption Scheme for Ensuring Data Confidentiality in Cloud Environment," *Procedia Computer Science*, 2014, vol. 36, pp. 381-386.
- [74] N. Shimbire and P. Deshpande, "Enhancing Distributed Data Storage Security for Cloud Computing Using TPA and AES Algorithm," in *Computing Communication Control and Automation (ICCUBE), 2015 International Conference on*, 2015, pp. 35-39.
- [75] P. Varalakshmi and H. Deventhiran, "Integrity checking for cloud environment using encryption algorithm," in *Recent Trends In Information Technology (ICRTIT), 2012 International Conference on*, 2012, pp. 228-232.
- [76] M. Kaur and M. Mahajan, "Using encryption algorithms to enhance the data security in cloud computing," *International Journal of Communication and Computer Technologies*, 2013, vol. 1, pp. 56-59.
- [77] K. Suresh and K. Prasad, "Security issues and Security algorithms in Cloud Computing," *International Journal of Advanced Research in Computer Science and Software Engineering*, 2012, vol. 2, pp. 12-15.
- [78] S. Rizvi, J. Ryoo, Y. Liu, D. Zazworsky, and A. Cappeta, "A centralized trust model approach for cloud computing," in *2014 23rd Wireless and Optical Communication Conference (WOCC)*, 2014, pp. 1-6.
- [79] M. Ben Haj Frej, J. Dichter, and N. Gupta, "Lightweight Accountable Privacy-Preserving Protocol Allowing the Cloud Client to Audit the Third-Party Auditor for Malicious Activities," *Applied Sciences*, 2019, vol. 9, p. 30-34.

- [80] M. Ben Haj Frej, J. Dichter, and N. Gupta, "Light-weight accountable privacy preserving (LAPP) protocol to determine dishonest role of third party auditor in cloud auditing," in *2018 IEEE International Conference on Consumer Electronics (ICCE)*, 2018, pp. 1-6.