INTEGRITY VERIFICATION IN MULTICLOUD STORAGE

By

Vishnuraj Nellithanam Raveendran

Submitted to

**The University of Roehampton**

In partial fulfilment of the requirements

for the degree of

**MASTER OF SCIENCE IN COMPUTING**

Abstract

Provable Data Possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. In this scheme, I address the construction of an efficient PDP scheme for distributed cloud storage to support the scalability of service and data migration, in which I consider the existence of multiple cloud service providers to cooperatively store and maintain the client’s data. I present a cooperative PDP (CPDP) scheme based on homomorphic verifiable response and hash index hierarchy. I prove the security of my scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero-knowledge properties. In addition, I articulate performance optimization mechanisms for my scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. Experiments show that the solution introduces lower computation and communication overheads in comparison with non-cooperative approaches.

The aim of the project is to analyse the challenges faced by the user about the integrity and security of a file they store in the cloud website, which was gained through a variety of research methodologies. Based on the results of these research methods, *Integrity Verification in Multi-Cloud Storage* is a website that allows the users to set their own security level for their uploading file. Although there exist similar websites which allow cloud storage the methodology used behind the development of this website makes it unique i.e. Multiple Replica-Provable Data Possession (MR-PDP).

Using MR-PDP to store t replicas is computationally much more efficient than using a single-replica PDP scheme to store the separate, unrelated files (e.g., by encrypting each file separately prior to storing it). Another advantage of MR-PDP is that it can generate further replicas on demand, at little expense, when some of the existing replicas fail. The generation of replicas is on demand by the user’s request that is based on the security choice selected by the user at the time of file upload. The user can choose three options Low, Medium, High at the time of file upload. The uploaded file is divided in to N blocks of different sizes to achieve the efficiency in storage and is also used to improve security, here Represent the number of clouds we are using. Low means the file is divided into blocks (here 3), and each block is stored in N different location of the single cloud. Medium means the file is divided into N blocks and each block is stored in N different clouds which improves the security of data but reduce the availability. High means the file is divided into N blocks and each N block is stored in N different clouds that are we are keeping the replicas of file in N Different clouds. The system maintains a download count to dynamically create the replicas in accordance with the user demand.

Signed (apply signature below)

**Declaration**

I hereby certify that this report constitutes my own work, that where the language of others is used, quotation marks so indicate, and that appropriate credit is given where I have used the language, ideas, expressions, or writings of others.

I declare that this report describes the original work that has not been previously presented for the award of any other degree of any other institution.

**Date:** Enter the date here

**Vishnuraj Nellithanam Raveendran**

Acknowledgements

Table of Contents

[**1.** **Introduction** 1](#_Toc163798953)

[Research Question or Problem statement 3](#_Toc163798954)

[Aims 3](#_Toc163798955)

[Objectives 3](#_Toc163798956)

[Legal, Social, Ethical and Professional Considerations 4](#_Toc163798957)

[Background 4](#_Toc163798958)

[Report overview 4](#_Toc163798959)

[**2.** **Literature - Technology Review** 5](#_Toc163798960)

[Literature Review 5](#_Toc163798961)

[Technology Review 5](#_Toc163798962)

[**3.** **Methodology** 6](#_Toc163798963)

[Project Management Describe which project management tools or techniques that you are using and comment on why you have made the choice(s). 6](#_Toc163798964)

[**4.** **Implementation** 8](#_Toc163798965)

[**5.** **Results** 9](#_Toc163798966)

[Evaluation 9](#_Toc163798967)

[Related Work 9](#_Toc163798968)

[**6.** **Conclusion** 10](#_Toc163798969)

[Reflection 10](#_Toc163798970)

[Future Work 10](#_Toc163798971)

[**7.** **References** 11](#_Toc163798972)

[**8.** **Appendices** 12](#_Toc163798973)

# **Introduction**

The project entitled INTEGRITY VERIFICATION IN MULTICLOUD STORAGE is an implementation of IEEE research journals *Cooperative Provable Data Possession for Integrity Verification in Multi-Cloud Storage,Ref:( Yan Zhu, Hongxin Hu, Gail-Joon Ahn, Senior Member, IEEE, )Meng yang Yu* and *MR-PDP: Multiple-Replica Provable Data Possession, Reza Curtmola, Reza Curtmola, Osama Khan, Randal Burns, Giuseppe Ateniese.* This is a system that combines the advantages of both CPDP (Cooperative PDP) and MRPDP (Multiple Replica PDP). The new proposed web application is a system that manages user’s data in multiple cloud storage by ensuring the integrity and availability of user’s data. To describe this system, we should have to describe both MRPDP and CPDP.

Provable data possession (PDP) is a technique for ensuring the integrity of data in storage outsourcing. In this scheme, we address the construction of an efficient PDP scheme for distributed cloud storage to support the scalability of service and data migration, in which we consider the existence of multiple cloud service providers to cooperatively store and maintain the clients’ data. We present a cooperative PDP (CPDP)[1] scheme based on homomorphic verifiable response and hash index hierarchy. We prove the security of our scheme based on multi-prover zero-knowledge proof system, which can satisfy completeness, knowledge soundness, and zero-knowledge properties. In addition, we articulate performance optimization mechanisms for our scheme, and in particular present an efficient method for selecting optimal parameter values to minimize the computation costs of clients and storage service providers. Our experiments show that our solution introduces lower computation and communication overheads in comparison with non-cooperative approaches.

Many storage systems rely on replication to increase the availability and durability of data on untrusted storage systems. At present, such storage systems provide no strong evidence that multiple copies of the data are actually stored. Storage servers can collude to make it look like they are storing many copies of the data, whereas in reality they only store a single copy. We address this shortcoming through multiple-replica provable data possession (MR-PDP) [2]: A provably-secure scheme that allows a client that stores t replicas of a file in a storage system to verify through a challenge-response protocol that each unique replica can be produced at the time of the challenge and that the storage system uses t times the storage required to store a single replica. MR-PDP extends previous work on data possession proofs for a single copy of a file in a client/server storage system. Using MR-PDP to store t replicas is computationally much more efficient than using a single-replica PDP scheme to store t separate, unrelated files (e.g., by encrypting each file separately prior to storing it). Another advantage of MR-PDP is that it can generate further replicas on demand, at little expense, when some of the existing replicas fail. The generation of replicas is on demand by the user’s request that is based on the security choice selected by the user at the time of file upload. The user can choose three options Low, Medium, High at the time of file upload. The uploaded file is divided in to N blocks of different sizes to achieve the efficiency in storage and is also used to improve security, here N represent the number of clouds we are using. Low means the file is divided into N blocks (here 3), and each block is stored in N different location of the single cloud. Medium means the file is divided into N blocks and each block is stored in N different clouds which improves the security of data but reduce the availability. High means the file is divided into N blocks and each N block is stored in N different clouds that are we are keeping the replicas of file in N different clouds. The system maintains a download count to dynamically create the replicas in accordance with the user’s demand.

The system which consists of three users namely User who have the access rights to upload, download and delete file, TPA (Third Party Auditor) who verifies the files that are uploaded by the registered user and the user can download the file only after this verification, Admin who own the system and who have the full access right, can create or delete TPAs and can view the uploaded files and details about the uploads. A single cloud can have different TPA’s, and the workload is divided among by using the random function to select the corresponding files from the cloud. The creation and deletion of TPA is based on the workload and efficiency of TPA which is monitored by the administrator. The data uploaded by the user is temporarily stored in an encrypted form by using the homomorphic encryption algorithm. We can use any kind of encryption algorithms along with these applications, but it is better to choose a zero-knowledge proof algorithm. This uses an encryption key which is automatically supplied to the user at the time of file upload. The data is stored in cloud only after it is verified by TPA. The actual storage of data is in an encrypted form called Meta Data, which ensures additional security measure for the cloud data. The user gets the original file when he/she downloads the needed file from the cloud storage, which ensures the integrity of data. The user is unaware of the background processes. This system reduces the overload of admin by creating TPAs. The TPA can be of any number for each cloud depending on the number of clouds we are using.

## Research Question or Problem statement

There exist various tools and technologies that help cloud providers construct a distributed cloud storage platform for managing clients’ data. However, if such an important platform is vulnerable to security attacks, it would bring irretrievable losses to the clients. For example, the confidential data in an enterprise may be illegally accessed through a remote interface provided by a multi-cloud, or relevant data and archives may be lost or tampered with when they are stored into an uncertain storage pool outside the enterprise. Therefore, it is indispensable for cloud service providers to provide security techniques for managing their storage services. Moreover another limitations of the existing system is that it is not suitable for multicloud storage services.

To check the availability and integrity of outsourced data in cloud storages, researchers have proposed two basic approaches called Provable Data Possession and Proofs of Retrievability. Ateniese et al. [2] first proposed the PDP model for ensuring possession of files on untrusted storages and provided an RSA-based scheme for a static case that achieves the communication cost. They also proposed a publicly verifiable version, which allows anyone, not just the owner, to challenge the server for data possession. They proposed a lightweight PDP scheme based on cryptographic hash function and symmetric key encryption, but the servers can deceive the owners by using previous metadata or responses due to the lack of randomness in the challenges. The numbers of updates and challenges are limited and fixed in advance and users cannot perform block insertions anywhere.

## Aims

1. To create a platform that enable the user to store his data files of any format (image, audio, video, document) based on his desired security level
2. To discover and address the issues that users face during the instance of cloud failure by creating multiple replicas of the files without causing data redundancy
3. To assess the website's usability and efficacy

## Objectives

1. to assess which content management systems are best suited for solving this particular challenge
2. to develop qualitative personas based on user research
3. to design the interface and procedures using application design techniques, like wireframes; and to integrate these designs into the application
4. to build a database in order to appropriately store, display, and retrieve all user data
5. to use usability tests to assess the precision of a web page's linking and the task execution's functionality

## Legal, Social, Ethical and Professional Considerations

A project's success in the linked and fast changing world of today is determined by more than just its financial or technical results. Instead, it is becoming more and more crucial to think about how a project will affect different stakeholders, society as a whole, and the environment. Therefore, by looking at the project's legal, ethical, social, and professional factors, this report seeks to address the project's multiple aspects.

## Legal Consideration

## Background

## Report overview

# **Literature - Technology Review**

## Literature Review

Websites for cloud storage have completely changed how people and companies exchange, store, and access data. With the help of these platforms, users can store files remotely and access them from any internet-connected device. Now let's explore the overview and background of cloud storage websites, including some important citations and pictures.

**Introduction of Cloud Storage**

Data stored online in virtualized storage pools, typically hosted by third parties, is referred to as cloud storage. Files like documents, pictures, videos, and more can be uploaded by users to distant servers that cloud storage companies maintain. This makes it possible to conveniently access data from any location with internet connectivity, doing away with the need for external hard disks or USB drives, among other physical storage devices.

Cloud storage websites have their roots in the early internet days and the development of remote storage solutions. This is an outline in chronological order:

* The late 1990s and early 2000s: The rise of online storage services:
  + Online storage services like Xdrive, iDrive, and Box.net (now Box) emerged as a result of the internet's growing popularity in the late 1990s and early 2000s.
  + With their constrained storage capacities and simple file-sharing features, these services were mainly aimed at individuals and small enterprises.
* 2006: AWS S3 (Amazon Web Services):
  + An important turning point in cloud storage history was reached in 2006 when Amazon Web Services (AWS) unveiled Amazon S3 (Simple Storage Service).
  + A lot of subsequent cloud storage solutions were built on the scalable and dependable cloud storage architecture that Amazon S3 gave developers.
* 2007: Dropbox - Increasing Consumer Access to Cloud Storage:
  + Dropbox was established in 2007 by Drew Houston and Arash Ferdowsi with the goal of making file synchronization and sharing easier for users.
  + Dropbox's easy-to-use interface and seamless device syncing contributed to the general public's adoption of cloud storage.
* 2010s: Development and Widening:
  + Cloud storage services kept growing and changing in the 2010s to meet the demands of different businesses.
  + After integrating cloud storage with their own ecosystems, Google Drive (2012), Microsoft OneDrive (previously SkyDrive), and Apple iCloud (2011) emerged as important players.
  + Solutions with an enterprise focus, such as Box (2005) and Dropbox for Business (2013), offered corporations improved security and teamwork functionalities.
* 2015: Cloud-Native Storage Solutions Are Introduced:
  + The increasing acceptance of cloud computing led to the rise in popularity of cloud-native storage solutions.
  + Businesses using cloud infrastructure have access to scalable and affordable storage alternatives thanks to services like Google Cloud Storage, Microsoft Azure Blob Storage, and IBM Cloud Object Storage.
* Today's Outlook: Innovation and Consolidation
  + With continuous provider consolidation and continual innovation in features and capabilities, the cloud storage market is still changing.

The varied needs of contemporary enterprises are being met by hybrid cloud storage solutions, which combine cloud storage and on-premises infrastructure. These solutions are growing in popularity.

In general, the development of storage technology and the growing use of remote storage options in both personal and professional contexts are reflected in the history of cloud storage websites. The way we store, access, and share data has changed dramatically as a result of cloud storage, from its modest beginnings in the late 1990s to the wide range of services that are offered today.

**Challenges of Existing System**

To manage the data of their clients, cloud providers might build a distributed cloud storage platform with the aid of multiple tools and technologies. Nonetheless, clients would suffer irreversible losses if such a significant platform was open to security breaches. For instance, sensitive information kept in an unreliable storage pool outside of the company may be lost or altered, or confidential data within an organization may be unlawfully accessed via a multi-cloud's external interface. As a result, cloud service providers must offer security methods for overseeing their storage offerings. Furthermore, the inadequacy of the current infrastructure for multi-cloud storage services is another drawback.

**Proposed System**

Ateniese et al. [2] focuses on addressing the shortcomings of storage systems that rely on replication to ensure the availability and durability of data. The authors introduce a provably-secure scheme called multiple-replica provable data possession (MR-PDP) that allows a client to verify the existence of multiple copies of stored data and enables cost-efficient generation of further replicas on demand. The paper utilizes a challenge-response protocol to demonstrate the ability to produce unique replicas at the time of challenge and ensure that the storage system uses t times the storage required to store a single replica. The MR-PDP scheme is shown to be computationally efficient and advantageous compared to using a single-replica PDP scheme to store multiple separate, unrelated files. The paper presents detailed cryptographic constructs and algorithms for the MR-PDP scheme, highlighting its efficiency, scalability, and security. The proposed scheme overcomes the limitations of existing replication systems and provides a practical solution for ensuring the existence and uniqueness of multiple replicas in distributed and untrusted storage systems.

The authors also compare the performance of MR-PDP with the traditional single-replica PDP scheme and an alternative solution called ENC-PDP. The paper provides insights into the computational efficiency of MR-PDP, demonstrating its ability to reduce the overall pre-processing costs by amortizing these costs over multiple replicas. The paper also evaluates the challenge phase performance, showing that MR-PDP places no additional burden on the storage server and has a scalable resource. Furthermore, the paper discusses the practical implications of MR-PDP guarantees and relevant limitations. The authors acknowledge the contributions and insightful comments of other researchers in the field. Overall, the paper provides a comprehensive framework for addressing the shortcomings of existing storage systems relying on replication and presents a practical and efficient solution in the form of the MR-PDP scheme.

## Technology Review

In a time of rapid technological change, it is critical to stay up to date on the newest developments and make sure that technology solutions are in line with organizational goals. Its goal is to give project stakeholders a thorough understanding of the technologies used in the project, emphasizing their advantages, disadvantages, and potential areas for development. We look for chances for optimization, enhancement, and future-proofing by thoroughly reviewing the technical infrastructure to make sure our company remains competitive and successful.

**Technologies Reviewed**

Content Management Systems (CMS) are software applications that enable users to create, manage, and publish digital content on the web without requiring advanced technical knowledge. These systems provide a user-friendly interface for adding, editing, and organizing content, making website management accessible to individuals and organizations of all sizes.

* WordPress:

The most popular content management system (CMS) in the world, WordPress is renowned for its adaptability, ease of use, and vast ecosystem of plugins and themes. WordPress was first introduced in 2003 as a blogging platform and has since developed into a comprehensive content management system (CMS) that powers millions of websites, including portfolios, e-commerce sites, blogs, and business websites. Its user-friendly design makes it simple for users to produce and share content, and its extensive plugin library expands functionality to suit a variety of needs, including social media integration, SEO optimization, and e-commerce features. Furthermore, users can quickly alter the appearance of their websites with WordPress's theme system. WordPress has a sizable and vibrant community of developers, designers, and users. As website owners' demands change, WordPress adapts by releasing updates and enhancements on a regular basis.

* Drupal:

Developers and businesses use Drupal because of its scalability, security, and flexibility. It is a powerful and highly configurable content management system. Drupal is an open-source platform that was founded in 2001 that enables users to create sophisticated websites and apps, from government portals to corporate intranets. Because of its modular design, developers can build their own or combine pre-configured modules to create specific features. Drupal is suited for websites with complex content structures because of its taxonomy system, which makes it easy for users to easily organize and classify material. Drupal is a reliable option for websites managing sensitive data because of its emphasis on security, which is further demonstrated by its devoted security team and regular security updates. Drupal's versatility and extensibility make it a favourite choice for projects demanding robust performance and advanced customisation, even though its learning curve may be severe for newcomers.

* Joomla

Joomla is another well-liked open-source content management system that strikes a mix between flexibility and usability, making it appropriate for a variety of websites, ranging from business portals to individual blogs. Since its first release in 2005 as a clone of the Mambo CMS, Joomla has developed into a robust platform with a thriving user and developer community. Joomla's user-friendly interface is one of its best qualities; it makes it simple for non-technical users to manage content, alter layouts, and add new features. Thousands of extensions make up Joomla's ecosystem, which offers extra features and functionalities including social networking, e-commerce, forums, and more. Furthermore, Joomla is a popularity among developers and designers alike because of its template system, which makes it possible to create distinctive and eye-catching website designs. Joomla has good performance right out of the box, and caching, SEO, and speed enhancements are all accessible. Joomla's ease of use and robust community support make it a desirable choice for individuals and businesses seeking to construct professional-looking websites without requiring advanced technical knowledge, even though it may not be as flexible or developer-friendly as Drupal.

**Technologies used for the *Integrity verification in Multicloud Storage* application**

Front-end Frameworks

With Visual Studio, frontend development for ASP.NET requires integrating a number of technologies, including as HTML, CSS, and JavaScript. These are frequently enhanced by frontend frameworks like jQuery or Bootstrap. Microsoft's ASP.NET is a powerful online application framework that is used to create dynamic webpages and web apps. Developers may expedite the frontend development process by combining Visual Studio, Microsoft's integrated development environment (IDE), with capabilities including code editing, debugging, and project management.

The foundation of the frontend is HTML, which gives web pages their structure and meaningful markup. HTML is used by developers to specify the forms, headings, paragraphs, images, and links on a webpage. Code editing capabilities in Visual Studio, such IntelliSense and code snippets, increase productivity by offering recommendations and short cuts for quickly producing HTML code.

The presentation of HTML elements can be styled using CSS (Cascading Style Sheets), which allows for control over elements including layout, color, font, and spacing. The CSS editor in Visual Studio provides several features for handling stylesheets, such as live preview, auto-completion, and syntax highlighting. With CSS frameworks like Bootstrap, developers can create mobile-friendly and aesthetically pleasing websites more quickly by employing pre-designed components and responsive layouts.

JavaScript is an essential component of frontend development, providing dynamic and interactive features on webpages, in addition to HTML and CSS. Rich user experiences can be more easily created with Visual Studio's support for JavaScript development, which includes tools like code navigation, debugging, and IntelliSense. Moreover, well-known JavaScript libraries and frameworks like jQuery can be used by developers to streamline AJAX queries, event handling, and DOM manipulation.

All things considered, Visual Studio's ASP.NET frontend development environment provides a complete setting for creating contemporary online apps. With Visual Studio's productivity-boosting capabilities and seamless ASP.NET backend integration, developers can construct dynamic and responsive user interfaces by utilizing HTML, CSS, JavaScript, and frontend frameworks.

Backend Framework-Databases

Utilizing SQL Server's capabilities as a database management system (DBMS) in conjunction with a backend programming language or framework is the process of creating backend frameworks with SQL Server. Microsoft's SQL Server is a feature-rich relational database management system that is well-known for its scalability, performance, and extensive feature set for managing and accessing data. Developers usually use an organized process to design, create, and manage the database schema and the server-side logic that communicates with the database when creating backend frameworks using SQL Server.

Using SQL Server Management Studio (SSMS) or other comparable tools, developers first build the database schema, including tables, columns, relationships, constraints, and indexes in accordance with the application's needs. With SQL Server, developers can store and process data effectively since it supports a wide range of data types, including numeric, text, date/time, and binary. Furthermore, SQL Server comes with capabilities like functions, views, triggers, and stored procedures that let programmers create business logic and manipulate data from inside the database.

Developers combine SQL Server with their preferred backend framework or programming language, such as ASP.NET Core, Node.js, Django, or Spring Boot, after the database schema has been established. Through the use of third-party libraries or libraries supplied by the DBMS vendor, connections are made from the backend application to the SQL Server database as part of this integration.

To work with the SQL Server database, developers create queries, instructions, and transactions in the backend code. These operations include removing, inserting, updating, and querying data. By mapping database entities to objects in the backend code, ORM (Object-Relational Mapping) frameworks—such as Entity Framework for.NET or Sequelize for Node.js—provide abstraction layers that streamline database interactions, minimize the need for manual SQL queries, and improve code maintainability.

In order to guarantee data integrity, security, and dependability, developers also incorporate authentication, authorization, validation, error handling, and other backend operations within the application logic. It is easier for developers to implement access controls and safeguard sensitive data from unwanted access or manipulation thanks to SQL Server's support for technologies like role-based security, encryption, and auditing.

Developers thoroughly test the backend framework at every stage of the development process, making sure it operates as intended and performs well in a variety of scenarios using unit, integration, and end-to-end testing. Utilizing either third-party monitoring solutions or SQL Server's built-in performance monitoring tools, they additionally analyze query execution plans, indexing methods, and resource consumption metrics in order to monitor and optimize database performance.

To sum up, creating backend frameworks with SQL Server entails creating and implementing database schemas, integrating SQL Server with backend frameworks or programming languages, putting application logic for business operations and data manipulation into place, and making sure that data security, performance, and integrity are all maintained through stringent testing and optimization procedures. Building scalable and dependable backend solutions for a variety of applications and industries is still made easier with SQL Server's feature-rich features and integration capabilities.

# **Methodology**

Project Management

|  |  |
| --- | --- |
| **Project Report Delivery Schedule**  Note: Reorder the sections in the order that you plan to complete them. | Deadline Date |
| Abstract | 06-03-2024 |
| Declaration | 06-04-2024 |
| Acknowledgements | 30-04-2024 |
| Introduction | 08-04-2024 |
| Literature - Technology Review | 10-04-2024 |
| Methodology | 20-04-2024 |
| Implementation and Results   * Evaluation * Related Work | 25-04-2024 |
| Conclusion   * Reflection * Future Work | 28-04-2024 |
| References | 05-05-2024 |
| Appendices | 07-05-2024 |

Table 1: Project Report Delivery Schedule

|  |  |
| --- | --- |
| **Artefact Delivery Schedule**  Note: Reorder the activities in the order that you plan to complete them. | **Deadline Date** |
| Artefact Planning and Resourcing | 06-03-2024 |
| Artefact Procurement Activities (e.g., data collection, source framework etc.) | 25-03-2024 |
| Artefact Design | 18-04-2024 |
| Artefact Development, Deployment, Implementation | 25-04-2024 |
| Artefact Evaluation and Testing | 30-04-2024 |
| Artefact Presentation and Demonstration | 08-05-2024 |
| Artefact Screencast | 10-05-2024 |

Table 2: Artefact Delivery Schedule

# **Implementation**

# **Results**

## Evaluation

Determining an application's effectiveness and how it will solve the initial issue requires evaluation. The goal of the integrity verification issue in multicloud storage was to enable users of cloud storage websites to upload their files quickly and easily, choosing the level of security that best suited their needs. Only once the cloud third-party auditor has properly verified the file is it uploaded to the cloud. The challenges faced by users were recognized through surveys, interviews, communication, observations, and firsthand experiences, and features were built in direct reaction to address these issues. Three purposes and five objectives were established at the outset of the project to specify the steps involved in solving a specific problem.

A software testing strategy incorporates the software test case creation approach into a methodically organized sequence of actions that lead to the software's effective building. The plan offers a roadmap that details the testing procedure to be followed, when it will be planned and executed, and how much time, energy, and resources will be needed. As a result, every testing strategy needs to include test case development, test execution, test planning, and the subsequent gathering and assessment of data. A software testing strategy must to be adaptable enough to encourage a personalized method of testing. It must, however, be strict enough to encourage sensible planning and management monitoring as the project moves forward.

## Related Work

# **Conclusion**

## Reflection

## Future Work

# **References**

[1] Yan Zhu,” Cooperative Provable Data Possession for Integrity Verification in Multicloud Storage” [2012]-IEEE Xplore [Online] Available: <https://ieeexplore.ieee.org/document/6152093> [Accessed: 03-Mar-2024]

[2] Ateniese Giuseppe,” MR-PDP: Multiple-Replica Provable Data Possession” [2008]-IEEE Xplore [Online] Available: <https://ieeexplore.ieee.org/document/4595910> or <https://web.njit.edu/~crix/publications/MRPDP-ICDCS08.pdf> [Accessed: 05-Mar-2024]

# **Appendices**