

**ANALYSIS ON VOTING DATA
DEDUPLICATION TECHNIQUES IN CLOUD**

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering degree in Computer Science and Engineering

by

**D.VINUTHNA (Reg.No - 39110245)
T.MANI CHANDANA (Reg.No – 39111036)**



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

SCHOOL OF COMPUTING

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

**Accredited with Grade “A” by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI - 600119**

APRIL - 2023



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)

Accredited with Grade "A" by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **VINUTHNA.D (39110245)** who carried out the Project Phase-2 entitled "**ANALYSIS ON VOTING DATA DEDUPLICATION TECHNIQUES IN CLOUD**" under my supervision from Jan 2023 to April 2023.

Internal Guide

Dr. M.D. ANTO PRAVEENA M.E., Ph.D.,

Head of the Department

Dr. L. LAKSHMANAN, M.E., Ph.D.,



Submitted for Viva voce Examination held on 20.04.2023

Internal Examiner

External Examiner

DECLARATION

I, **D.VINUTHNA(39110245)**, hereby declare that the Project Phase-2 Report entitled **ANALYSIS ON VOTING DATA DEDUPLICATION TECHNIQUES IN CLOUD”** done by me under the guidance of **Dr M.D. ANTO PRAVEENA M.E., Ph.D.** is submitted in partial fulfillment of the requirements for the award of Bachelor of Engineering degree in **Computer Science and Engineering**.

DATE:20-04 -2023

vinuthna

PLACE: Chennai

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of SATHYABAMA** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T.Sasikala M.E., Ph. D, Dean**, School of Computing, **Dr. L. Lakshmanan M.E., Ph.D.**, Head of the Department of Computer Science and Engineering for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. M.D. Anto Praveena M.E., Ph.D.**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my phase-2 project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

Data de duplication is one of important data compression techniques for eliminating duplicate copies of repeating data, and has been widely used in cloud storage to reduce the amount of storage space and save bandwidth. To protect the confidentiality of sensitive data while supporting de duplication, the convergent encryption technique has been proposed to encrypt the data before outsourcing. To better protect data security, this project makes the first attempt to formally address the problem of authorized data de duplication. Different from traditional de duplication systems, the differential privileges of users are further considered in duplicate check besides the data itself. We also present several new de duplication constructions supporting authorized duplicate check in hybrid cloud architecture. Security analysis demonstrates that our scheme is secure in terms of the definitions specified in the proposed security model. As a proof of concept, the proposed work implements a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments using our prototype. The proposed work shows that our proposed authorized duplicate check scheme incurs minimal overhead compared to normal operations.

Chapter No	TITLE	Page No.
	ABSTRACT	1
	LIST OF FIGURES	3
1	INTRODUCTION	4
2	LITERATURE SURVEY	7
3	AIM AND SCOPE OF PROJECT	
	3.1 Existing system	9
4	REQUIREMENTS ANALYSIS	
	4.1 Feasibility Studies	10
	4.2 Software and Hardware Requirements	13
	4.3 System Use Case	14
5	DESCRIPTION OF THE PROPOSED SYSTEM	
	5.1 Methodology	18
	5.2 Architecture and design of Proposed System	19
	5.3 Description of Software	20
6	IMPLEMENTATION DETAILS	
	5.1 Algorithms	31
	5.2 Testing	33
7	CONCLUSION	
	6.1 Conclusion and future work	37
	REFERENCES	37
	APPENDIX	
	A. SOURCE CODE	40
	B. SCREENSHOTS	41
	C. RESEARCH PAPER	47

LIST OF FIGURES

FIGURE NO.	FIGURE NAME	PAGE NO.
4.1	Use Case Diagram	14
4.2	Class Diagram	15
4.3	Sequence Diagram	17
4.4	Activity Diagram	18
5.1	Architecture Diagram	19
5.3	Login page	26
5.4	Browser window	27

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Cloud computing provides seemingly unlimited “virtualized” resources to users as services across the whole Internet, while hiding platform and implementation details. Today’s cloud service providers offer both highly available storage and massively parallel computing resources at relatively low costs. As cloud computing becomes prevalent, an increasing amount of data is being stored in the cloud and shared by users with specified privileges, which define the access rights of the stored data. One critical challenge of cloud storage services is the management of the ever-increasing volume of data. To make data management scalable in cloud computing, de duplication has been a well-known technique and has attracted more and more attention recently. While distributed computing products hide locations and specifics, for Internet users they also seem to have no "virtualization" restrictions. Currently, cloud specialized companies provide the most cost-effective storage options with a wide range of low-cost equipment. How much data is present in the system when distributed computing is distinguished access to store data is granted thanks to expanded distributed storage that is provided to specific customers. This is one of the administrations for distributed storage. Cloud computing is a climbing plan that really has a drawn principal thought from each one exchange and academia. Users get benefits over the web using cloud computing. Client will use net associations of unusual gathering as opposed to purchasing or putting in them. All clients are furthermore seen as in duplicate truly investigate other than the genuine data. We moreover show a few new deduplication movements supporting endorsement duplicate present a protection a cream cloud building plan.

Data de duplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. The technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple data copies with the same content, deduplication eliminates redundant data by keeping only one physical copy and

referring other redundant data to that copy. De duplication can take place at either the file level or the block level. For filelevel de duplication, it eliminates duplicate copies of the same file. De duplication can also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files. Although data deduplication brings a lot of benefits, security and privacy concerns arise as users' sensitive data are susceptible to both insider and outsider attacks. Traditional encryption, while providing data confidentiality, is incompatible with data deduplication.

Specifically, traditional encryption requires different users to encrypt their data with their own keys. Thus, identical data copies of different users will lead to different cipher texts, making de duplication impossible. Convergent encryption has been proposed to enforce data confidentiality while making de duplication feasible.

To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found. After the proof, subsequent users with the same file will be provided a pointer from the server without needing to upload the same file. A user can download the encrypted file with the pointer from the server, which can only be decrypted by the corresponding data owners with their convergent keys.

Thus, convergent encryption allows the cloud to perform de duplication on the cipher texts and the proof of ownership prevents the unauthorized user to access the file. However, previous de duplication systems cannot support differential authorization duplicate check, which is important in many applications. In such an authorized de duplication system, each user is issued a set of privileges during system initialization. Each file uploaded to the cloud is also bounded by a set of privileges to specify which kind of users is allowed to perform the duplicate check and access the files. Before submitting his duplicate check request for some file, the user needs to take this file and his own privileges as inputs.

In order to save cost and efficiently management, the data will be moved to the storage server provider (SSP) in the public cloud with specified privileges and the de duplication technique will be applied to store only one copy of the same file. Because of privacy consideration, some files will be encrypted and allowed the duplicate check by employees with specified privileges to realize the access control. Traditional de

duplication systems based on convergent encryption, although providing confidentiality to some extent; do not support the duplicate check with differential privileges. In other words, no differential privileges have been considered in the de duplication based on convergent encryption technique. It seems to be contradicted if we want to realize both de duplication and differential authorization duplicate check at the same time.

1.2 ORGANIZATION OF THE THESIS

Chapter 1 deals with an introduction to the project where the existing system is been discussed. It also gives an overview of how de duplication technique has been implemented with the high security.

Chapter 2, a detailed description of the literature survey of the papers which are referred during the course of the project was summarized.

Chapter 3 gives a brief explanation on the aim and scope of the project. Here proposed system has been compared with the existing system. The issues in the existing system and the advantages of the proposed system are also discussed.

Chapter 4 deals with the methods and algorithms used. The hardware and software requirements are provided along with the system design, architecture and flow of overall project.

Chapter 5 deals with the system implementation of the project.

Chapter 6, the Results and Discussion along with the screenshots of each module has been depicted.

Chapter 7 deals with the summary and conclusion of the project. It also includes the future scope of the project.

CHAPTER 2

LITERATURE SURVEY

Literature survey is the most important step in software development process. Before developing the tool it is necessary to determine the time factor, economy and company strength. Once these things are satisfied, then the next step is to determine which operating system and language can be used for developing the tool. Once the programmers start building the tool the programmers need lot of external support. This support can be obtained from senior programmers, from book or from websites. Before building the system the above consideration are taken into account for developing the proposed system.

The major part of the project development sector considers and fully survey all the required needs for developing the project. For every project Literature survey is the most important sector in software development process. Before developing the tools and the associated designing it is necessary to determine and survey the time factor, resource requirement, man power, economy, and company strength. Once these things are satisfied and fully surveyed, then the next step is to determine about the software specifications in the respective system such as what type of operating system the project would require, and what are all the necessary software are needed to proceed with the next step such as developing the tools, and the associated operations they proposed an architecture that provides secure deduplication storage resisting brute force attacks, and realize it in a system called dupless. It enables clients encrypted data with an existing service. The encryption for deduplicated storage can achieve performance and space saving close to that of using the storage service with plaintext data.

There is a mechanism to reclaim space from incidental duplication to make it available for controlled file replication. This mechanism convergent encryption, which enable duplicate files to be coalesced into the space file, even if the files are encrypted with different user's keys.

It is a baseline approach in which each user holds an independent master key for encrypting the convergent keys and outsourcing them to the cloud. However, such a baseline key management scheme generates an enormous number of keys with the

increasing number of users and requires users to dedicatedly protect the master key.

They construct a private de duplication protocol based on the standard cryptographic assumptions is then presented and analyzed. They show that the private data de duplication protocol is probably secure assuming that the underlying hash function is collision-resilient, the discrete logarithm is hard and the erasure coding algorithm can erasure up to many fractions of the bits.

They design an encryption scheme that guarantees semantic security for unpopular data and provides weaker security and better storage and bandwidth benefits for popular data. This way, data de duplication can be effective for popular data, whilst semantically secure encryption protects unpopular content. We show that our scheme is secure under the Symmetric External Decisional Diffie-Hellman Assumption.

CHAPTER 3

AIM AND SCOPE OF THE PROJECT

SCOPE OF THE PROJECT

SSData de duplication techniques are widely employed to backup data and minimize network and storage overhead by detecting and eliminating redundancy among data.

OBJECTIVE

The main goal is to enable de duplication and distributed storage of the data across multiple storage servers.

PROBLEM DEFINITION

The existing system only performs the de duplication either on block level or file level. It does not provide very high security needed for the message to be transmitted. Due to this the third party or hacker may find the data that is being transmitted between the users.

3.1 EXISTING SYSTEM

Data de duplication systems, the private cloud are involved as a proxy to allow data owner/users to securely perform duplicate check with differential privileges. Such architecture is practical and has attracted much attention from researchers. The data owners only outsource their data storage by utilizing public cloud while the data operation is managed in private cloud.

Data de duplication is a specialized data compression technique for eliminating duplicate copies of repeating data in storage. The technique is used to improve storage utilization and can also be applied to network data transfers to reduce the number of bytes that must be sent. Instead of keeping multiple data copies with the same content, de duplication eliminates redundant data by keeping only one physical copy and referring other redundant data to that copy.

De duplication can take place at either the file level or the block level. For file level de duplication, it eliminates duplicate copies of the same file. De duplication can

also take place at the block level, which eliminates duplicate blocks of data that occur in non-identical files. Identical data copies of different users will lead to different cipher texts, making de duplication impossible.

3.1.1 *Disadvantages*

- Traditional encryption, while providing data confidentiality, is incompatible with data de duplication.
- Identical data copies of different users will lead to different cipher texts, making de duplication impossible.

CHAPTER 4

REQUIREMENTS ANALYSIS

Requirement analysis, also called requirement engineering, is the process of determining user expectations for a new modified product. It encompasses the tasks that determine the need for analyzing, documenting, validating and managing software or system requirements. The requirements should be documentable, actionable, measurable, testable and traceable related to identified business needs or opportunities and define to a level of detail, sufficient for system design.

4.1 FEASIBILITY STUDIES

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

- ◆ **ECONOMICAL FEASIBILITY**
- ◆ **TECHNICAL FEASIBILITY**
- ◆ **SOCIAL FEASIBILITY**

4.1.1 ECONOMICAL FEASIBILITY

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

4.1.2 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

4.1.3 SOCIAL FEASIBILITY

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

4.1.4 FUNCTIONAL REQUIREMENTS

It is a technical specification requirement for the software products. It is the first step in the requirement analysis process which lists the requirements of particular software systems including functional, performance and security requirements. The function of the system depends mainly on the quality hardware used to run the software with given functionality.

◆ Usability

It specifies how easy the system must be use. It is easy to ask queries in any format which is short or long, porter stemming algorithm stimulates the desired response for user.

◆ **Robustness**

It refers to a program that performs well not only under ordinary conditions but also under unusual conditions. It is the ability of the user to cope with errors for irrelevant queries during execution.

◆ **Security**

The state of providing protected access to resource is security. The system provides good security and unauthorized users cannot access the system there by providing high security.

◆ **Reliability**

It is the probability of how often the software fails. The measurement is often expressed in MTBF (Mean Time Between Failures). The requirement is needed in order to ensure that the processes work correctly and completely without being aborted. It can handle any load and survive and survive and even capable of working around any failure.

◆ **Compatibility**

It is supported by version above all web browsers. Using any web servers like localhost makes the system real-time experience.

◆ **Flexibility**

The flexibility of the project is provided in such a way that it has the ability to run on different environments being executed by different users.

◆ **Safety**

Safety is a measure taken to prevent trouble. Every query is processed in a secured manner without letting others to know one's personal information.

4.1.5 NON- FUNCTIONAL REQUIREMENTS

◆ **Portability**

It is the usability of the same software in different environments. The project can be run in any operating system.

◆ **Performance**

These requirements determine the resources required, time interval, throughput and everything that deals with the performance of the system.

◆ **Accuracy**

The result of the requesting query is very accurate and high speed of retrieving information. The degree of security provided by the system is high and effective.

◆ **Maintainability**

Project is simple as further updates can be easily done without affecting its stability. Maintainability basically defines that how easy it is to maintain the system. It means that how easy it is to maintain the system, analyse, change and test the application. Maintainability of this project is simple as further updates can be easily done without affecting its stability.

4.2 HARDWARE REQUIREMENT AND SOFTWARE REQUIREMENT

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and it also shows how it should be implemented. It specifies the speed of the system that should be used in this project. The hardware requirement for this project is mentioned below.

System	: Pentium IV 2.4 GHz
Hard Disk	: 40 GB
Floppy Drive	: 44 Mb

Monitor	: 15 VGA Colour
Ram	: 512 Mb

SOFTWARE REQUIREMENT

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team and tracking the team's progress throughout the development activity. The software requirement specifies the application software that is being used in the project. The software needed to develop this project is mentioned below.

- Operating system : Windows XP/7
- Coding Language :python

4.3 SYSTEM USE CASE

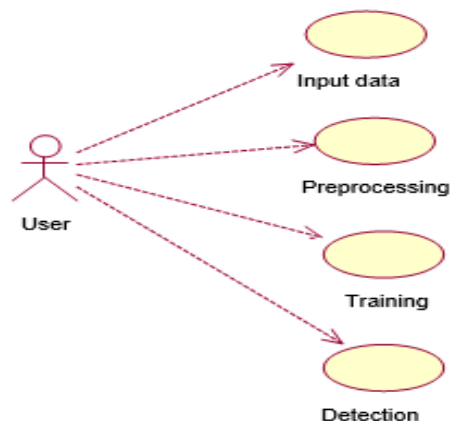


Fig 4.1 Use Case Diagram

In the given fig 4.1, A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

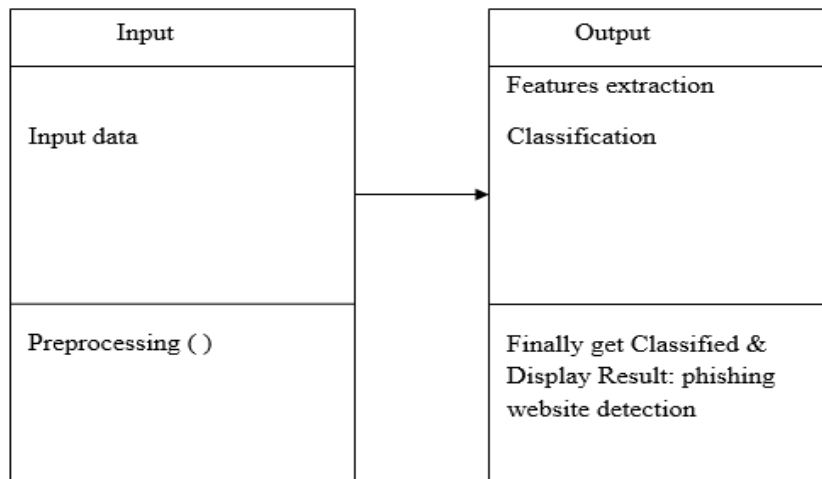


Fig 4.2 Class Diagram

In the given fig 4.2, Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modeling of objectoriented systems because they are the only UML diagrams, which can be mapped directly with object-oriented languages.

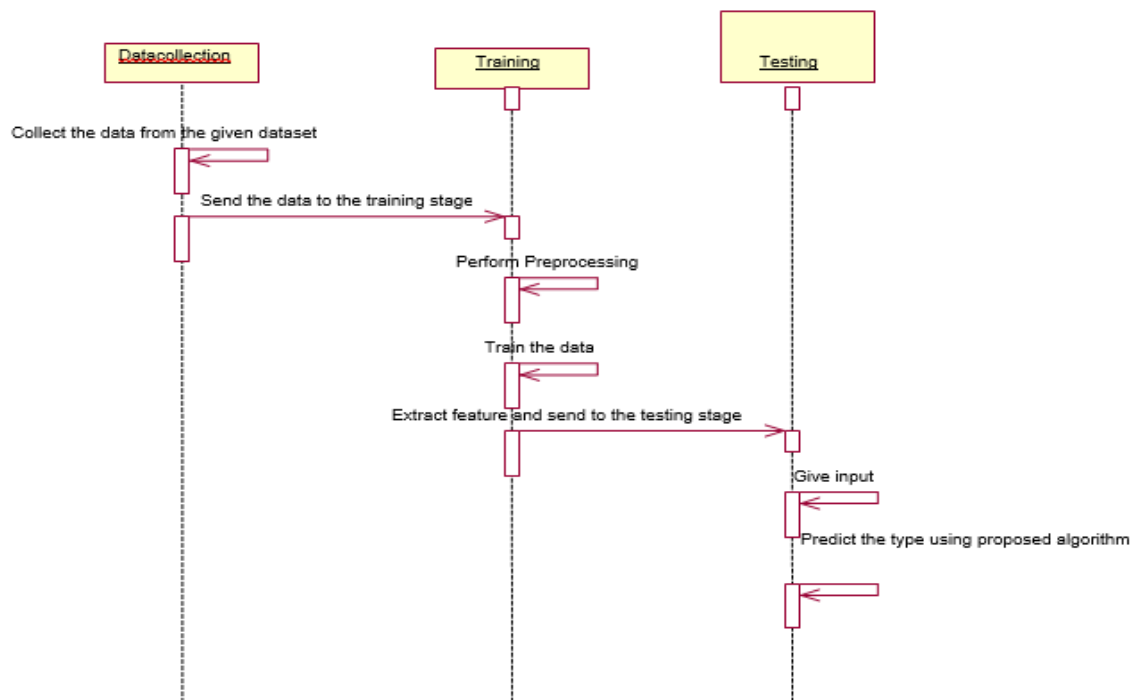


Fig 4.3 Sequence Diagram

In the given fig 4.3, A sequence diagram is a type of interaction diagram because it describes how—and in what order—a group of objects works together. These diagrams are used by software developers and business professionals to understand requirements for a new system or to document an existing process.

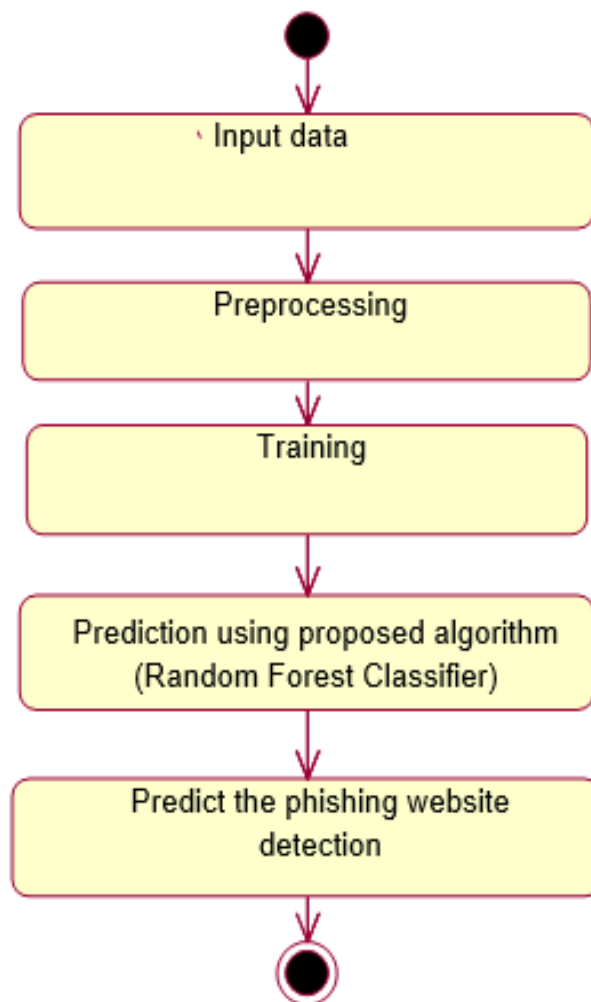


Fig 4.4 Activity Diagram

In the given fig 4.4, Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

CHAPTER 5

DESCRIPTION OF PROPOSED SYSTEM

- In this proposed work, the system enhanced with security. Specifically, it present an advanced scheme to support stronger security by encrypting the file with differential privilege keys. In this way, the users without corresponding privileges cannot perform the duplicate check. Furthermore, such unauthorized users cannot decrypt the cipher text even collude with the S-CSP. Security analysis demonstrates that our system is secure in terms of the definitions specified in the proposed security model.
- Convergent encryption has been proposed to enforce data confidentiality while making de duplication feasible. It encrypts/decrypts a data copy with a convergent key, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain the keys and send the cipher text to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies will generate the same convergent key and hence the same cipher text. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found.
- **M3 encryption**
- The algorithm itself is very complex and secure, but using it is as simple.
- The basic principle of this algorithm is character-remapping based on key self-mutation.
- The lifespan of a key is equal to the length of the key. This means that any state of the key will only be responsible for encrypting a part of the clear-text that is equal in length to the length of that version of the key before the key is self-mutated into a new version. This new version will then encrypt the next part of the clear-text, etc.
- In the overall process, you have a clear-key entered by the user which is diverted into 4 separate "threads" of different and constantly self-mutating keys. These 4 different keys are responsible for simultaneously converting the clear-text letters one letter at a time into cipher text by 2 different methods, these methods being: array remapping, and a sort of dynamic "substitution cipher". This whole process is

repeated over and over again, re-encrypting everything a number of times before the cipher-text is finalized.

- An attempt of reversing the process by a potential attacker would require figuring out the end state of 4 different keys simultaneously going backwards one mutation-version at a time. As 2 of these keys are used for array remapping, it is necessary to get the whole of these keys per letter decoded in the clear-text.

5.1.1 Advantages

- The user is only allowed to perform the duplicate check for files marked with the corresponding privileges.
- We present an advanced scheme to support stronger security by encrypting the file with differential privilege keys.
- Reduce the storage size of the tags for integrity check. To enhance the security of de duplication and protect the data confidentiality.

ARCHITECTURE AND DESIGN OF PROPOSED SYSTEM

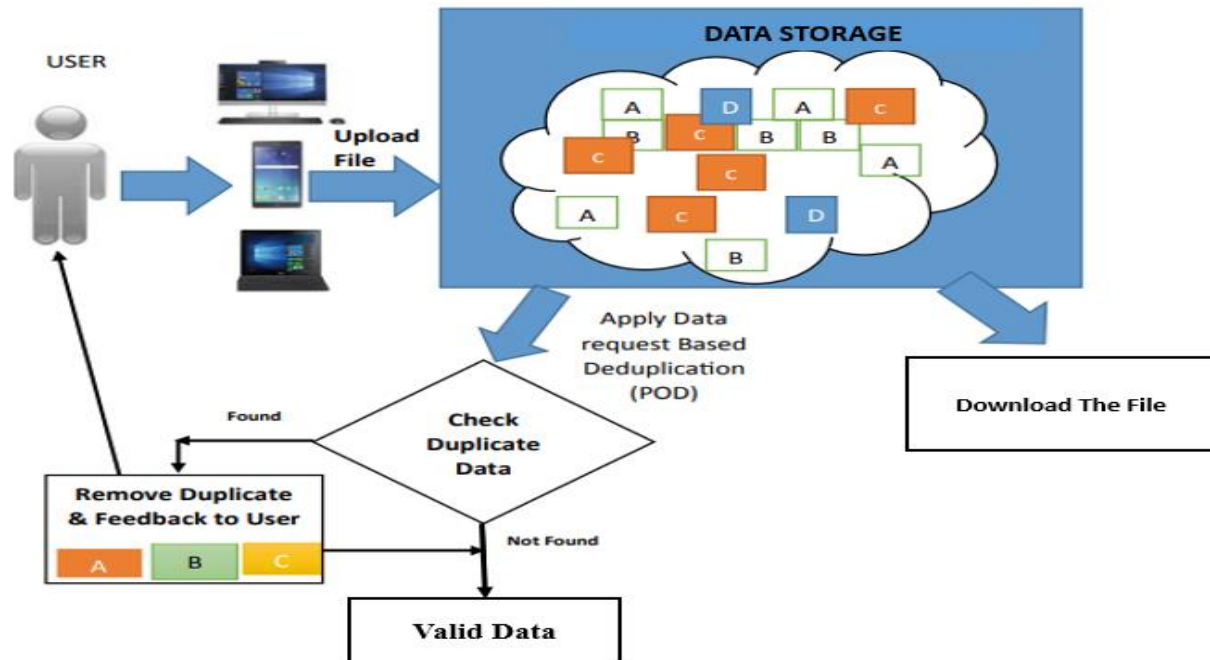


Fig. 5.1 Architecture of the system

5.3 DESCRIPTION OF SOFTWARE

Python:

Python is a high-level, interpreted, interactive and object-oriented scripting language. Python is designed to be highly readable. It uses English keywords frequently where as other languages use punctuation, and it has fewer syntactical constructions than other languages.

- **Python is Interpreted** – Python is processed at runtime by the interpreter. You do not need to compile your program before executing it. This is similar to PERL and PHP.

- **Python is Interactive** – You can actually sit at a Python prompt and interact with the interpreter directly to write your programs.
- **Python is Object-Oriented** – Python supports Object-Oriented style or technique of programming that encapsulates code within objects.
- **Python is a Beginner's Language** – Python is a great language for the beginner-level programmers and supports the development of a wide range of applications from simple text processing to WWW browsers to games.

History of Python

Python was developed by Guido van Rossum in the late eighties and early nineties at the National Research Institute for Mathematics and Computer Science in the Netherlands.

Python is derived from many other languages, including ABC, Modula-3, C, C++, Algol-68, SmallTalk, and Unix shell and other scripting languages.

Python is copyrighted. Like Perl, Python source code is now available under the GNU General Public License (GPL).

Python is now maintained by a core development team at the institute, although Guido van Rossum still holds a vital role in directing its progress.

Python Features

Python's features include –

- **Easy-to-learn** – Python has few keywords, simple structure, and a clearly defined syntax. This allows the student to pick up the language quickly.
- **Easy-to-read** – Python code is more clearly defined and visible to the eyes.
- **Easy-to-maintain** – Python's source code is fairly easy-to-maintain.
- **A broad standard library** – Python's bulk of the library is very portable and cross-platform compatible on UNIX, Windows, and Macintosh.
- **Interactive Mode** – Python has support for an interactive mode which allows interactive testing and debugging of snippets of code.

- **Portable** – Python can run on a wide variety of hardware platforms and has the same interface on all platforms.
- **Extendable** – You can add low-level modules to the Python interpreter. These modules enable programmers to add to or customize their tools to be more efficient.
- **Databases** – Python provides interfaces to all major commercial databases.
- **GUI Programming** – Python supports GUI applications that can be created and ported to many system calls, libraries and windows systems, such as Windows MFC, Macintosh, and the X Window system of Unix.
- **Scalable** – Python provides a better structure and support for large programs than shell scripting.

Apart from the above-mentioned features, Python has a big list of good features, few are listed below –

- It supports functional and structured programming methods as well as OOP.
- It can be used as a scripting language or can be compiled to byte-code for building large applications.
- It provides very high-level dynamic data types and supports dynamic type checking.
- It supports automatic garbage collection.
- It can be easily integrated with C, C++, COM, ActiveX, CORBA, and Java.

Python is available on a wide variety of platforms including Linux and Mac OS X. Let's understand how to set up our Python environment.

Getting Python

The most up-to-date and current source code, binaries, documentation, news, etc., is available on the official website of Python <https://www.python.org>.

Windows Installation

Here are the steps to install Python on Windows machine.

- Open a Web browser and go to <https://www.python.org/downloads/>.

- Follow the link for the Windows installer python-XYZ.msifile where XYZ is the version you need to install.
- To use this installer python-XYZ.msi, the Windows system must support Microsoft Installer 2.0. Save the installer file to your local machine and then run it to find out if your machine supports MSI.
- Run the downloaded file. This brings up the Python install wizard, which is really easy to use. Just accept the default settings, wait until the install is finished, and you are done.

The Python language has many similarities to Perl, C, and Java. However, there are some definite differences between the languages.

First Python Program

Let us execute programs in different modes of programming.

Interactive Mode Programming

Invoking the interpreter without passing a script file as a parameter brings up the following prompt –

```
$ python

Python2.4.3(#1,Nov112010,13:34:43)

[GCC 4.1.220080704(RedHat4.1.2-48)] on linux2

Type"help","copyright","credits"or"license"for more information.

>>>
```

Type the following text at the Python prompt and press the Enter –

```
>>>print"Hello, Python!"
```

If you are running new version of Python, then you would need to use print statement with parenthesis as in **print ("Hello, Python!")**;. However in Python version 2.4.3, this

produces the following result –

```
Hello, Python!
```

Script Mode Programming

Invoking the interpreter with a script parameter begins execution of the script and continues until the script is finished. When the script is finished, the interpreter is no longer active.

Let us write a simple Python program in a script. Python files have extension **.py**. Type the following source code in a test.py file –

```
print"Hello, Python!"
```

We assume that you have Python interpreter set in PATH variable. Now, try to run this program as follows –

```
$ python test.py
```

This produces the following result –

```
Hello, Python!
```

Flask Framework:

Flask is a web application framework written in Python. Armin Ronacher, who leads an international group of Python enthusiasts named Pocco, develops it. Flask is based on Werkzeug WSGI toolkit and Jinja2 template engine. Both are Pocco projects.

Http protocol is the foundation of data communication in world wide web. Different methods of data retrieval from specified URL are defined in this protocol.

TABLE 5.2 different http methods

S.No	Methods & Description
1	GET Sends data in unencrypted form to the server. Most common method.
2	HEAD Same as GET, but without response body
3	POST Used to send HTML form data to server. Data received by POST method is not cached by server.
4	PUT Replaces all current representations of the target resource with the uploaded content.
5	DELETE Removes all current representations of the target resource given by a URL

By default, the Flask route responds to the **GET** requests. However, this preference can be altered by providing methods argument to **route()** decorator.

In order to demonstrate the use of **POST** method in URL routing, first let us create an HTML form and use the **POST** method to send form data to a URL.

Save the following script as login.html

```
<html>

<body>

<formaction="http://localhost:5000/login"method="post">

<p>Enter Name:</p>

<p><inputtype="text"name="nm"/></p>

<p><inputtype="submit"value="submit"/></p>

</form>

</body>

</html>
```

Now enter the following script in Python shell.

```
from flask import Flask, redirect, url_for, request

app=Flask(__name__)

@app.route('/success/<name>')

def success(name):

return'welcome %s'% name

@app.route('/login',methods=['POST','GET'])

def login():

if request.method=='POST':

user=request.form['nm']

return redirect(url_for('success',name= user))
```

```
else:

user=request.args.get('nm')

return redirect(url_for('success',name= user))

if __name__ == '__main__':

app.run(debug =True)
```

After the development server starts running, open **login.html** in the browser, enter name in the text field and click **Submit**.

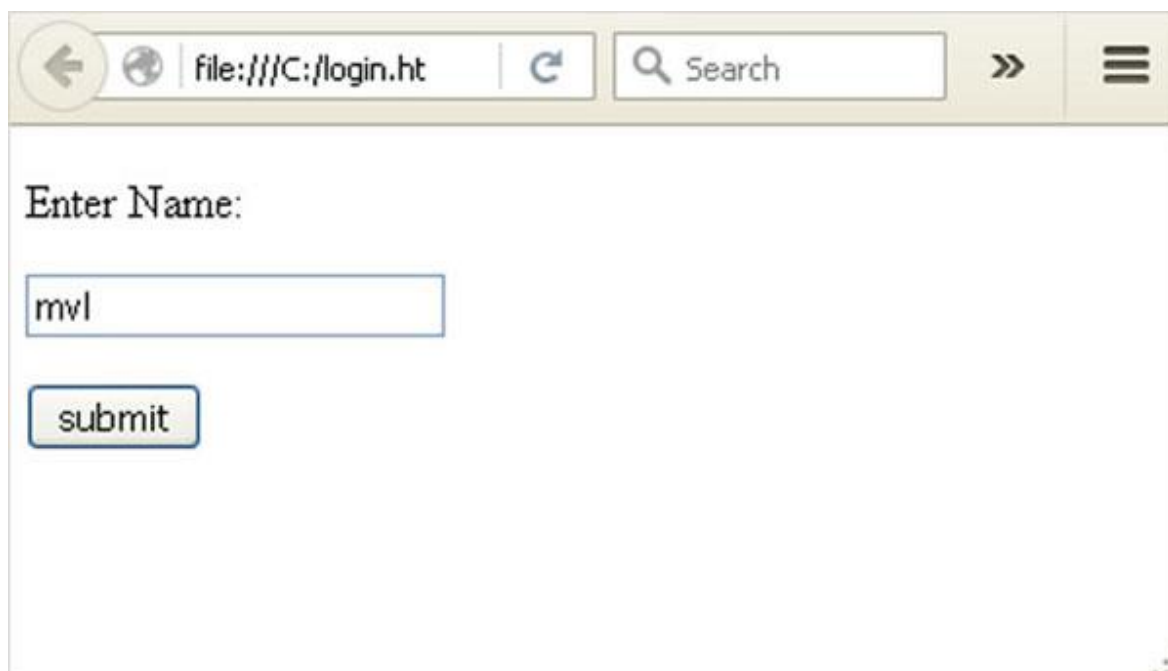
A screenshot of a web browser window. The address bar shows the file path 'file:///C:/login.ht'. The page content includes the text 'Enter Name:', a text input field containing the text 'mvl', and a 'submit' button.

Fig 5.3 login page

In the given fig 5.3, login page allows a user to gain access to an application by entering their username and password or by authenticating using a social media login.

Form data is POSTed to the URL in action clause of form tag.

http://localhost/login is mapped to the **login()** function. Since the server has received data by **POST** method, value of 'nm' parameter obtained from the form data is obtained by –


```
user = request.form['nm']
```

It is passed to **‘/success’** URL as variable part. The browser displays a **welcome** message in the window.

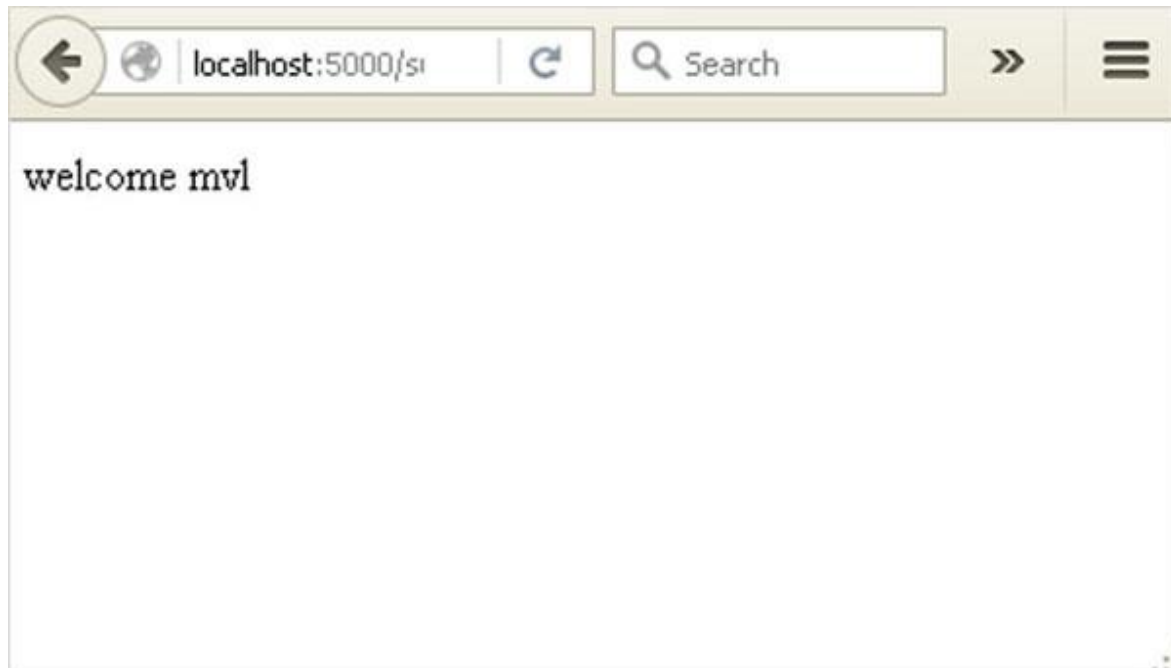


Fig 5.4 Browser window

In the given fig 5.4, The browser window is associated with a hierarchical collection of objects. The main object of interest for us would be the document object.

Change the method parameter to **‘GET’** in **login.html** and open it again in the browser. The data received on server is by the **GET** method. The value of **‘nm’** parameter is now obtained by –

```
User = request.args.get('nm')
```

Here, **args** is dictionary object containing a list of pairs of form parameter and its corresponding value. The value corresponding to **‘nm’** parameter is passed on to **‘/success’** URL as before.

5.4 MODULE OVERVIEW

- User Module
- Server start up and Upload file
- Secure DE duplicate System
- Download file

5.4.1 user module

In this module, Users are having authentication and security to access the detail which is presented in the ontology system. Before accessing or searching the details user should have the account in that otherwise they should register first. At the very least, you need to provide an email address, username, password, display name, and whatever profile fields you have set to required..

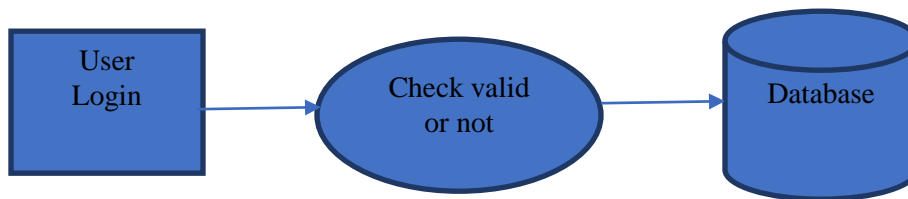


Fig. 4.2 user module

5.4.2 Server start up and upload file

The user can start up the server after cloud environment is opened. Then the user can upload the file to the cloud.



Fig. 4.3 server start up and upload file

5.4.3 Secure de duplication system

To support authorized de duplication the tag of a file F will be determined by the file F and the privilege. To show the difference with traditional notation of tag, we call it file token instead. To support authorized access a secret key K_P will be bounded with a privilege p to generate a file Token. De duplication exploits identical content, while encryption attempts to make all content appear random; the same content encrypted with two different keys results in very different cipher text. Thus, combining the space efficiency of de duplication with the secrecy aspects of encryption is problematic.

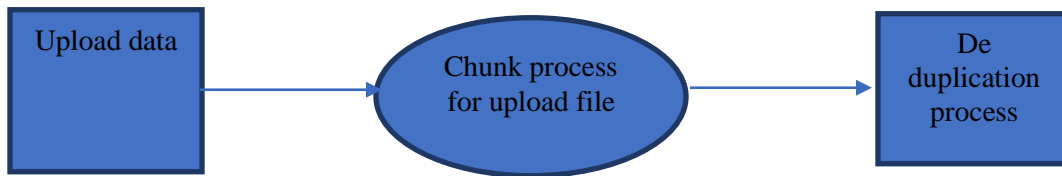


Fig 4.4 Secure de duplication system

5.5.4 Download file

After the cloud storage, the user can download the file based on key or token. Once the key request was received, the sender can send the key or he can decline it. With this key and request id which was generated at the time of sending key request the receiver can decrypt the message.



Fig 4.5 Download file

CHAPTER 6

IMPLEMENTATION DETAILS

6.1 PROPOSED ALGORITHM

ENCRYPTION

M3 encryption:

- The algorithm itself is very complex and secure, but using it is as simple.
- The basic principle of this algorithm is character-remapping based on key self-mutation.
- The lifespan of a key is equal to the length of the key. This means that any state of the key will only be responsible for encrypting a part of the clear-text that is equal in length to the length of that version of the key before the key is self-mutated into a new version. This new version will then encrypt the next part of the clear-text, etc.
- In the overall process, you have a clear-key entered by the user which is diverted into 4 separate "threads" of different and constantly self-mutating keys. These 4 different keys are responsible for simultaneously converting the clear-text letters one letter at a time into cipher text by 2 different methods, these methods being: array remapping, and a sort of dynamic "substitution cipher". This whole process is repeated over and over again, re-encrypting everything a number of times before the cipher-text is finalized.
- An attempt of reversing the process by a potential attacker would require figuring out the end state of 4 different keys simultaneously going backwards one mutation-version at a time. As 2 of these keys are used for array remapping, it is necessary to get the whole of these keys per letter decoded in the clear-text.

DECRYPTION ALGORITHM

Data Encryption Standard (DES)

This stands for Data Encryption Standard and it was developed in 1977. It was the

first encryption standard to be recommended by NIST (National Institute of Standards and Technology). DES is 64 bits key size with 64 bits block size. Since that time, many attacks and methods have witnessed weaknesses of DES, which made it an insecure block cipher.

Algorithm:

function DES_Encrypt (M, K)

where $M = (L, R)$

$M \leftarrow IP(M)$

For round $\leftarrow 1$ to 16 do

$K \leftarrow SK(K, \text{round})$

$L \leftarrow L \text{ xor } F(R, K_i)$

swap(L, R)

end

swap (L, R)

$M \leftarrow IP^{-1}(M)$

return M

End

6.1.2 chunking technique for deduplication

Chunking is a process to split a file into smaller files called chunks. In some applications, such as remote data compression, data synchronization, and data deduplication, chunking is important because it determines the duplicate detection performance of the system. Content-defined chunking (CDC) is a method to split files into variable length chunks, where the cut points are defined by some internal features of the files. Unlike fixed-length chunks, variable-length chunks are more resistant to byte shifting. Thus, it increases the probability of finding duplicate chunks within a file and between files. However, CDC algorithms require additional computation to find the cut points which might be computationally expensive for some applications. In our previous work (Widodo et al., 2016), the hash-based CDC algorithm used in the system took more process time than other processes in the deduplication system. This proposed work shows high throughput hash-less chunking. Instead of using hashes, RAM uses bytes value to declare the cut points. The algorithm utilizes a fix-sized window and a variable-sized window to find a maximum-valued byte which is the cut point. The

maximum-valued byte is included in the chunk and located at the boundary of the chunk. This configuration allows RAM to do fewer comparisons while retaining the CDC property. We compared RAM with existing hash-based and hash-less deduplication systems. The experimental results show that our proposed algorithm has higher throughput and bytes saved per second compared to other chunking algorithms.

6.2 SYSTEM DESIGN AND TESTING PLAN

INPUT DESIGN

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processing can be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

- What data should be given as input?
- How the data should be arranged or coded?
- The dialog to guide the operating personnel in providing input.
- Methods for preparing input validations and steps to follow when error occur.

OUTPUT DESIGN

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system's relationship to help user decision-making.

The output form of an information system should accomplish one or more of the

following objectives.

- Convey information about past activities, current status or projections of the
- Future.
- Signal important events, opportunities, problems, or warnings.
- Trigger an action.
- Confirm an action.

6.3 Test plan

Software testing is the process of evaluation a software item to detect differences between given input and expected output. Also to assess the feature of a software item. Testing assesses the quality of the product. Software testing is a process that should be done during the development process. In other words software testing is a verification and validation process.

Verification

Verification is the process to make sure the product satisfies the conditions imposed at the start of the development phase. In other words, to make sure the product behaves the way we want it to.

Validation

Validation is the process to make sure the product satisfies the specified requirements at the end of the development phase. In other words, to make sure the product is built as per customer requirements.

Basics of software testing

There are two basics of software testing: black box testing and white box testing.

Black box Testing

Black box testing is a testing technique that ignores the internal mechanism of the

system and focuses on the output generated against any input and execution of the system. It is also called functional testing.

White box Testing

White box testing is a testing technique that takes into account the internal mechanism of a system. It is also called structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

Types of testing

There are many types of testing like

- Unit Testing
- Integration Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Acceptance Testing
- Regression Testing
- Beta Testing

Unit Testing

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing expected output against given input.

Integration Testing

Integration testing is testing in which a group of components are combined to produce output. Also, the interaction between software and hardware is tested in integration testing if software and hardware components have any relation. It may fall under both white box testing and black box testing.

Functional Testing

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

System Testing

System testing is the testing to ensure that by putting the software in different environments (e.g., Operating Systems) it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

Stress Testing

Stress testing is the testing to evaluate how system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

Performance Testing

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in performance requirements. It falls under the class of black box testing.

Usability Testing

Usability testing is performed to the perspective of the client, to evaluate how the GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing.

CHAPTER 7

CONCLUSION

In this paper, the notion of authorized data deduplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new deduplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

7.1 FUTURE WORK

Future work will aim to develop a system that can learn by itself about new types of deduplication techniques in cloud for storage. The scope of this approach not only helps in adding more enhanced features but also updating the existing features to improve its importance level to make deduplication more efficient and reduce the false positive rate to a large extent

REFERENCES

- [1] P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted deduplication. In Proc. of USENIX LISA, 2010
- [2] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.
- [3] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296–312, 2013.
- [4] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.
- [5] M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002.
- [6] M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.
- [7] M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296– 312, 2013.
- [8] M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.
- [9] M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002.
- [10] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.
- [11] S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.

- [12] J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In ICDCS, pages 617–624, 2002.
- [13] D. Ferraiolo and R. Kuhn. Role-based access controls. In 15th NIST-NCSC National Computer Security Conf., 1992.
- [14] S. Halevi, D. Harnik, B. Pinkas, and A. ShulmanPeleg. Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis, and V. Shmatikov, editors, ACM Conference on Computer and Communications Security, pages 491–500. ACM, 2011.
- [15] J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou. Secure deduplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2013.
- [16] C. Ng and P. Lee. Revdedup: A reverse deduplication storage system optimized for reads to latest backups. In Proc. of APSYS, Apr 2013.
- [17] W. K. Ng, Y. Wen, and H. Zhu. Private data deduplication protocols in cloud storage. In S. Ossowski and P. Lecca, editors, Proceedings of the 27th Annual ACM Symposium on Applied Computing, pages 441–446. ACM, 2012.
- [18] R. D. Pietro and A. Sorniotti. Boosting efficiency and security in proof of ownership for deduplication. In H. Y. Youm and Y. Won, editors, ACM Symposium on Information, Computer and Communications Security, pages 81–82. ACM, 2012.
- [19] S. Quinlan and S. Dorward. Venti: a new approach to archival storage. In Proc. USENIX FAST, Jan 2002. [18] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui. A secure cloud backup system with assured deletion and version control. In 3rd International Workshop on Security in Cloud Computing, 2011.
- [20] R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. IEEE Computer, 29:38–47, Feb 1996.
- [21] J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. A secure data deduplication scheme for cloud storage. In Technical Report, 2013.

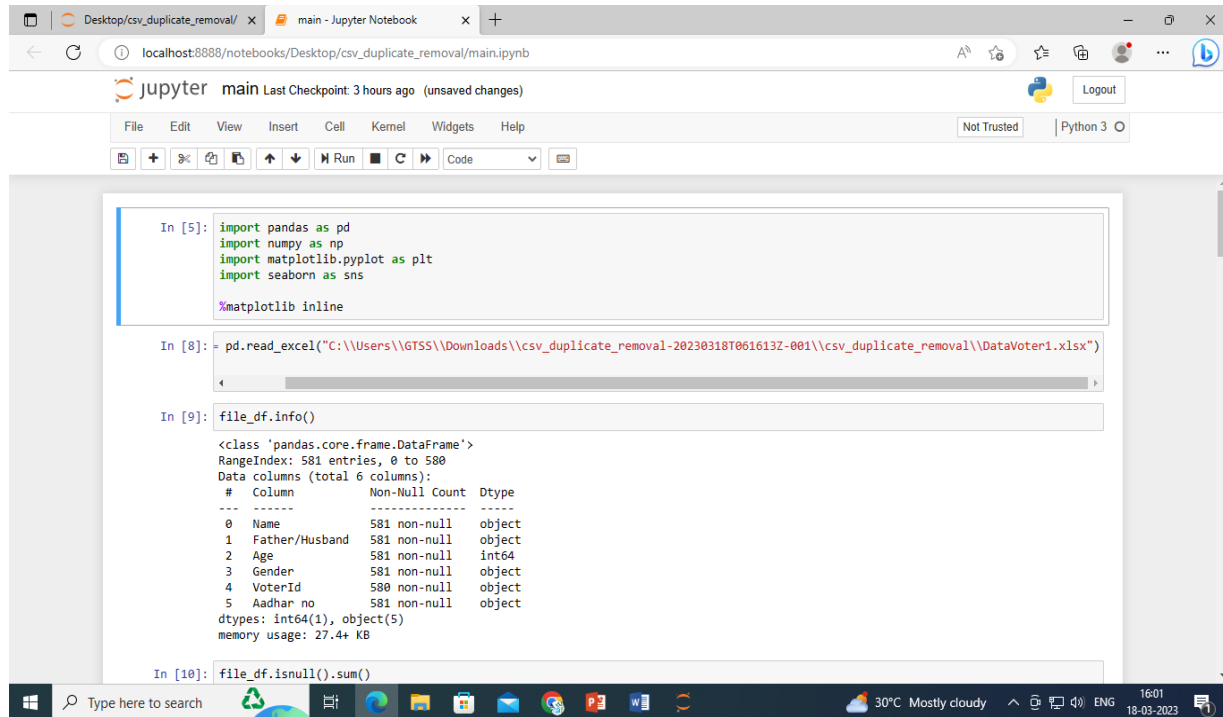
APPENDIX

A.SOURCE CODE

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline
file_df.info()
file_df.shape
print(file_df.describe())
count = 0
for i in file_df['Name'].unique():
    count = count + 1
    print(count, ' ', i)
sns.set_style('darkgrid')
file_df_first_record.to_excel("C:\\Users\\vinut\\OneDrive\\Desktop\\csv_duplicate_removal\\saving_files\\First_Record.xlsx", index=False)
file_df_last_record = file_df.drop_duplicates(subset=["Name", "Father/Husband ", "Age", "Gender", "VoterId", "Aadhar no"], keep="last")
sns.pairplot(file_df, hue = 'Gender')
plt.show()
sns.pairplot(file_df, hue = 'Gender')
plt.show()
file_df_last_record.to_excel("C:\\Users\\vinut\\OneDrive\\Desktop\\csv_duplicate_removal\\saving_files\\Last_Record.xlsx", index=False)
file_df_remove_all = file_df.drop_duplicates(subset=["Name", "Father/Husband ", "Age", "Gender", "VoterId", "Aadhar no"], keep=False)
sns.heatmap(file_df.corr(), annot = True, cmap='inferno')
plt.show()
file_df_remove_all.to_excel("C:\\Users\\vinut\\OneDrive\\Desktop\\csv_duplicate_removal\\saving_files\\All_Removed.xlsx", index=False)
duplicate_row_index = file_df.duplicated(subset=["Name", "Father/Husband ", "Age", "Gender", "VoterId", "Aadhar no"], keep=False)
file_df.drop_duplicates()
print(file_df)
all_duplicate_rows = file_df[duplicate_row_index]
all_duplicate_rows.to_excel("C:\\Users\\vinut\\OneDrive\\Desktop\\csv_duplicate_removal\\saving_files\\Duplicate_Rows.xlsx", index=True)
```


B SCREENSHOTS



The screenshot displays a Jupyter Notebook running in a web browser at localhost:8888. The notebook is titled 'main' and shows the following code cells:

```
In [5]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

%matplotlib inline

In [8]: pd.read_excel("C:\\Users\\GT55\\Downloads\\csv_duplicate_removal-20230318T061613Z-001\\csv_duplicate_removal\\DataVoter1.xlsx")

In [9]: file_df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 581 entries, 0 to 580
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0    Name            581 non-null    object
1    Father/Husband   581 non-null    object
2    Age             581 non-null    int64
3    Gender          581 non-null    object
4    VoterId         580 non-null    object
5    Aadhar no       581 non-null    object
dtypes: int64(1), object(5)
memory usage: 27.4+ KB

In [10]: file_df.isnull().sum()
```

The output of the `file_df.info()` cell shows the structure of the DataFrame, including the number of entries (581), the data types of the columns, and the memory usage (27.4+ KB).

Desktop/csv_duplicate_removal/ X main - Jupyter Notebook X +

localhost:8888/notebooks/Desktop/csv_duplicate_removal/main.ipynb

jupyter main Last Checkpoint: 3 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Father/Husband 0
Age 0
Gender 0
VoterId 1
Aadhar no 0
dtype: int64

In [11]: file_df.shape
Out[11]: (581, 6)

In [12]: file_df_first_record = file_df.drop_duplicates(subset=["Name", "Father/Husband ", "Age", "Gender", "VoterId", "Aadhar no"], keep="first")

In [13]: file_df.describe()

Out[13]:

	Age
count	581.000000
mean	31.227194
std	12.260956
min	18.000000
25%	23.000000
50%	25.000000
75%	38.000000
max	86.000000

Type here to search 30°C Mostly cloudy 16:01 18-03-2023

Desktop/csv_duplicate_removal/ X main - Jupyter Notebook x +

localhost:8888/notebooks/Desktop/csv_duplicate_removal/main.ipynb

jupyter main Last Checkpoint: 3 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 O

```

2. Guni Reddy Subbalakshamma
3. Padmavathi
4. Vela Diwakar Reddy
5. Parvathamma ChinnapuReddy
6. Sneha Chinna Ranga
7. Lakshmi Narsamma Bhumireddy
8. Harinath Reddy Bala
9. Guni Reddy Subbalakshamma
10. Narayanamma Chinnapureddy gari
11. Chinnapureddy Gari Suhasini
12. Diwakar Reddy vella
13. Rama kittamma A
14. Chinnapureddy Anki Reddy
15. Challa Devi
16. Sasikala Challa
17. Challa kasinathareddy
18. Ankireddy bhumireddy gari
19. SatynarayanareddyHanumanthareddygari

```

In [15]: `sns.set_style('darkgrid')`

In [16]: `file_df_first_record.to_excel("C:\\Users\\GTSS\\Downloads\\csv_duplicate_removal-20230318T061613Z-001\\csv_duplicate_removal\\sav`

In [17]: `file_df_last_record = file_df.drop_duplicates(subset=["Name", "Father/Husband ", "Age", "Gender", "VoterId", "Aadhar no"], keep="l`

In [18]: `sns.pairplot(file_df, hue = 'Gender')`
`plt.show()`

Desktop/csv_duplicate_removal/ X main - Jupyter Notebook x +

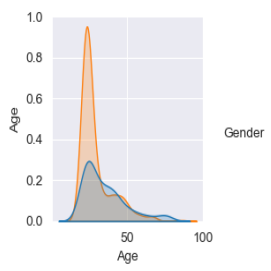
localhost:8888/notebooks/Desktop/csv_duplicate_removal/main.ipynb

jupyter main Last Checkpoint: 3 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 O

In [17]: `file_df_last_record = file_df.drop_duplicates(subset=["Name", "Father/Husband ", "Age", "Gender", "VoterId", "Aadhar no"], keep="l`

In [18]: `sns.pairplot(file_df, hue = 'Gender')`
`plt.show()`



In [19]: `file_df_last_record.to_excel("C:\\Users\\GTSS\\Downloads\\csv_duplicate_removal-20230318T061613Z-001\\csv_duplicate_removal\\sav`

In [20]: `file_df_remove_all = file_df.drop_duplicates(subset=["Name", "Father/Husband ", "Age", "Gender", "VoterId", "Aadhar no"], keep=False`

In [21]: `sns.heatmap(file_df.corr(), annot = True, cmap='inferno')`
`plt.show()`

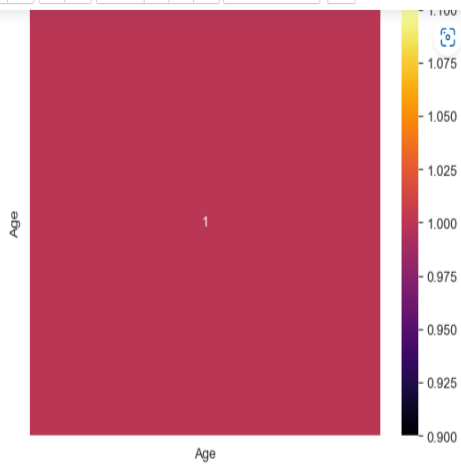
Desktop/csv_duplicate_removal/ X main - Jupyter Notebook X +

localhost:8888/notebooks/Desktop/csv_duplicate_removal/main.ipynb

Jupyter main Last Checkpoint: 3 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code



Age

Age

```
In [22]: file_df_remove_all.to_excel("C:\\Users\\GTSS\\Downloads\\csv_duplicate_removal-20230318T061613Z-001\\csv_duplicate_removal\\savei  
In [23]: duplicate_row_index = file_df.duplicated(subset=["Name", "Father/Husband ", "Age", "Gender", "VoterId", "Aadhar no"], keep=False)  
In [24]: file_df.drop_duplicates()
```

Type here to search 30°C Mostly cloudy 16:01 18-03-2023

Desktop/csv_duplicate_removal/ x main - Jupyter Notebook x +

localhost:8888/notebooks/Desktop/csv_duplicate_removal/main.ipynb

jupyter main Last Checkpoint: 3 hours ago (unsaved changes) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3

Run Code

0	Prasanna Lakshmi Chnnapu Reddy	Siva Reddy.C	25	F	RGS0616	7395-9878-1222
1	Gundreddy Mlikarjuna	Byreddy	66	M	BGL2946838	9444-0032-8237
2	Padmavathi	Chinnappa	41	F	BGL2372399	8447-7449-4509
3	Vela Diwakar Reddy	Ramanjaneya Reddy	30	M	BGL2946762	9952-0938-6723
4	Parvathamma ChinnapuReddy	Venkata Siva Reddy	54	F	BGL2373546	9396-8425-9855
...
495	KANCHARLA MADHAV CHOWDARY	KANCHARLA NAGESH BABU	24	M	RGS1381495	9734-5077-9000
496	KANDULA HAREESH BABU	KANDULA VENKATA RAO	23	M	BGL2947006	9908-7183-5121
497	KANISSETTY PAVAN KALYAN	KANISSETTY VENKATESHWARLU	23	M	AP231580501117	9785-0791-7735
498	KARANAM SAI RAKESH	KARANAM MURALI KRISHNA	22	M	BGL2945128	9778-1102-2277
499	KARNATI RAGHAVENDRA REDDY	KARNATI ASHOK REDDY	22	M	RGS0169929	9552-7812-2800

500 rows x 6 columns

```
In [25]: print(file_df)
```

	Name	Father/Husband	Age	Gender	
0	Prasanna Lakshmi Chnnapu Reddy	Siva Reddy.C	25	F	
1	Gundreddy Mlikarjuna	Byreddy	66	M	
2	Padmavathi	Chinnappa	41	F	
3	Vela Diwakar Reddy	Ramanjaneya Reddy	30	M	
4	Parvathamma ChinnapuReddy	Venkata Siva Reddy	54	F	
...
576	Krishna Reddy	Pulla Reddy	39	M	
577	Varalakshmi	Srinivasa Rao	44	F	
578	Srimanarayanna	Rosaya	36	M	
579	Kishore Kumar	Sangith Rao	18	M	
580	Swathi Komati	Ramayya	35	F	

	VoterId	Aadhar no
0	RGS0616	7395-9878-1222
1	BGL2946838	9444-0032-8237
2	BGL2372399	8447-7449-4509
3	BGL2946762	9952-0938-6723
4	BGL2373546	9396-8425-9855
...
576	BGL2946762	9542-4837-1977
577	BGL2373546	8951-9967-1720
578	RGS0836329	7981-5502-9303
579	AP231580504144	8978-7039-2967
580	RGS102395	9705-6700-2001

[581 rows x 6 columns]

```
In [26]: all_duplicate_rows = file_df[duplicate_row_index]
```

```
In [27]: all_duplicate_rows.to_excel("C:\\Users\\GTSS\\Downloads\\csv_duplicate_removal-20230318T061613Z-001\\csv_duplicate_removal\\savi
```

30°C Mostly cloudy 16:01 18-03-2023

C. RESEARCH PAPER

ANALYSIS ON VOTING DATA BY USING DEDUPLICATION TECHNIQUES IN CLOUD

VINUTHNA D^[1], MANI CHANDHANA T^[2], ANTO PRAVEENA ^[3]

[1][2] UG Student, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

[3] Assistant Professor, Dept. of CSE, Sathyabama Institute of Science and Technology, Chennai, India

ABSTRACT:- Data deduplication stands the data compression methods for clearing in computing storage, duplicate copies of repeated data are frequently used to reduce the amount of extra space required to save bandwidth. Data deduplication is perhaps the most used method of gathering data to eliminate duplicate data. It is typically used in distributed storage to reduce unnecessary space and online stockpiling. Along with imitating the impersonation, a method for data covering before sending it out was familiarized to ensure the classification of the data. This project is the first step in overcoming the issue of replicating genuine data in order to ensure data security. Also, we proposed a few de duplication enhancements that facilitate approved duplicate checking in our cloud design. The proposed security assessment findings stated that the proposed system is secure to the extent that it is not predetermined by the proposed security model. The suggested work demonstrates how our supported duplication check plan achieves unimportant the above and differs from conventional exercises.

Key Words: Data deduplication, Cloud computing, Data management, Cloud storage, Block level, file level, Client side, Server side

I INTRODUCTION

While distributed computing products hide locations and specifics, for Internet users they also seem to have no "virtualization" restrictions. Currently, cloud specialized companies provide the most cost-effective storage options with a wide range of low-cost equipment. How much data is present

in the system when distributed computing is distinguished access to store data is granted thanks to expanded distributed storage that is provided to specific customers. The is one of the administrations for distributed storage. Cloud computing is a climbing plan that really has a drawn principal thought from each one exchange and academia. Users gets benefits over the web using cloud computing. Client will use net associations of unusual gathering as opposed to purchasing or putting in them. All clients are furthermore seen as in duplicate truly investigate other than the genuine data. We moreover show a few new deduplication movements supporting endorsement duplicate present a protection a cream cloud building plan. Security analysis demonstrates that our technique complies with the definitions presented in the security model that has been proposed. The Data deduplication method is used. Data deduplication removes redundant data by maintaining the original copy and directing additional data to that copy rather than maintaining multiple copies of the same data. Both the record level and the block level are susceptible to deduplication. At the report level, it eliminates duplicate copies of comparable records, and at the block level, it also eliminates duplicate data chunks found in non-identical archives. Data deduplication is a specific type of data compression method for getting rid of redundant copies of repeated data In order to save cost and efficiently management, the data will be moved to the storage server provider (SSP) in the public

cloud with specified privileges and the de duplication technique will be applied to store only one copy of the same file. Because of privacy consideration, some files will be encrypted and allowed the duplicate check by employees with specified privileges to realize the access control. Traditional de duplication systems based on convergent encryption, although providing confidentiality to some extent; do not support the duplicate check with differential privileges. In other words, no differential privileges have been considered in the de duplication based on convergent encryption technique. It seems to be contradicted if we want to realize both de duplication and differential authorization duplicate check at the same time.

AIM AND SCOPE OF THE PROJECT

SCOPE OF THE PROJECT

Data de duplication techniques are widely employed to backup data and minimize network and storage overhead by detecting and eliminating redundancy among data.

OBJECTIVE

The principle objective is to have the option to reenact and appropriate the information base to numerous servers.

PROBLEM DEFINITION

The current framework just works at the suspension level or at the document level for reenactment purposes. It doesn't give the security needed to communicate something specific. Accordingly, data sent between outsiders or programmer clients can be gotten to.

LITERATURE SURVEY

Deduplication in cloud Computing for

Improvising Efficiency Towards Potential

J.Johin, R.K. Rohith, R.S.S.Yukesh Kumar, G.Paavai Anand. 2020

In this framework, assuming there are two records on the cloud server, the program needs to contrast the source documents and the records that have been made accessible so the client doesn't show up in the pictures and there is no impersonation record behind the scenes. be that as it may, the documents will be replicated by the application, so try not to duplicate the record to the server, in this way lessening the requirement for extra room on the server. The effect of the current framework is a security and execution examination. They can't resolve two data sets in a record. They don't take care of the issue of control frameworks.

Secure Data De-Duplication On 2PVC Algorithm In Web Services

N.Amutha¹,Dr.S.Sujatha. 2017

We request a postponed level of strategy anticipation and an assortment of ways of executing it to more readily comprehend the presentation of the assistance. Intelligibility can emerge when utilizing a situation dependent on the option to utilize a cloud-based capacity framework and to secure weak resources.

Data Deduplication in Cloud Storage with and Ownership Management.

Nandhana Kuttappan. 2020

Impersonation is incredible when numerous clients give a similar data. This is a period for security and protection. Possession plans permit those with a similar data to completely demonstrate that they own the data in the distributed storage. Numerous clients would now be able to conceal their data prior to moving it to the document to keep up with classification, however because of the novel idea of the control, this evades impersonation.

Database Management and Storage

Optimization Using Data De-Duplication and Fog Computing

Ashutosh Avadhani, Amruta Chaudhari, Aniket Powar, Ishwar Borse, XMrs. M.D. Sale. 2018

The framework is intended to be little and enormous. The motivation behind this undertaking is to make an Android application that can just store exceptional things and records downloaded from the server and can be found anyplace on the webpage.

Achieving Efficient Data Deduplication and Key Aggregation Encryption System in Cloud

Dr.MK Jayanthi, P.Sri vaibhavi, P.V.Naga Saithya, Y.Harshitha Reddy. 2020

The task has executed a help adaptation to duplicate the sent data. Gives functional data to clients. Suitable observing techniques have been set up to forestall undesirable access and admittance to illicit data. Approval of data is expected to ensure data security and forestall unapproved access. Encryption layer deduplication can store a great deal of memory and utilize memory.

Fog Computing and Its Role in the Internet of Things

F. Bonomi, R. Milito, J. Zhu, & S.Addepalli2017

Subsequently, the cloud should uphold an assortment of capacity, from brief to low, from extremely durable to high. We likewise recollect that the higher the level, the more extensive and longer it is. Brand names and worldwide cloud administrations are utilized as month to month and yearly information bases as a reason for business investigation. Regularly, HMI climate reports and groups show execution keys.

Formulating a Security Layer of Cloud Data Storage Framework Based on Multi Agent System Architecture

R.Atan, A.M.Talib, &M. A. A. Murad

2018

In this article, we investigate the data security issues in the data set to more readily comprehend the data that clients can use in the data set; We needed security administrations and the development of the MAS to make it more straightforward to store data in the cloud. The security framework comprises of two principle parts: a specialist instrument and a cloud information base. The development of the MAS comprises of five sorts of exercises: UIA, UA, DERA, DRA, and DDPA. To improve security, the construction of our MAS gives eleven security capacities dependent on four fundamental security approaches: cloud-based data remedy, trustworthiness, privacy and openness.

Efficient data collection in sensor-cloud system with multiple mobile sinks

Y. Li, T.Wang, G.Wang, J. Liang, & H.Chen 2017

The outcomes show that our calculation can communicate information from WSN to the Cloud in a brief time frame and lessen power utilization. Distributed computing further develops information handling and wireless sensor networks (WSNs). stockpiling. Notwithstanding, because of WSN's restricted abilities, the manner in which information is moved to the Cloud in the present moment turns into a framework cloud proviso.

EXISTING SYSTEM

Private cloud replicating frameworks go about as intermediaries that permit proprietors/clients to viably duplicate copyrights and different freedoms. Such improvements are significant and have drawn in light of a legitimate concern for scientists. Information base proprietors give the data set just open mists, while media exercises are utilized with private clouds.

Replicating information is a special method of gathering information to erase two duplicates from a data set. This strategy is utilized to further develop data set use and

can be utilized as an information move to lessen the quantity of bytes to send. Copy duplicates erase secured data by saving a solitary duplicate as opposed to putting away numerous duplicates of a similar substance, and send more data to that duplicate.

The impersonation might rely upon the document level or the suspended level. At the level of the copy document, it erases a solitary record duplicate. Deduplication can likewise happen at the reserve level, which wipes out deduplication of data found in contradictory documents. A few duplicates from various clients lead to various records and make it difficult to duplicate.

Disadvantages

- Enc Traditional recognizable proof affirms the secrecy of the data and can't duplicate the data.
- Duplicate One duplicate of the data from various clients makes ciphertext and makes it difficult to duplicate.

PROPOSED SYSTEM

In this proposed work, the system enhanced with security. Specifically, it present an advanced scheme to support stronger security by encrypting the file with differential privilege keys. In this way, the users without corresponding privileges cannot perform the duplicate check. Furthermore, such unauthorized users cannot decrypt the cipher text even collude with the S-CSP. Security analysis demonstrates that our system is secure in terms of the definitions specified in the proposed security model.

Convergent encryption has been proposed to enforce data confidentiality while making de duplication feasible. It encrypts/decrypts a data copy with a convergent key, which is obtained by computing the cryptographic hash value of the content of the data copy. After key generation and data encryption, users retain

the keys and send the cipher text to the cloud. Since the encryption operation is deterministic and is derived from the data content, identical data copies will generate the same convergent key and hence the same cipher text. To prevent unauthorized access, a secure proof of ownership protocol is also needed to provide the proof that the user indeed owns the same file when a duplicate is found.

Advantages

- The user is only allowed to perform the duplicate check for files marked with the corresponding privileges.
- We present an advanced scheme to support stronger security by encrypting the file with differential privilege keys.
- Reduce the storage size of the tags for integrity check. To enhance the security of de duplication and protect the data confidentiality.

METHODS AND ALGORTIHMS USED

HARDWARE REQUIREMENT

The hardware requirements may serve as the basis for a contract for the implementation of the system and should therefore be a complete and consistent specification of the whole system. They are used by software engineers as the starting point for the system design. It shows what the system does and it also shows how it should be implemented. It specifies the speed of the system that should be used in this project. The hardware requirement for this project is mentioned below.

System	: Pentium IV 2.4 GHz
Hard Disk	: 40 GB
Floppy Drive	: 44 Mb
Monitor	: 15 VGA Colour
Ram	: 512 Mb

SOFTWARE REQUIREMENT

The software requirements document is the specification of the system. It should include both a definition and a specification of requirements. It is a set of what the system should do rather than how it should do it. The software requirements provide a basis for creating the software requirements specification. It is useful in estimating cost, planning team activities, performing tasks and tracking the team and tracking the team's progress throughout the development activity. The software requirement specifies the application software that is being used in the project. The software needed to develop this project is mentioned below.

Operating system : Windows XP/7

IDE : Eclipse

Coding Language: Java

SOFTWARE ENVIRONMENT

INTRODUCTION

Java is one of the world's most important and widely used computer languages, and it has held this distinction for many years. Unlike some other computer languages whose influence has waned with passage of time, while Java's has grown.

APPLICATION OF JAVA

Java is widely used in every corner of world and of human life. Java is not only used in softwares but is also widely used in designing hardware controlling software components. There are more than 930 million JRE downloads each year and 3 billion mobile phones run java.

Following are some other usage of Java:

1. Developing Desktop Applications
2. Web Applications like
Linkedin.com, Snapdeal.com etc
3. Mobile Operating System like
Android
4. Embedded Systems
5. Robotics and games etc.

FEATURES OF JAVA

The prime reason behind creation of Java was to bring portability and security feature into a computer language. Beside these two major features, there were many other features that played an important role in moulding out the final form of this outstanding language. Those features are;

1) Simple

Java is easy to learn and its syntax is quite simple, clean and easy to understand. The confusing and ambiguous concepts of C++ are either left out in Java or they have been re-implemented in a cleaner way.

Eg: Pointers and Operator Overloading are not there in java but were an important part of C++.

2) Object Oriented

In java everything is Object which has some data and behaviour. Java can be easily extended as it is based on Object Model.

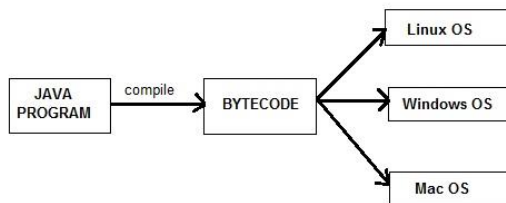
3) Robust

Java makes an effort to eliminate error prone codes by emphasizing mainly on compile time error checking and runtime checking. But the main areas which Java improved were Memory Management and mishandled Exceptions by introducing automatic Garbage Collector and Exception Handling.

4) Platform Independent

Unlike other programming languages such as C, C++ etc. which are compiled into platform specific machines. Java is guaranteed to be write-once, run-anywhere language.

On compilation Java program is compiled into byte code. This byte code is platform independent and can be run on any machine, plus this byte code format also provide security. Any machine with Java Runtime Environment can run Java Programs.



Collection framework was not part of original Java release. Collections was added to J2SE 1.2. Prior to Java 2, Java provided adhoc classes such as Dictionary, Vector, Stack and Properties to store and manipulate groups of objects. Collection framework provides many important classes and interfaces to collect and organize group of alike objects.

5) Secure

When it comes to security, Java is always the first choice. With java secure features it enable us to develop virus free, temper free system. Java program always runs in Java runtime environment with almost null interaction with system OS, hence it is more secure.

6) Multi-Threading

Java multithreading feature makes it possible to write program that can do many tasks simultaneously. Benefit of multithreading is that it utilizes same memory and other resources to execute multiple threads at the same time, like While typing, grammatical errors are checked along.

7) Architectural Neutral

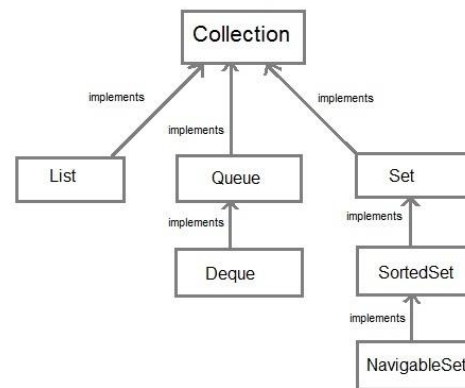
Compiler generates byte codes, which have nothing to do with a particular computer architecture, hence a Java program is easy to interpret on any machine.

8) Portable

Java Byte code can be carried to any platform. No implementation dependent features. Everything related to storage is predefined, example: size of primitive data types

10) High Performance

Java is an interpreted language, so it will never be as fast as a compiled language like C or C++. But, Java enables high performance with the use of just-in-time compiler.



Test Objectives

1. All field entries must work properly.
2. Pages must be activated from the identified link.
3. The entry screen, messages and responses must not be delayed.
4. Features to be tested
5. Verify that the entries are of the correct format
6. No duplicate entries should be allowed
7. All links should take the user to the correct page.

COLLECTION FRAMEWORK

Integration Testing

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications.

Acceptance Testing

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

ALPHA TESTING

In software development, alpha test will be a test among the teams to confirm that your product works. Originally, the term alpha test meant the first phase of testing in a software development process. The first phase includes unit testing, component testing, and system testing. It also enables us to test the product on the lowest common denominator machines to make sure download times are acceptable and preloads work.

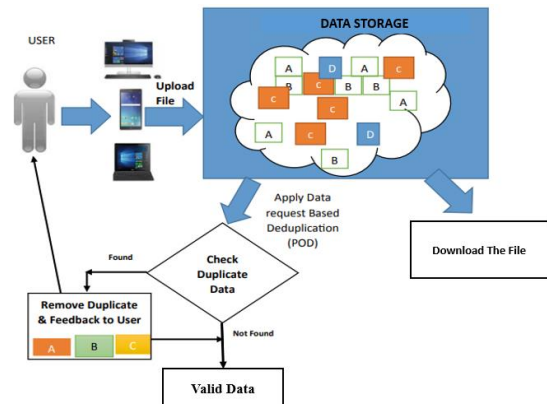
BETA TESTING

In software development, a beta test is the second phase of software testing in which a sampling of the intended audience tries the product out. Beta testing can be considered “pre-release testing.” Beta test versions of software are now distributed to curriculum specialists and teachers to give the program a “real-world” test.

4.3 SYSTEM ARCHITECTURE

The system architecture establishes the basic structure of the system, defining the essential core design features and elements that provide the framework for the system. The systems architecture provides the architects view of the users’ vision for what the system needs to be and do, and the paths along which it must be able to evolve and strives to maintain the integrity

of that vision as it evolves during detailed design and implementation



**Fig 1 . Architecture of the system
SYSTEM IMPLEMENTATION**

DATA FLOW DIAGRAM

The Data Flow diagram is a graphic tool used for expressing system requirements in a graphical form. The DFD also known as the “bubble chart” has the purpose of clarifying system requirements and identifying major transformations that to become program in system design. Thus DFD can be stated as the starting point of the design phase that functionally decomposes the requirements specifications down to the lowest level of detail. The DFD consist of series of bubbles joined by lines. The bubbles represent data transformations and the lines represent data flows in the system. A DFD describes what that data flow in rather than how they are processed. So it does not depend on hardware, software, data structure or file organization

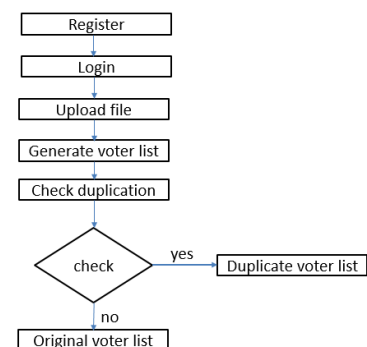


Fig 2. data flow diagram

CLASS DIAGRAM

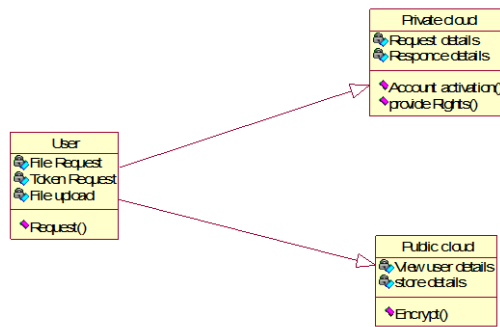


Fig 3. Class diagram

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. The message sending procedure involves user registering with the system and logs into the system. The authenticated secret message involves the message extraction and authentication of the message.

USE CASE DIAGRAM

A use case is a set of scenarios that describing an interaction between a user and a system. A use case diagram displays the relationship among actors and use cases. The two main components a user or another system that will interact with the system modeled. A use case is an external view of the system that represents some action the user might perform in order to complete a task.

Fig 4. Use case diagram

ACTIVITY DIAGRAM

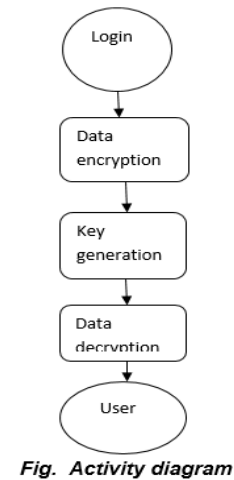


Fig. Activity diagram

SEQUENCE DIAGRAM

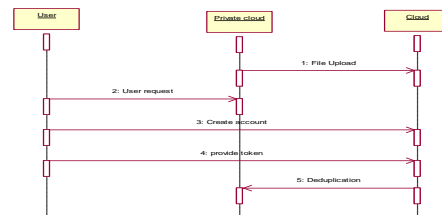


Fig5. Sequence diagram

The sequence diagram is used to show the flow of the system with the time frame of each activity. The sender logs into the system and generates the file. Then it gets split using the chunk algorithm and encrypts the secret message using blowfish algorithm. The key is generated. The sender gets the private key when he registers with the system and applies inverse blowfish algorithm and split the image into different blocks. Using the private key the message is decrypted and original message is received by the user.

CONCLUSION

In this paper, the notion of authorized data de duplication was proposed to protect the data security by including differential privileges of users in the duplicate check. We also presented several new de duplication constructions supporting authorized duplicate check in hybrid cloud architecture, in which the duplicate-check tokens of files are generated by the private cloud server with private keys. Security analysis demonstrates that our schemes are secure in terms of insider and outsider attacks specified in the proposed security model. As a proof of concept, we implemented a prototype of our proposed authorized duplicate check scheme and conduct test bed experiments on our prototype. We showed that our authorized duplicate check scheme incurs minimal overhead compared to convergent encryption and network transfer.

MODULES

- **User Module**
- **Data entry module**
- **Secure DE duplicate System**
- **Download file**

Use module

- In this module, Users are having authentication and security to access the detail which is presented in the ontology system. Before accessing or searching the details user should have the account in that otherwise they should register first. At the very least, you need to provide an email address, username, password, display name, and whatever profile fields you have set to required. The display name is what will be used when the

system needs to display the proper name of the user.

Data entry module

- The user can start up the server after cloud environment is opened. Then the user can enter details to the cloud.

Secure DE duplicate System

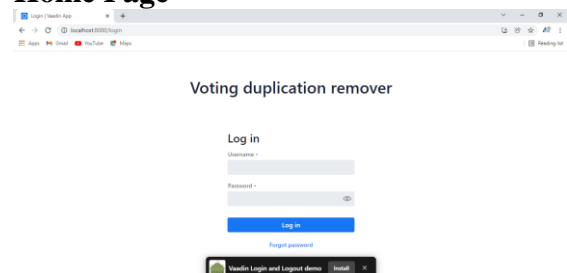
To support authorized de duplication the tag of a file F will be determined by the file F and the privilege. To show the difference with traditional notation of tag, we call it file token instead. To support authorized access a secret key KP will be bounded with a privilege p to generate a file Token. De duplication exploits identical content, while encryption attempts to make all content appear random; the same content encrypted with two different keys results in very different cipher text. Thus, combining the space efficiency of de duplication with the secrecy aspects of encryption is problematic.

Download file

After the cloud storage, the user can download the file based on key or token. Once the key request was received, the sender can send the key or he can decline it. With this key and request id which was generated at the time of sending key request the receiver can decrypt the message.

IMPLEMENTATION

Home Page



Admin Login Page

Voting duplication remover

Log in

Username
admin

Password
admin

Log in

[Forgot password](#)

[Admin Login and Logout demo](#)

User Login Page

Voting duplication remover

Log in

Username
user

Password
password

Log in

[Forgot password](#)

[Admin Login and Logout demo](#)

Add Aadhar Details

Voting duplication remover

Filter by name: [Add aadhar](#)

Number	Name	Aadhar number
487041001129	Demo name	
051491812391	Thalapathi Selva	

First Name
sandeep

Last Name
reddy

Father Name
Ranganthreddy

Mother Name
LakshmiDevi

Door number
2/258

Street name
sak nagar

Landmark
Anatapur

District
Anatapur

State
Andhra Pradesh

Pincode
515001

[Save](#) [Delete](#) [Cancel](#)

Add Voters Details

Voting duplication remover

Filter by name: [Add voter](#)

ID	Name	Father Name	Voter ID
IT5882354	Demo name	Demo father	RCV0002091
UV0007549	Demo name 1	Demo father	
CEH232057	Demo name 2	Demo father	
THH1981327	Thalapathi Selva	Selva kumar	
HC7897434	Thalapathi Selva 1	Selva kumar	
RCV0002091	Thalapathi Selva 2	Selva kumar	

Name
Thalapathi Selva 2

Father Name
Selva kumar

Gender
Male

Date of birth
31/12/2001

Door number
1111

Street name
Demo street

Landmark
Gowdly

Voting duplication remover

City
Chennai

District
Chennai

State
Tamil Nadu

Demo pin
600001

Assembly Constituency Number
123

Assembly Constituency Name
Demo Constituency Name

Par 1 Number
123

Par 1 Name
Demo part 1

[Save](#) [Delete](#) [Cancel](#)

Voting duplication remover

Filter by name: [Add voter](#)

ID	Name	Father Name
IT5882354	Demo name	Demo father
UV0007549	Demo name 1	Demo father
CEH232057	Demo name 2	Demo father
THH1981327	Thalapathi Selva	Selva kumar
HC7897434	Thalapathi Selva 1	Selva kumar
RCV0002091	Thalapathi Selva 2	Selva kumar

REFERENCES

1. P. Anderson and L. Zhang. Fast and secure laptop backups with encrypted de-duplication. In Proc. of USENIX LISA, 2010
2. M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.
3. M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked encryption and secure deduplication. In EUROCRYPT, pages 296–312, 2013.
4. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.
5. M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002.
6. M. Bellare, S. Keelveedhi, and T. Ristenpart. Dupless: Serveraided encryption for deduplicated storage. In USENIX Security Symposium, 2013.
7. M. Bellare, S. Keelveedhi, and T. Ristenpart. Message-locked
8. M. Bellare, C. Namprempre, and G. Neven. Security proofs for identity-based identification and signature schemes. J. Cryptology, 22(1):1–61, 2009.
9. M. Bellare and A. Palacio. Gq and schnorr identification schemes: Proofs of security against impersonation under active and concurrent attacks. In CRYPTO, pages 162–177, 2002.
10. S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.
11. S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider. Twin clouds: An architecture for secure cloud computing. In Workshop on Cryptography and Security in Clouds (WCSC 2011), 2011.
12. J. R. Douceur, A. Adya, W. J. Bolosky, D. Simon, and M. Theimer. Reclaiming space from duplicate files in a serverless distributed file system. In ICDCS, pages 617– 624, 2002.
13. D. Ferraiolo and R. Kuhn. Role-based access controls. In 15th NIST-NCSC National Computer Security Conf., 1992.

14. S. Halevi, D. Harnik, B. Pinkas, and A. ShulmanPeleg. Proofs of ownership in remote storage systems. In Y. Chen, G. Danezis, and V. Shmatikov, editors, ACM Conference on Computer and Communications Security, pages 491–500. ACM, 2011.
15. J. Li, X. Chen, M. Li, J. Li, P. Lee, and W. Lou. Secure deduplication with efficient and reliable convergent key management. In IEEE Transactions on Parallel and Distributed Systems, 2013.
16. C. Ng and P. Lee. Revdedup: A reverse deduplication storage system optimized for reads to latest backups. In Proc. of APSYS, Apr 2013.
17. W. K. Ng, Y. Wen, and H. Zhu. Private data deduplication protocols in cloud storage. In S. Ossowski and P. Lecca, editors, Proceedings of the 27th Annual ACM Symposium on Applied Computing, pages 441–446. ACM, 2012.
18. R. D. Pietro and A. Sorniotti. Boosting efficiency and security in proof of ownership for deduplication. In H. Y. Youm and Y. Won, editors, ACM Symposium on Information, Computer and Communications Security, pages 81–82. ACM, 2012.
19. S. Quinlan and S. Dorward. Venti: a new approach to archival storage. In Proc. USENIX FAST, Jan 2002. [18] A. Rahumed, H. C. H. Chen, Y. Tang, P. P. C. Lee, and J. C. S. Lui. A secure cloud backup system with assured deletion and version control. In 3rd International Workshop on Security in Cloud Computing, 2011.
20. R. S. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. IEEE Computer, 29:38–47, Feb 1996.
21. J. Stanek, A. Sorniotti, E. Androulaki, and L. Kencl. A secure data deduplication scheme for cloud storage. In Technical Report, 2013.