Design and Implementation of Remote Data Possession Checking Protocol using Homomorphic Hash Fuction for Cloud Data

Ajay S Kulkarni[#1], Guntumadugu Girish Raj[#2]
[#1]BE, Computer Science Engineering, K.S School of Engineering, Bengaluru, Karnataka, India.
[#2]BE, Mechanical Engineering, SJB Institute of Technology, Bengaluru, Karnataka, India.

Abstract

Cloud computing emerges as a novel computing paradigm subsequent to grid computing. As an important application in cloud computing, cloud storage offers user scalable, flexible, and high-quality data storage and computation services. A growing number of data owners choose to outsource data files to the cloud. Cloud service provider tries to provide a promising service for data storage, which saves the users costs of investment and resource. Nonetheless, cloud storage also brings various security issues for the outsourced data. Because cloud storage servers are not fully trustworthy, data owners need dependable means to check the possession for their files outsourced to remote cloud servers. But many existing schemes have vulnerabilities in efficiency or data dynamics. In this system, we provide a new efficient RDPC protocol based on homomorphic hash function [1]. Our scheme employs a homomorphic hash function to verify the integrity for the files stored on remote server, and reduces the storage costs and computation costs of the data owner. In this system, we study the issue for integrity checking of data files outsourced to remote server and propose an efficient secure RDPC protocol with data dynamics. Using our new data structure, the data owner can perform insert, modify or delete operation on file blocks with high efficiency and check the possession of the data.

Keywords - *Cloud Storage, Data Possession Check, Homomorphic Hash Function, Dynamic Operation, AES Algorithm.*

I. INTRODUCTION

Cloud computing is a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction. Cloud computing is a method for delivering information technology (IT) services in which resources are retrieved from the Internet through web-based tools and applications, as opposed to a direct connection to a server. Rather than keeping files on a proprietary hard drive or local storage device, cloud-based storage makes it possible to save them to a remote database. As long as an electronic device has access to the web, it has access to the data and the software programs to run it. Cloud computing emerges as a novel computing paradigm subsequent to grid computing. By managing a great number of distributed computing resources in Internet, it possesses huge virtualized computing ability and storage space. Thus, cloud computing is widely accepted and used in many real applications. It's called cloud computing because the information being accessed is found in "the cloud" and does not require a user to be in a specific place to gain access to it. This type of system allows employees to work remotely. Cloud computing enables companies to consume a compute resource, such as a virtual machine (VM), storage or an application, as a utility just like electricity rather than having to build and maintain computing infrastructures in house.

A. Background

Remote Data Possession Checking (RDPC) is an effective technique to ensure the integrity for data files stored on CSS. RDPC supplies a method for data owner to efficiently verify whether cloud service provider faithfully stores the original files without retrieving it.

In RDPC, the data owner is able to challenge the CSS on the integrity for the target file. The CSS can generate proofs to prove that it keeps the complete and uncorrupted data.

The fundamental requirement is that the data owner can perform the verification of file integrity without accessing the complete original file. Moreover, the protocol must resist the malicious server which attempts to verify the data integrity without accessing the complete and uncorrupted data. Another desired requirement is that dynamic data operations should be supported by the protocol. In general, the data owner may append, insert, delete or modify the file blocks as needed. Besides, the computing complexity and communication overhead of the protocol should be taken into account for real applications.

B. *Existing Systems and Drawbacks*

The first RDPC was proposed by Deswarte et al based on RSA hash function. The major drawback was that, it needed to access the entire file blocks for each challenge. [3]

In 2007, the Provable Data Possession (PDP) model was presented by Ateniese et al which used the probabilistic proof technique for remote data integrity checking without accessing the whole file. The major drawback of this scheme was that it didn't support dynamic operations. [4]

Wang et al. used Merkle Hash Tree (MHT) to propose another dynamic method for remote data checking, in which each block was hashed to be a leaf node of MHT. The disadvantage is heavy computational cost. [2]

Chen applied algebraic signature to introduce a new remote data checking protocol. The drawback of this technic was that it was insecure to Replay attack. [5]

C. *Motivation*

It is essential for data owners to verify the integrity for the data stored on CSS before using it. For example, a big international trading company stores all the imports and exports record files on CSS. According to these files, the company can get the key information such as the logistics quantity, the trade volume etc. If any record file is discarded or tampered, the company will suffer from a big loss which may cause bad influence on its business and development.

To avoid this kind of circumstances, it is mandatory to check the integrity for outsourced data files. Furthermore, since these files may refer to business secret, any information exposure is unacceptable.

If the company competitor can execute the file integrity checking, by frequently checking the files they may obtain some useful information such as when the file changes, the growth rate of the file etc., by which they can guess the development of the company. Thus, to avoid this situation, we consider the private verification type in our scheme, that is, the data owner is the unique verifier checking the integrity of data.

D. *Proposed System*

We present a novel efficient RDPC scheme by using Homomorphic hash function to construct RDPC schemes. To overcome drawbacks of previous schemes, we refer to the idea and introduce a private key for each tag generation in our RDPC scheme.

II. SYSTEM REQUIREMENTS

The web portal of this project is implemented using JAVA J2EE.

A. *Hardware Requirements*

- CPU: 2.4 GHZ or above.
- RAM: 8GB RAM.
- Hard disk: 40GB or above.

B. *Software Requirements*

- Backend: Core Java, Advanced Java, J2EE (JDBC, Servlet, JSPs).
- Front end: HTML, CSS, JavaScript, Bootstrap, Ajax, Jquery.
- Database: MySQL 5.5.
- Servers: Apache Tomcat v-9.0.
- Development environment: Eclipse.
- Cloud service: DigitalOcean.

III. DESIGN

Design is the technique which is used to do the system analysis, it would be necessary to identify the data that is required to be processed to produce the outputs. Design features can ensure reliability of the system and generate correct reports from the accurate data. It is also possible to determine whether the user can interact efficiently with the system.

A. *Architecture Diagram*

The An architectural model (in software) is a rich and rigorous diagram, created using available standards, in which the primary concern is to illustrate a specific set of trade off inherent in the structure and design of a system or ecosystem.

A System Architecture is the conceptual model that defines the structure, behaviour, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviours of the system.

The architecture diagram has 4 layers namely,

- Client layer
- Application layer
- Business logic layer
- Database layer

As shown in figure Fig 1, the client layer consists of client devices. Client layer generates request. Browser is an example of client layer component. The client layer can also be called as front end. Application layer, business logic layer and database layer are collectively called the back end. The application layer has servers like Apache tomcat or wildfly. The business logic layer is where the code is written. Database layer consists of platforms to store data like MySQL, Oracle, and Hadoop etc.

Question may arise if business logic layer can be incorporated in web layer. Yes, but this would not be an efficient way because of scalability and security. There are many clients but only one server. So if web layer directly has business logic layer in it and it interacts with database, then it can handle only one client at a time. Hence we use separate business logic layer which interacts with database layer.

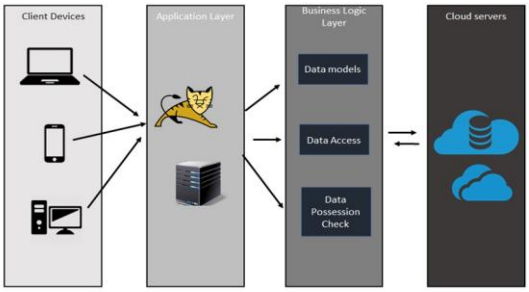


Fig 1: Architectural Diagram of the System

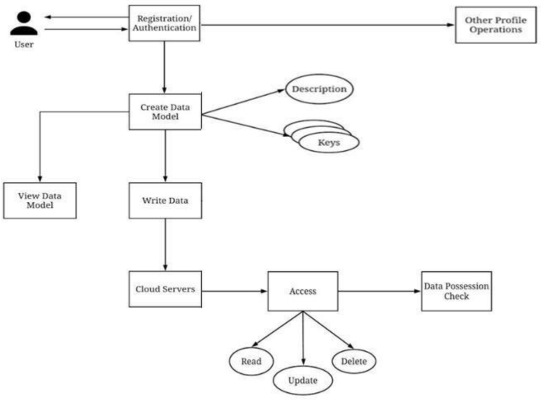


Fig 2: Outlook of Overall System

B. *Outlook of Overall System*

The figure Fig 2, represents the overall outlook of the system. The database is accessed for authentication/registration of the user based on the credentials provided, enabling the user to login to the system. The user can perform profile operations such as delete account, change password etc., once he logs into the system. The user can perform other multiple tasks once logged in.

User can create data model by giving description about model and by specifying different keys (fields) of that data model. For example : to store student details, the user can give the description as ‘student information’ and can give the keys as ‘name’, ’USN’, ’aggregate’ etc. Then, the user can view all the data models that he has created. Viewing data model module contains details like the data model ID, description, keys, time when the model was created. It also has a delete option to delete any model.

Once data model is created, the user can write data into the models i.e., giving values to the keys that user had specified.

While accessing the data, the user can update data, delete it, and also the user can make a data possession check to see if modification of data has occurred. As cloud servers are not fully trustworthy, they can modify the data which is written onto the cloud. So, data possession checking is required.

IV. ALGORITHM USED

A. *Homomorphic Hash Function*

1) *Computing Hash Value:*

- Select 3 parameters: p, q, and g.
- Here p and q are two random prime numbers.
- Let ‘S’ denote a message

  H(s) → X denotes the hash of ‘S’.
- ‘S’ is divided into m sectors

  $S = (s_1, s_2, s_3, \dots s_m)$
- The hash value is calculated as

$$H_k(S) = \prod_{i=1}^{m} g_i^{s_i} \, mod \, p$$

2) *Homomorphic Property:*

- For any two messages S_i and S_j

  We have,

  $S_i = (s_{1i}, s_{2i}, s_{3i}, \dots s_{mi})$

  $S_j = (s_{1j}, s_{2j}, s_{3j}, \dots s_{mj})$

  $S_i + S_j = (S_{1i}+S_{1j}, S_{2i}+S_{2j}, \dots S_{mi}+S_{mj})$

$$\mathrm{H}(s_i + s_i) = \prod_{t=1}^{m} g_t^{s_{ti}+s_{tj}} \, mod \, p$$

$$= \prod_{t=1}^{m} g_t^{s_{ti}} \, mod \, p . \prod_{t=1}^{m} g_t^{s_{tj}} \, mod \, p$$

$$=H(s_i).H(s_j)$$

B. *AES(Advanced Encryption Standards)*

The Advanced Encryption Standard, or AES, is a symmetric block cipher chosen by the U.S. government to protect classified information and is implemented in software and hardware throughout the world to encrypt sensitive data.

NIST specified the new advanced encryption standard algorithm must be a block cipher capable of handling 128 bit blocks, using keys sized at 128, 192, and 256 bits; other criteria for being chosen as the next advanced encryption standard algorithm included:

Security: Competing algorithms were to be judged on their ability to resist attack, as compared to other submitted ciphers, though security strength was to be considered the most important factor in the competition.

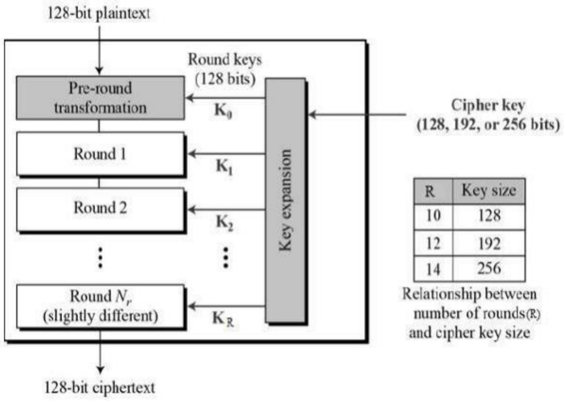
Cost: Intended to be released under a global, nonexclusive and royalty-free basis, the candidate algorithms were to be evaluated on computational and memory efficiency.

Implementation: Algorithm and implementation characteristics to be evaluated included the flexibility of the algorithm; suitability of the algorithm to be implemented in hardware or software and overall, relative simplicity of implementation.

The features of AES are as follows:

- Symmetric key symmetric block Cipher.
- 128-bit data, 128/192/256-bit keys.
- Stronger and faster than Triple-DES.
- Provide full specification and design details.
- Software implementable in C and Java.

AES comprises three block ciphers: AES-128, AES-192 and AES-256. Each cipher encrypts and decrypts data in blocks of 128 bits using cryptographic keys of 128, 192 and 256 bits, respectively.



R	Key size
10	128
12	192
14	256

Relationship between number of rounds(R) and cipher key size

Fig 3: Schematic Representation of AES Structure

The Fig 3. Depicts schematic representation of AES structure. Symmetric (also known as secret-key) ciphers use the same key for encrypting and decrypting, so the sender and the receiver must both know and use the same secret key.

There are 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys a round consists of several processing steps that include substitution, transposition and mixing of the input plaintext and transform it into the final output of cipher text.

The first step of the cipher is to put the data into an array; after which the cipher transformations are repeated over a number of encryption rounds. The number of rounds is determined by the key length, with 10 rounds for 128-bit keys, 12 rounds for 192-bit keys and 14 rounds for 256-bit keys.

The first transformation in the AES encryption cipher is substitution of data using a substitution table; the second transformation shifts data rows, the third mixes columns. The last transformation is a simple exclusive or (XOR) operation performed on each column using a different part of the encryption key longer keys need more rounds to complete.

Decryption Process: The process of decryption of an AES cipher text is similar to the encryption process in the reverse order. Each round consists of the four processes conducted in the reverse order.

- Add round key
- Mix columns
- Shift rows
- Byte substitution

Since sub-processes in each round are in reverse manner, unlike for a Feistel Cipher, the encryption and decryption algorithms needs to be separately implemented, although they are very closely related.

V. IMPLEMENTATION

This project is implemented considering the following aspects:

- Usability Aspect.
- Technical Aspect.

A. *Usability Aspect*

The usability aspect of implementation of the project is realized using two principles:

- The project is implemented as a Java application. There could be many ways of implementing the project. We have chosen JAVA to come up with the required reader.

  The reason being many: Firstly, Java provides a wonderful libraries which simplifies the implementation part of it.

  Secondly, JAVA is platform independent, meaning the project can run on literally any platform which has JVM installed within it.

  Thirdly, Oracle Corporation claims more than 70 billion devices run on JAVA which makes the end users used to it.

  Lastly, it can be readily portable to any devices like mobile phones, ipads, PDA, and any hand held devices that are capable of running JAVA.
- The user-friendly interface using Java's view architecture. The interface provided by this application is very user friendly and is developed using Java Swings.

B. *Technical Aspect*

The technical aspect of implementation of the project is realized as explained below:

1) Servers:

Apache Tomcat to develop the product: Apache Tomcat (or simply Tomcat, formerly also Jakarta Tomcat) is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Sun Microsystems, and provides a "pure Java" HTTP web server environment for Java code to run.

Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files.

JBOSS Application server to host the product: WildFly, formerly known as JavaBeans Open Source Software Application Server (JBoss AS, or simply JBoss) is an application server that implements the Java Platform, Enterprise Edition (Java EE).

JBoss is written in Java and as such is cross-platform: usable on any operating system that supports Java.

JBoss was developed by JBoss, now a division of Red Hat. Licensed under the terms of the GNU Lesser General Public License, JBoss is free and open source software.

The renaming to WildFly was done to reduce confusion. The renaming only affects the JBoss Application Server project. The JBoss Community or the Red Hat JBoss product line (with JBoss Enterprise Application Platform) all retain their names.

*2) **Database:***

MySQL officially, but also called "My Sequel" is (as of 2008) the world's most widely used open source relational database management system (RDBMS) that runs as a server providing multi-user access to a number of databases. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter. The SQL phrase stands for Structured Query Language [6].

The MySQL development project has made its source code available under the terms of the GNU General Public License, as well as under a variety of proprietary agreements. MySQL was owned and sponsored by a single for-profit firm, the Swedish company MySQL AB, now owned by Oracle Corporation.

MySQL is a popular choice of database for use in web applications, and is a central component of the widely used LAMP open source web application software stack (and other 'AMP' stacks). LAMP is an acronym for "Linux, Apache, MySQL, Perl/PHP/Python." Free-software-open source projects that require a full-featured database management system often use MySQL.

For commercial use, several paid editions are available, and offer additional functionality. Applications which use MySQL databases include: TYPO3, Joomla, WordPress, phpBB, MyBB, Drupal and other software. MySQL is also used in many high-profile, large-scale World Wide Web products, including Wikipedia, Google (though not for searches), Facebook, Twitter, Flickr, Nokia.com, and YouTube.

VI. LIST OF MODULES

A. Login Page Module

In the login page, if the user has an account, he can directly login by giving valid email id and password.

If the user does not have an account, he must first register by giving certain inputs like mail id, password, address, phone number, address and gender. Once the user has registered his account, then he can login to his account to perform further operations.

B. Creating Data Model Module

To create a data model, the user must select ‘create data model’ menu. In the create data model page, the user has to give a description of the data model that he is going to create.

Next, the user must specify the keys (fields) of the data model. The user can create multiple keys by clicking ‘Add’ option.

Once the user has given all required keys for data model, he has to click on ‘create data model’ option, and the data model will be created.

C. Viewing Module

The user can view all that data models that he has created by selecting ‘view data model’ menu. If no data models are created, then this page displays a message as ‘no data models found’.

If data models were created, then this page displays details like model id, description, keys, and created time.

User can delete data model created by selecting ‘delete’ option.

D. Write Module

To write data into data models, user has to select ‘write data’ menu. The user must then select the specific data model in which he has to write from the drop down list. The user must enter values for all the keys specified while creating the data model. After entering values for all keys, click on ‘write data’ option to save it. The encrypted data will be stored on the cloud.

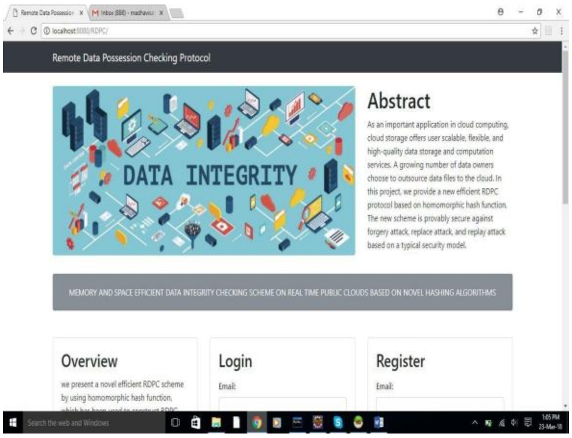
E. Access Data Module

Click on Access data module and select ‘View data’ option. The decrypted data will be displayed and the data model along with hashed values for key and values will be displayed. Also, proof for hash property will be displayed.

The values of keys field can be changed and click on ‘Update data’ option to update the data. Click on ‘Delete data’ option to delete particular data. Click on ‘Data possession check button’ to check for data integrity results. If the data is modified, on checking possession, status will be shown as ‘failed’ else it displays ‘success’.

VII. RESULTS AND DISCUSSIONS

It deals with the various results that are obtained after the successful execution of the application.


localhost:8080/RDPC/

Remote Data Possession Checking Protocol

Abstract

As an important application in cloud computing, cloud storage offers user scalable, flexible, and high-quality data storage and computation services. A growing number of data owners choose to outsource data files to the cloud. In this project, we provide a new efficient RDPC protocol based on homomorphic hash function. The new scheme is provably secure against forgery attack, replace attack, and replay attack based on a typical security model.

MEMORY AND SPACE EFFICIENT DATA INTEGRITY CHECKING SCHEME ON REAL TIME PUBLIC CLOUDS BASED ON NOVEL HASHING ALGORITHMS

Overview

we present a novel efficient RDPC scheme by using homomorphic hash function,

Login

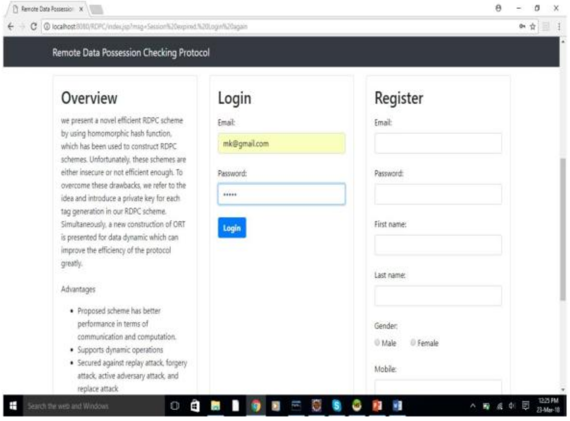
Email:

Register

Email:

Fig 4: Index Page

The index page gets loaded as shown in the above figure Fig 4. The user can login if he has an account. Else, user must first register and then login.


localhost:8080/RDPC/index.jsp?msg=Session%20expired.%20Login%20again

Remote Data Possession Checking Protocol

Overview

we present a novel efficient RDPC scheme by using homomorphic hash function, which has been used to construct RDPC schemes. Unfortunately, these schemes are either insecure or not efficient enough. To overcome these drawbacks, we refer to the idea and introduce a private key for each tag generation in our RDPC scheme. Simultaneously, a new construction of ORT is presented for data dynamic which can improve the efficiency of the protocol greatly.

Advantages

- Proposed scheme has better performance in terms of communication and computation.
- Supports dynamic operations
- Secured against replay attack, forgery attack, active adversary attack, and replace attack

Login

Email: mk@gmail.com

Password: •••••

Register

Email:

Password:

First name:

Last name:

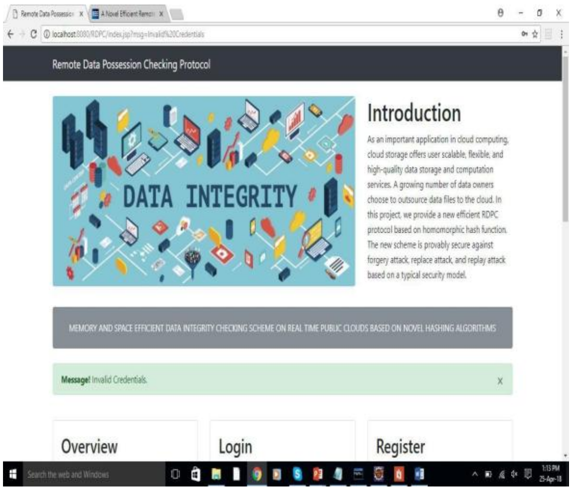
Gender:

○ Male ○ Female

Mobile:

Fig 5: Logging into Account

The above figure Fig 5, shows that the user can login into his account by providing correct email id and password. If credentials are correct, welcome page will be displayed (as shown in figure Fig 7) else a message ‘invalid credentials’ will be displayed on index page as shown in below figure Fig 6.


localhost:8080/RDPC/index.jsp?msg=Invalid%20Credentials

Remote Data Possession Checking Protocol

Introduction

As an important application in cloud computing, cloud storage offers user scalable, flexible, and high-quality data storage and computation services. A growing number of data owners choose to outsource data files to the cloud. In this project, we provide a new efficient RDPC protocol based on homomorphic hash function. The new scheme is provably secure against forgery attack, replace attack, and replay attack based on a typical security model.

MEMORY AND SPACE EFFICIENT DATA INTEGRITY CHECKING SCHEME ON REAL TIME PUBLIC CLOUDS BASED ON NOVEL HASHING ALGORITHMS

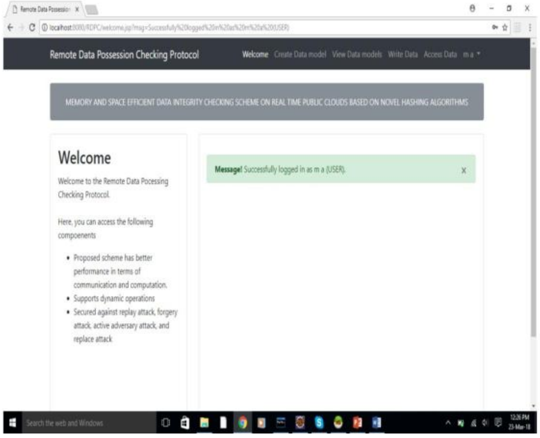
Message! Invalid Credentials.

Overview

Login

Register

Fig 6: When Invalid Credentials are given


localhost:8080/RDPC/welcome.jsp?msg=Successfully%20logged%20in%20as%20m%20a%20(USER)

Remote Data Possession Checking Protocol

MEMORY AND SPACE EFFICIENT DATA INTEGRITY CHECKING SCHEME ON REAL TIME PUBLIC CLOUDS BASED ON NOVEL HASHING ALGORITHMS

Welcome

Welcome to the Remote Data Pocessing Checking Protocol.

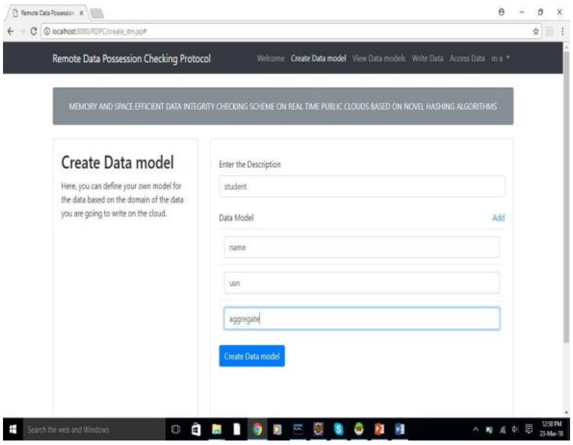
Here, you can access the following compoenents

- Proposed scheme has better performance in terms of communication and computation.
- Supports dynamic operations
- Secured against replay attack, forgery attack, active adversary attack, and replace attack

Message! Successfully logged in as m a (USER).

Fig 7: Welcome Page

This page is displayed when user gives valid credentials.


localhost:8080/RDPC/create_dm.jsp#

Remote Data Possession Checking Protocol

MEMORY AND SPACE EFFICIENT DATA INTEGRITY CHECKING SCHEME ON REAL TIME PUBLIC CLOUDS BASED ON NOVEL HASHING ALGORITHMS

Create Data model

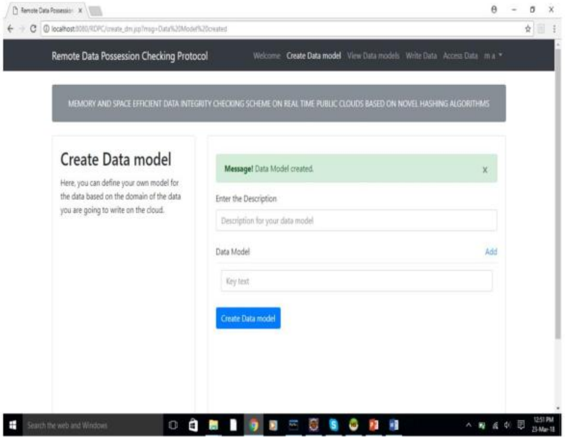
Here, you can define your own model for the data based on the domain of the data you are going to write on the cloud.

Enter the Description: student

Data Model: name, usn, aggregate

Fig 8: Create Data Model Page

The figure Fig 8, shows the create data model page. The user can create data model by giving a description about the data model and then by giving the keys (fields). User can create multiple keys by clicking ‘add’ option. User has to click on ‘create data model’ option to successfully create model. This is shown in below figure Fig 9.


localhost:8080/RDPC/create_dm.jsp?msg=Data%20Model%20created

Remote Data Possession Checking Protocol

MEMORY AND SPACE EFFICIENT DATA INTEGRITY CHECKING SCHEME ON REAL TIME PUBLIC CLOUDS BASED ON NOVEL HASHING ALGORITHMS

Create Data model

Here, you can define your own model for the data based on the domain of the data you are going to write on the cloud.

Message! Data Model created.

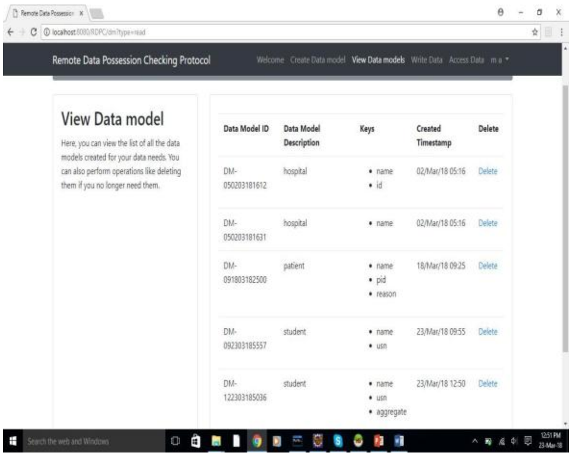
Enter the Description

Description for your data model

Data Model

Key text

Fig 9: Successful Creation of Data Model



localhost:8080/RDPC/dm?type=read

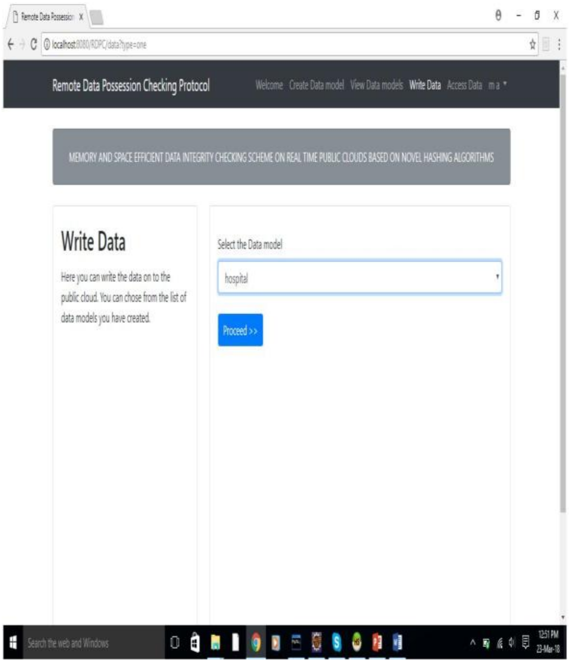
View Data model

Here, you can view the list of all the data models created for your data needs. You can also perform operations like deleting them if you no longer need them.

Data Model ID	Data Model Description	Keys	Created Timestamp	Delete
DM-050203181612	hospital	• name • id	02/Mar/18 05:16	Delete
DM-050203181631	hospital	• name	02/Mar/18 05:16	Delete
DM-091803182500	patient	• name • pid • reason	18/Mar/18 09:25	Delete
DM-092303185557	student	• name • usn	23/Mar/18 09:55	Delete
DM-122303185036	student	• name • usn • aggregate	23/Mar/18 12:50	Delete

Fig 10: View Data Model Page

The figure Fig 10, shows the view data model page. The user can see the details of the data models that he has created. This page shows details like data model id, description, keys, created time, and an option to delete the data model.



localhost:8080/RDPC/data?type=one

MEMORY AND SPACE EFFICIENT DATA INTEGRITY CHECKING SCHEME ON REAL TIME PUBLIC CLOUDS BASED ON NOVEL HASHING ALGORITHMS

Write Data

Here you can write the data on to the public cloud. You can chose from the list of data models you have created.

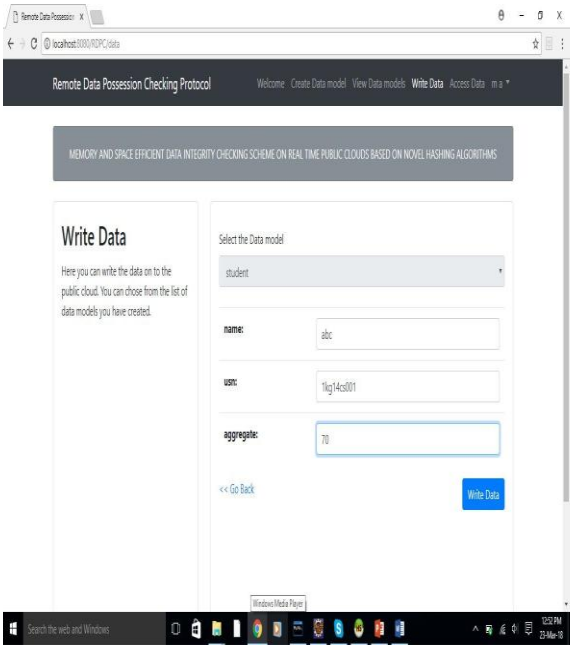
Select the Data model

hospital

Proceed >>

Fig 11: Write Data Model Page

The user can write into the data models created. First the user must select the data model in which data must be written by clicking on drop down option as shown in figure Fig 11. Then the user must click on 'proceed' option. Then the user must fill in values for the keys of data model as shown in figure Fig 12. The encrypted data will then be stored on cloud.



localhost:8080/RDPC/data

MEMORY AND SPACE EFFICIENT DATA INTEGRITY CHECKING SCHEME ON REAL TIME PUBLIC CLOUDS BASED ON NOVEL HASHING ALGORITHMS

Write Data

Here you can write the data on to the public cloud. You can chose from the list of data models you have created.

Select the Data model

student

name: abc

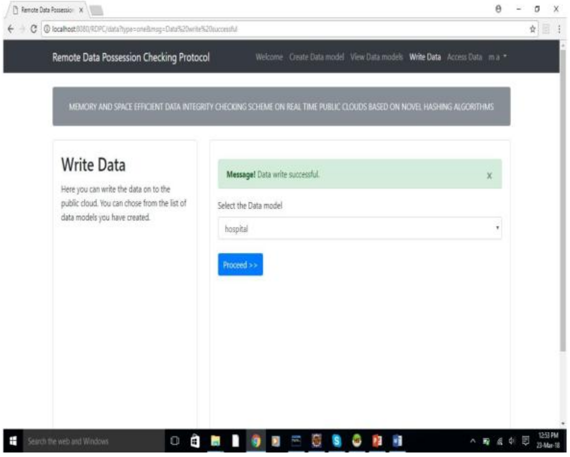
usn: 1kg14cs001

aggregate: 70

<< Go Back

Write Data

Fig 12: Entering Values in Data Models



MEMORY AND SPACE EFFICIENT DATA INTEGRITY CHECKING SCHEME ON REAL TIME PUBLIC CLOUDS BASED ON NOVEL HASHING ALGORITHMS

Write Data

Here you can write the data on to the public cloud. You can chose from the list of data models you have created.

Message! Data write successful.

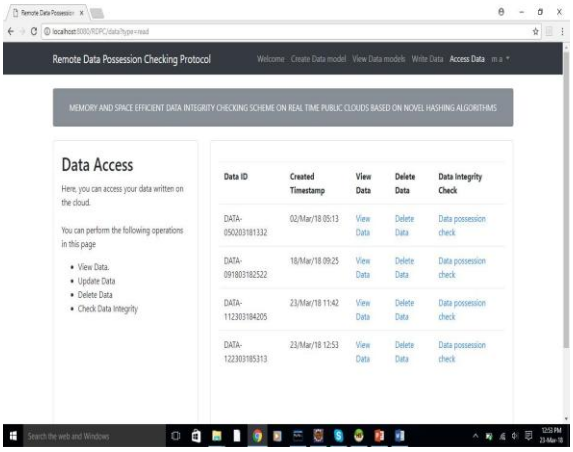
Select the Data model

hospital

Proceed >>

Fig 13: Successful Data Write Operation

The user has to click on 'Write Data' option to successfully write data into the model. The success message is shown in above figure Fig 13.



localhost:8080/RDPC/data?type=read

MEMORY AND SPACE EFFICIENT DATA INTEGRITY CHECKING SCHEME ON REAL TIME PUBLIC CLOUDS BASED ON NOVEL HASHING ALGORITHMS

Data Access

Here, you can access your data written on the cloud.

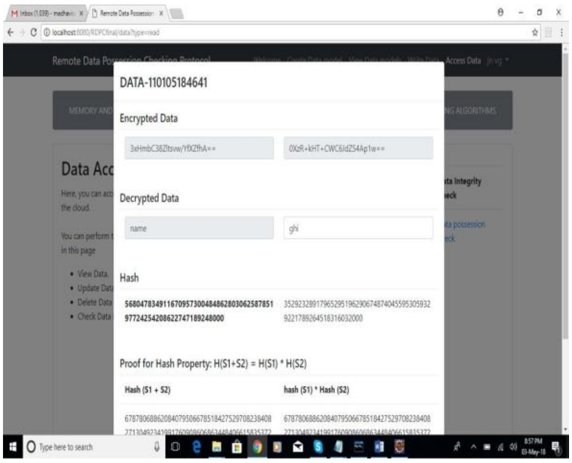
You can perform the following operations in this page

- View Data.
- Update Data
- Delete Data
- Check Data Integrity

Data ID	Created Timestamp	View Data	Delete Data	Data Integrity Check
DATA-050203181332	02/Mar/18 05:13	View Data	Delete Data	Data possession check
DATA-091803182522	18/Mar/18 09:25	View Data	Delete Data	Data possession check
DATA-112303184205	23/Mar/18 11:42	View Data	Delete Data	Data possession check
DATA-122303185313	23/Mar/18 12:53	View Data	Delete Data	Data possession check

Fig 14: Data Access Page

The above figure Fig 14, shows the details of the data access page. Once the user has created data model and writes into the data model, he can access the data to check for possession. This page displays details like data id, created time, user can view particular data, delete data and also can check integrity of data.



DATA-110105184641

Encrypted Data

3xHmbC38ZItsvw/YfXZfhA== | 0XzR+kHT+CWC6IdZS4Ap1w==

Decrypted Data

name | ghi

Hash

5680478349116709573004848628030625878519772425420862274718924800 | 35292328917965295196290674874045595305932922178926451831603200

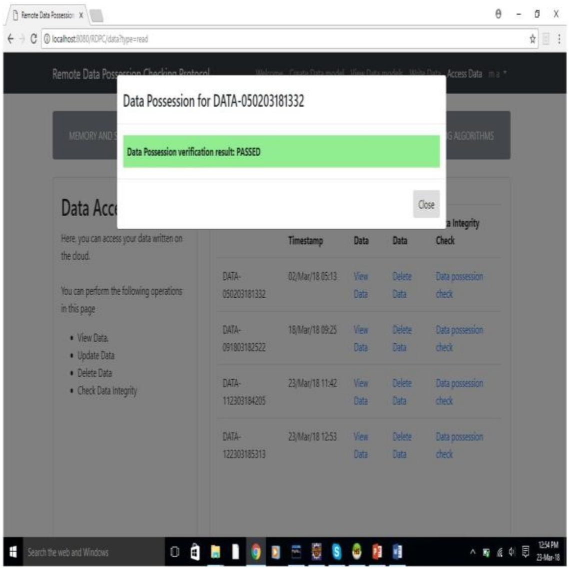
Proof for Hash Property: H(S1+S2) = H(S1) * H(S2)

Hash (S1 + S2)	hash (S1) * Hash (S2)
67878068862084079506678518427529708238408	67878068862084079506678518427529708238408

Fig 15: 'View Data' Option Under Access Data

When the user clicks on 'view data' option, the above details will be displayed as shown in figure Fig 15, hash values will be calculated foe each key and value. The proof for hash property will also be displayed. If the user wants to make changes to data, he can modify and click on 'update' option.

The below figure Fig 16, shows the success status of possession check. The status will be 'passed' if the data is not modified. Else, the status will be displayed as 'failed' as shown in the figure Fig 17.



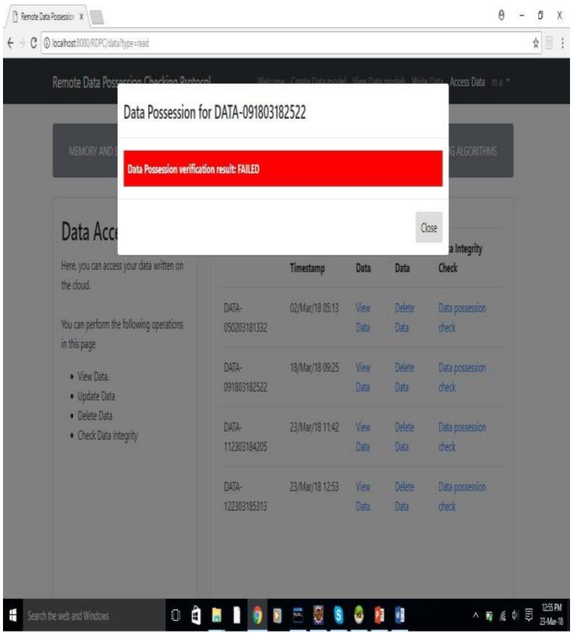
Data Possession for DATA-050203181332

Data Possession verification result: PASSED

	Timestamp	Data	Data	Check
DATA-050203181332	02/Mar/18 05:13	View Data	Delete Data	Data possession check
DATA-091803182522	18/Mar/18 09:25	View Data	Delete Data	Data possession check
DATA-112303184205	23/Mar/18 11:42	View Data	Delete Data	Data possession check
DATA-122303185313	23/Mar/18 12:53	View Data	Delete Data	Data possession check

Remote Data Possession Checking Protocol
Access Data
Close
MEMORY AND S
G ALGORITHMS
Data Acc
Here, you can access your data written on the cloud.
You can perform the following operations in this page
View Data.
Update Data
Delete Data
Check Data Integrity
a Integrity
localhost:8080/RDPC/data?type=read
Search the web and Windows
12:54 PM
23-Mar-18

Fig 16: Data Possession Check



Data Possession for DATA-091803182522

Data Possession verification result: FAILED

	Timestamp	Data	Data	Check
DATA-050203181332	02/Mar/18 05:13	View Data	Delete Data	Data possession check
DATA-091803182522	18/Mar/18 09:25	View Data	Delete Data	Data possession check
DATA-112303184205	23/Mar/18 11:42	View Data	Delete Data	Data possession check
DATA-122303185313	23/Mar/18 12:53	View Data	Delete Data	Data possession check

Remote Data Possession Checking Protocol
Access Data
Close
MEMORY AND S
G ALGORITHMS
Data Acc
Here, you can access your data written on the cloud.
You can perform the following operations in this page
View Data.
Update Data
Delete Data
Check Data Integrity
a Integrity
localhost:8080/RDPC/data?type=read
Search the web and Windows
12:55 PM
23-Mar-18

Fig 17: Data Possession Check

VIII. ADVANTAGES

The Proposed scheme:

- Has better performance in terms of communication and computation.
- Supports dynamic operations.

IX. CONCLUSION AND FUTURE WORK

A. Conclusion

In the proposed system, we study the issue for integrity checking of data files outsourced to remote server and propose an efficient secure RDPC protocol with. Our scheme employs a homomorphic hash function to verify the integrity for the files stored on remote server.

We design data model approach to store data onto the cloud. Our system supports data dynamics i.e., using our new data structure, the data owner can perform insert, modify or delete operation on file blocks with high efficiency. If data read by the user is the same data that the user had written on the cloud, then the possession check is passed. If the data on cloud is modified, then user reads modified data i.e., possession check is failed.

B. Future Enhancement

In future, we wish to

- Extend our system by integrating it with another system which identifies the changed data and retrieves the original data.
- Develop an android based application to access our portal.

REFERENCES

[1] Hao Yan, Jiguo Li, Jinguang Han and Yichen Zhang, "A Novel Efficient Remote Data Possession Checking Protocol in Cloud Storage", IEEE Transactions on Information Forensics and Security , vol. 12, no. 1, pp. 78-88, 2017.

[2] Q.Wang, C.Wang, K.Ren, W.Lou and J.Li “Enabling public auditability and data dynamics for storage security in cloud computing”,IEEE Trans. Parrael Disrib. Syst, vol.22, pp. 847-859, 2011.

[3] Y.Deswarte, J.J.Quisquater and A.Saidane, “Remote integrity checking”, Proc. 6th Work. Conf. Integr. Int. Control Inf. Syst. (IICIS), vol. 16, pp. 1-11, 2003.

[4] G.Ateniese, R.Di.Pietro, L.V.Mancin and G,Tsudik "Scalable and efficient provable data possession," Proc. 4th Int. Conf. Secur. Privacy Commun. Netw. (SecureComm), vol. 9, 2008.

[5] L.Chen, S.Zhou, X.Huang and L.Xu "Data dynamics for remote data possession checking in cloud storage," Comput. Electr. Eng, vol. 39, pp. 2413-2424, 2013.

[6] Wikipedia.