Data Security in Cloud Computing Using Biometrics

A Seminar Report

Submitted in a partial fulfilment of the requirement for the degree of

Master of Technology
In
Information Technology

By
Nitin Yadav
Roll No. 12IT60R06

Under the supervision of

Dr. Debasis Samanta

School of Information Technology
Indian Institute of Technology Kharagpur
Kharagpur-721302, India

CERTIFICATE

This is to certify that the Seminar entitled "**Data Security in Cloud Computing Using Biometrics**" has been submitted by **Nitin Yadav**under my guidance in partial fulfilment of the award of the degree of Master of Technology in Information Technology at School of Information Technology, IIT Kharagpur during the academic year 2012-2013 (Semester-II).

Date: 4thApril, 2013
Place: Kharagpur

Seminar Guide	**Seminar In-charge**
Dr. Debasis Samanta	**Dr. Debasis Samanta**
Associate Professor	**Associate Professor**
Information Technology	**Information Technology**
IIT Kharagpur	**IIT Kharagpur**

Acknowledgement

It is matter of great pleasure for me to submit this seminar report on **“Data Security in Cloud Computing Using Biometrics”**, as a part of curriculum for the award of the degree of Master of Technology in Information Technology at School of Information Technology, IIT Kharagpur. I am thankful to my seminar guide **Dr. Debasis Samanta**, Associate Professor, SIT, IIT Kharagpur for his constant encouragement and guidance.

I take this opportunity to express my deep sense of gratitude towards those, who have helped me in various ways, for preparing my seminar report.

Nitin Yadav
12IT60R06

Date: 4th April, 2013
Place: Kharagpur

Table of Contents

Abstract:

Cloud Computing becomes the next generationarchitecture of IT Enterprise. In contrast to traditional solutions,Cloud computing moves the application software and databasesto the large data centers, where the management of the data andservices may not be fully trustworthy. This unique feature,however, raises many new security challenges which have notbeen well understood. In cloud computing, both data andsoftware are fully not contained on the user's computer; DataSecurity concerns arising because both user data and programare residing in Provider Premises.
Clouds typically have a singlesecurity architecture but have many customers with differentdemands. Every cloud provider solves this problem by encryptingthe data by using encryption algorithms.
In this seminar we focus on the cloud data storage security, which has always been an important aspect of quality of service. To ensure the data correctness of user’s data in the cloud, we propose an effective and flexible distributed scheme. By utilizing the token with distributed verification of erasure-coded data, our scheme achieves the integration of storage correctness insurance and data error localization i.e., the identification of misbehaving server(s).
Keywords: Security, Error correcting code, Biometrics, Cloud Computing.

1. Introduction:

In the traditional model of computing, both data andsoftware are fully contained on the user's computer; in cloudcomputing, the user's computer may contain almost no softwareor data (perhaps a minimal operating system and web browser,display terminal for processes occurring on a network).Cloud computing is based on five attributes: multi-tenancy(shared resources), massive scalability, elasticity, pay as yougo, and self-provisioning of resources, it makes new advancesin processors, Virtualization technology, disk storage,broadband Internet connection, and fast, inexpensive servershave combined to make the cloud a more compelling solution.

The main attributes of cloud computing are illustrated asfollows [4]:

1. Multi-tenancy (shared resources): Cloud computing is based on a business model in which resources are shared (i.e., multiple users use the same resource) at the network level, host level, and application level.
2. Massive scalability: Cloud computing provides the ability to scale to tens of thousands of systems, as well as the ability to massively scale bandwidth and storage space.
3. Elasticity: Users can rapidly increase and decrease their computing resources as needed.
4. Pay as you used: Users to pay for only the resources they actually use and for only the time they require them.
5. Self-provisioning of resources: Users self-provision resources, such as additional systems (processing capability, software, storage) and network resources.

Moving data into the cloud offers great convenienceto users since they don’t have to care about the complexitiesof direct hardware management. The pioneerof Cloud Computing vendors, Amazon Simple StorageService (S3) and Amazon Elastic Compute Cloud (EC2) are both well-known examples. While theseinternet-based online services do provide hugeamountsof storage space and customizable computing resources,this computing platform shift, however, is eliminatingthe responsibility of local machines for data maintenanceat the same time. As a result, users are at the mercyof their cloud service providers for the availability and integrity of their data.

Cloud computing poses new challenging security threats for number of reasons.

1. Traditional cryptographic primitivesfor the purpose of data security protection cannot be directlyadopted due to the users’ loss control of data under CloudComputing. Therefore, verification of correct data storagein the cloud must be conducted without explicit knowledgeof the whole data.
2. Cloud Computing is not just a third party data warehouse. The data stored in the cloud may be frequently updated by the users, including insertion, deletion, modification, appending, reordering, etc.
3. The deployment of Cloud Computing is powered by data centers running in a simultaneous, cooperated and distributed manner. Individual user’s data is

redundantly stored in multiple physical locations to further reduce the data integrity threats.

2.System Model for Cloud Data Storage

A representative network architecture for cloud data storageis illustrated in Figure1. There are three different network entities in the model:

- Users: Users, who have data to be stored in the cloud and rely on the cloud for data computation, consist of both individual consumers and organizations.
- Cloud Service Provider (CSP): A CSP, who has significant resources and expertise in building and managing distributed cloud storage servers, owns and operates live Cloud Computing systems.
- Third Party Auditor (TPA): an optional TPA, who has expertise and capabilities that users may not have, is trusted to assess and expose risk of cloud storage services on behalf of the users upon request.

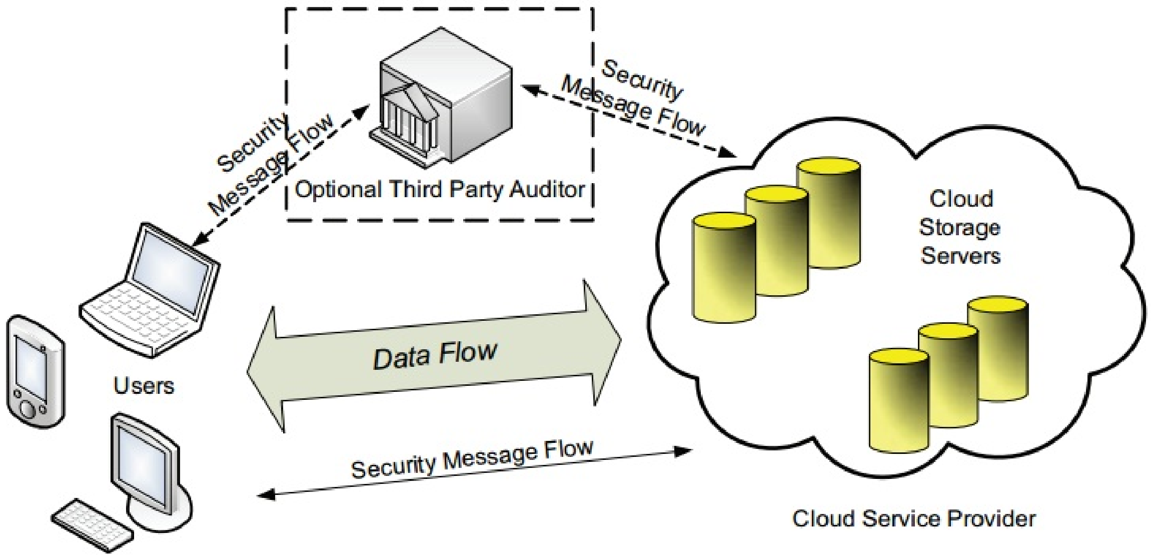


Figure 1: Cloud Data Storage Architecture [1]

In cloud data storage, a user stores his data through a CSPinto a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Data redundancycan be employed with technique of erasure-correcting codeto further tolerate faults or server crash as user’s data growsin size and importance. Thereafter, for application purposes,the user interacts with the cloud servers via CSP to access orretrieve his data.As users no longer possess their data locally, it is of criticalimportance to assure users that their data are being correctlystored and maintained. That is, users should be equipped withsecurity means so that they can make continuous correctnessassurance of their stored data even without the existence oflocal copies.

3. Data Security Problem in Cloud Computing

A security threat faced by cloud data storage can comefrom two different sources.

- A CSP can be self-interested, untrusted and possibly malicious and it may also attempt to hide a data loss incident due to management errors, Byzantine failures and so on.
- There may also exist an economicallymotivated adversary, who has the capability to compromise a number of cloud data storage servers in different time intervals and subsequently is able to modify or delete users' data while remaining undetected by CSPs for a certain period.

There are two types of adversary with different level of capability:

- Weak Adversary: The adversary is interested in corrupting the user's data files stored on individual servers.
- Strong Adversary: This is the worst case scenario, in which we assume that the adversary can compromise all the storage servers so that he can intentionally modify the data files as long as they are internally consistent.

Byzantine Failures: It encompasses those faults that are commonly referred to as "crash failures" and "send and omission failures". When a Byzantine failure has occurred, the system may respond in any unpredictable way, unless it is designed to have Byzantine fault tolerance.

4. Design Goals

There are certain goals for ensuring the security and dependability for cloud datastorage under the aforementioned adversary model. They are:

- Storage Correctness: To ensure users that their data are indeed stored appropriately and kept intact all the time in the cloud.
- Fast localization of data error: To effectively locate the malfunctioning server when data corruption has been detected.
- Dynamic data support: To maintain the same level of storage correctness assurance even if users modify, delete or append their data files in the cloud.
- Dependability: To enhance data availability against Byzantine failures, malicious data modification and server colluding attacks.

5. Brief Introduction of Error Correcting Codes

5.1. Simple Error correcting and error detecting codes

In an error correcting code, a message M is encoded as a sequence of symbols a_1, a2………a_n, called a codeword. The set of possible symbols is fixed in advance; for instance each symbol might be a byte (8 bits) of binary data. The code incorporates some redundancy, so that if some of the symbols in a codeword are changed, we can still figure out what the original message must have been. Usually, the number of symbols in the codeword, n, is also fixed, so each message carries a fixed amount of data. To encode larger amounts of data, one would break it up into a number of messages and encode each one separately.

5.2. Algebraic Code (Reed-Solomon code)

Encoding using Reed-Solomon Code:

Let p be a prime number and let m <= n <= p. The Reed-Solomon code over the field Z_p with m message symbols and n code symbols is defined as follows. Given a message vector$[x_1 x_2 \ldots x_m]$, let P(t) be the polynomial

$$P(t) = x_m t^{m-1} + x_{m-1} t^{m-2} + \ldots\ldots\ldots x_2 t + x_1$$

with coefficients given by the message symbols. Thus P (t) is a polynomial of degree at most m-1 in one variable t, with coefficients in Z_p. Then the code vector a for this message vector is the list of the first n values of the polynomial P (t):

$$a = [a_1 a_2 \ldots a_n] = [P(0)\ P(1) \ldots P(n-1)]$$

(Evaluated using modular arithmetic in Z_p).

The Reed-Solomon code over Z_p with m message symbols and n code symbols is the linear code with matrix

$$C = \begin{bmatrix} 1 & 1 & 1 & \ldots\ldots\ldots & 1 \\ 0 & 1 & 2.\ldots\ldots\ldots & & (n-1) \\ 0^2 & 1^2 & 2^2\ldots\ldots\ldots & & (n-1)^2 \\ 0^{m-1} & 1^{m-1} & 2^{m-1}\ldots\ldots & & (n-1)^{m-1} \end{bmatrix}$$

all entries taken (mod p).

Now encoding can be done as

$$a = xC \qquad (1)$$

where, a = encode matrix , x = original message matrix, and C = Reed-Solomon code matrix.

Decoding Reed-Solomon Codes:

The discussion in this section will always refer to the Reed-Solomon code over Z_p with

m = number of message symbols;
n = number of code symbols
n= m + 2e;
e = number of errors the code can correct.

Let the transmitted code vector be [P(0) P(1) . . . P(n-1)], and the received vector be [R_0 R_1 R_3 . . . R_{n-1}]. Assume there are at most e errors, that is, at moste values I suck that R_i != P(i).

Then there exist non-zero polynomials

E(t) of degree <=e

Q(t) of degree <= m+e-1,

Such that

$$Q(i) = R_iE(i) \quad \text{for all } i = 0, 1, 2, \ldots, n. \qquad (2)$$

If E(t) and Q(t) satisfy the equation (2), and the number of errors is at most e, then Q(t) = P(t)E(t).

Hence, we can compute P(t) as Q(t)/E(t).

6. Ensuring Security in Cloud Data Storage

Our main scheme for ensuringcloud data storage is presented in this section. The first part ofthe section is devoted to a review of basic tools from codingtheory that are needed in our scheme for file distribution acrosscloud servers. Then, the homomorphic token is introduced.The token computation function we are considering belongsto a family of universal hash function, chosen to preserve the homomorphic properties, which can be perfectlyintegrated with the verification of erasure-coded data.

Subsequently, it is also shown how to derive a challengeresponse protocol for verifying the storage correctness as wellas identifying misbehaving servers. Finally, the procedure forfile retrieval and error recovery based on erasure-correctingcode is outlined.

Notation and Preliminaries

F - The data file to be stored. We assume that F can be denoted as a matrix of m equal-sized data vectors, each consisting of l blocks.

A -The dispersal matrix used for Reed-Solomon coding.

G-The encoded file matrix, which includes a set ofn = m + k vectors, each consisting of l blocks.

f_{key}(.) -Pseudorandom function (PRF)

ϕ_{key}(.) - Pseudorandom permutation (PRP)

Biometrics:

"Biometrics" [5] means "life measurement" but the term is usually associated with the use of unique physiological characteristics to identify an individual. The application which most people associate with biometrics is security. However, biometrics identification has eventually a much broader relevance as computer interface becomes more natural. Knowing the person with whom you are conversing is an important part of human interaction and one expects computers of the future to have the same capabilities.

A biometric system can be either an 'identification' system or a 'verification' (authentication) system, which are defined below.

Identification: One to Many: Biometrics can be used to determine a person's identity even without his knowledge or consent. For example, scanning a crowd with a camera and using face recognition technology, one can determine matches against a known database.

Verification: One to One: Biometrics can also be used to verify a person's identity. For example, one can grant physical access to a secure area in a building by using finger scans or can grant access to a bank account at an ATM by using retinal scan.

Example:

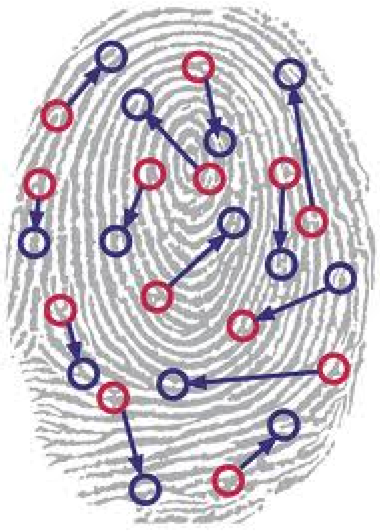


Figure . 2 - Finger Print

Use of Biometrics:

In this work biometric is used to extract the features from a finger print and use it to generate the key for pseudorandom function and pseudorandom permutation which are going to be used for implementing the model to enhance data security in cloud computing.

- **File Distribution Preparation**

  Incloud data storage, we rely on this technique to disperse thedata file F redundantly across a set of n = m+ k distributedservers. A (m + k, k) Reed-Solomon erasure-correcting codeis used to create k redundancy parity vectors from m datavectors in such a way that the original m data vectors can bereconstructed from any m out of the m + k data and parityvectors. By placing each of the m + k vectors on a differentserver, the original data file can survive the failure of any k ofthe m+k servers without any data loss.

  Let F= {F_1 , F_2 , …… F_m] is the data file to be stored.The systematic layout with parityvectors is achieved with the information dispersal matrix A.Note that A is derivedfrom a Vandermonde matrix, thus it has the property that anym out of the m+ k columns form an invertible matrix.

By multiplying F by A, the user obtains the encoded file:

$$G = F . A = (G^{(1)}, G^{(2)}, G^{(3)}, G^{(4)}, ..., G^{(m)}, G^{(m+1)}, ..., G^{(n)},)$$

where $G^{(j)} = (g_1^{(j)}, g_2^{(j)}, ..., g_1^{(j)})^T \; (j \in \{1, ...n\})$.

- **Challenge Token Computation**

The main idea is as follows: before file distribution the user pre-computes a certain number of short verification tokens on individual vector G(j) (j ∈ {1, . . . , n}), each token covering a random subsetof data blocks. Later, when the user wants to make sure the storage correctness for the data in the cloud, he challenges the cloud servers with a set of randomly generated block indices. Upon receiving challenge, each cloud server computes a short "signature" over the specified blocks and returns them to the user. The values of these signatures should match the corresponding tokens pre-computed by the user.

Suppose the user wants to challenge the cloud servers t times to ensure the correctness of data storage. Then, he must pre-compute t verification tokens for each $G^{(j)} \; (j \in \{1, ...n\})$, using a PRF f(.), a PRP $\Phi(.)$, a challenge key k_{chal} and a master permutation key K_{PRP}. To generate the i^{th} token for server j, the user acts as follows:

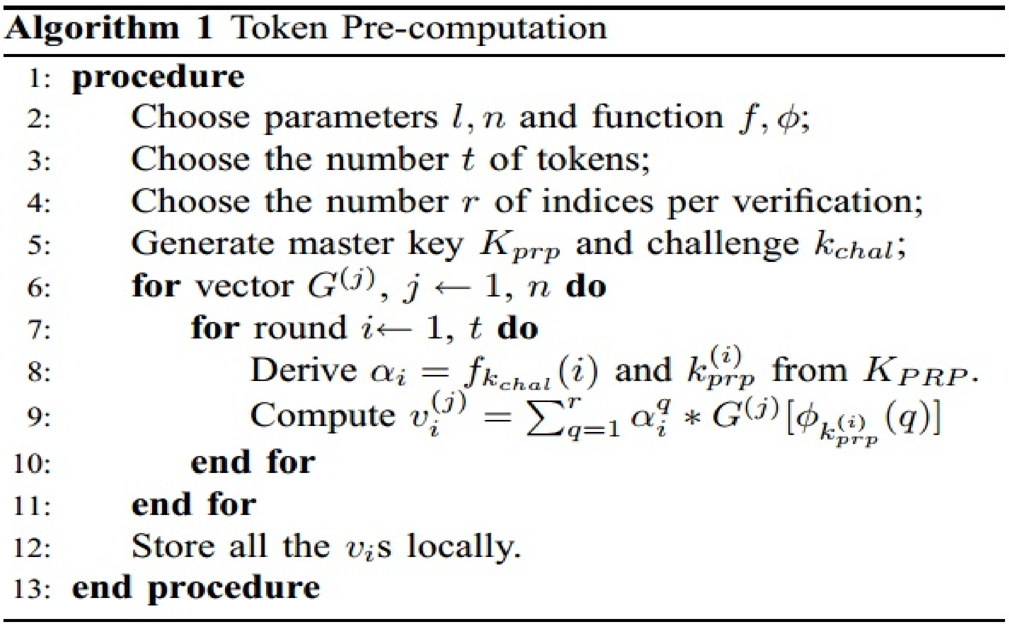
1. Derive a random challenge value α_i of GF(2^p) by α_i = f_{kchal}(i)and a permutation key $K^{(i)}{}_{PRP}$ based on K_{PRP}.
2. Compute the set of randomly chosen indices:

$$\{I_q \in [1, ... l] \;\; | \, 1 \le q \le r\}, \; where \;\; I_q = \phi^{(i)}_{k_{PRP}} \; (q).$$

3. Calculate the token as:

$$v_i^{(j)} = \sum_{q=1}^{r} \alpha_i^q * G^{(j)}[I_q], \;\; where \;\; G^{(j)}[I_q] = g^{(j)}_{I_q}.$$

Algorithm can be given as:



Algorithm 1 Token Pre-computation

```
1:  procedure
2:      Choose parameters $l, n$ and function $f, \phi$;
3:      Choose the number $t$ of tokens;
4:      Choose the number $r$ of indices per verification;
5:      Generate master key $K_{prp}$ and challenge $k_{chal}$;
6:      for vector $G^{(j)}$, $j \leftarrow 1, n$ do
7:          for round $i \leftarrow 1, t$ do
8:              Derive $\alpha_i = f_{k_{chal}}(i)$ and $k_{prp}^{(i)}$ from $K_{PRP}$.
9:              Compute $v_i^{(j)} = \sum_{q=1}^{r} \alpha_i^q * G^{(j)}[\phi_{k_{prp}^{(i)}}(q)]$
10:         end for
11:     end for
12:     Store all the $v_i$s locally.
13: end procedure
```

Figure 3. : Algorithm of Token-Pre-computation

- **Correctness Verification and Error Localization:**

  The procedure of the ith challenge-response fora cross-check over the n servers is described as follows:

  1. The user reveals the αi as well as the ith permutation key $k^{(i)}{}_{PRP}$ to each servers.
  2. The server storing vector $G^{(j)}$ aggregates those r rows specified by index $k^{(i)}{}_{PRP}$ into a linear combination

     $$R_i^{(j)} = \sum_{q=1}^{r} \alpha_i^q * G^{(j)}[\phi_{k_{PRP}^{(i)}}(q)].$$

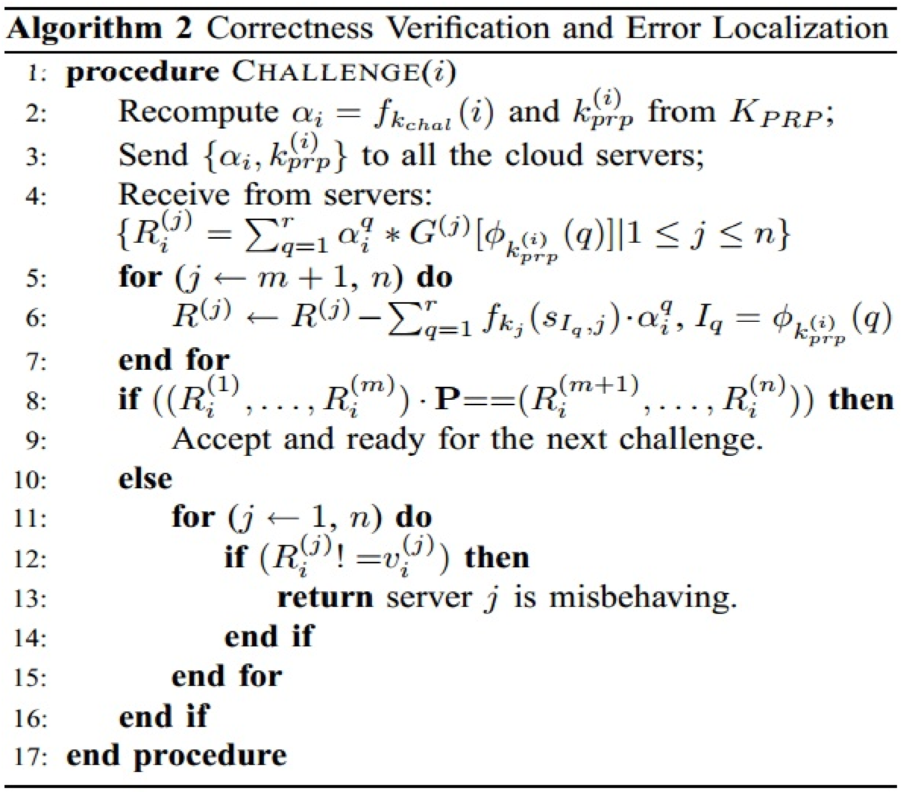
  3. Upon receiving $R_i^{(j)}$ s from all the servers, the user takes away blind values in $R^{(j)}$ $(j \in \{m+1, \ldots\ldots\ldots n\})$ by

     $$R_i^{(j)} \leftarrow \mathrm{R}_i^{(j)} - \sum_{q=1}^{r} f_{k_j}(SI_q.j).\alpha_i^q, \ where \ \ I_q = \phi_{k_{PRP}^{(i)}}(q)$$

  4. Then the user verifies whether the received values remain a valid codeword determined by secret matrix P:

     $$(R_i^{(1)}, \ldots\ldots, R_i^{(m)}).P \overset{?}{=} (R_i^{(m+1)}, \ \ldots\ \ldots\ldots, \ R_i^{(n)}).$$

Algorithm can be given as:



```
Algorithm 2 Correctness Verification and Error Localization
procedure CHALLENGE($i$)
    Recompute $\alpha_i = f_{k_{chal}}(i)$ and $k_{prp}^{(i)}$ from $K_{PRP}$;
    Send $\{\alpha_i, k_{prp}^{(i)}\}$ to all the cloud servers;
    Receive from servers:
    $\{R_i^{(j)} = \sum_{q=1}^{r} \alpha_i^q * G^{(j)}[\phi_{k_{prp}^{(i)}}(q)] | 1 \leq j \leq n\}$
    for ($j \leftarrow m+1, n$) do
        $R^{(j)} \leftarrow R^{(j)} - \sum_{q=1}^{r} f_{k_j}(s_{I_q,j}) \cdot \alpha_i^q, I_q = \phi_{k_{prp}^{(i)}}(q)$
    end for
    if $((R_i^{(1)}, \ldots, R_i^{(m)}) \cdot \mathbf{P} == (R_i^{(m+1)}, \ldots, R_i^{(n)}))$ then
        Accept and ready for the next challenge.
    else
        for ($j \leftarrow 1, n$) do
            if ($R_i^{(j)} ! = v_i^{(j)}$) then
                return server $j$ is misbehaving.
            end if
        end for
    end if
end procedure
```

Figure 4. : Algorithm of Correctness Verification and Error Localization

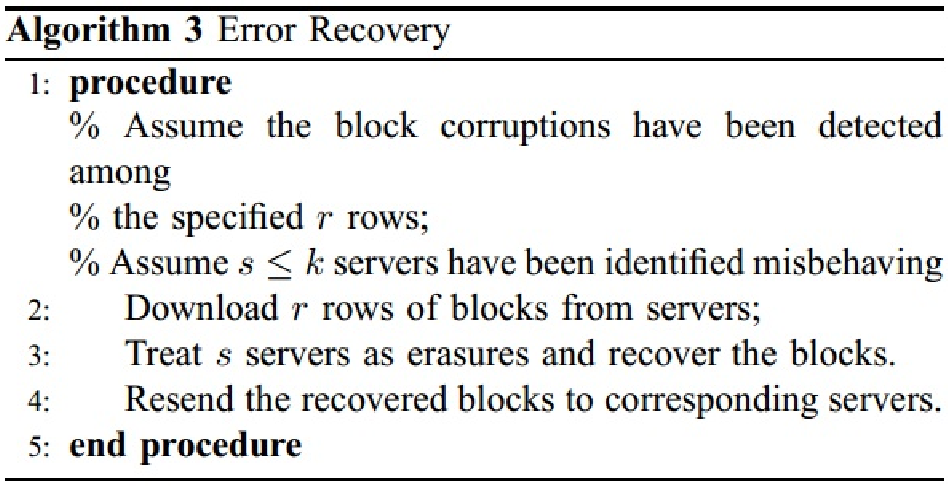
Because all the servers operate over the same subset ofindices, the linear aggregation of these r specified rowshas to be a codeword in the encoded filematrix. If the above equation holds, the challenge is passed.Otherwise, it indicates that among those specified rows, thereexist file block corruptions.

Once the inconsistency among the storage has been successfully detected, we can rely on the pre-computed verificationtokens to further determine where the potential data error(s)lies in.

- **File Retrieval and Error Recovery**

  Whenever the datacorruption is detected, the comparison of pre-computed tokensand received response values can guarantee the identificationof misbehaving server(s), again with high probability, whichwill be discussed shortly. Therefore, the user can alwaysask servers to send back blocks of the r rows specified inthe challenge and regenerate the correct blocks by erasurecorrection, shown in Algorithm 3, as long as there are at mostk misbehaving servers are identified. The newly recoveredblocks can then be redistributed to the misbehaving serversto maintain the correctness of storage.

Algorithm can be given as:



Algorithm 3 Error Recovery

```
1: procedure
   % Assume the block corruptions have been detected among
   % the specified $r$ rows;
   % Assume $s \leq k$ servers have been identified misbehaving
2:     Download $r$ rows of blocks from servers;
3:     Treat $s$ servers as erasures and recover the blocks.
4:     Resend the recovered blocks to corresponding servers.
5: end procedure
```

Figure 5. : Algorithm of Error Recovery

7. Conclusion:

To ensure the correctness of users' data in clouddata storage we rely on erasure-correcting codein the file distribution preparation to provide redundancy parityvectors and guarantee the data dependability. By utilizing thehomomorphic token with distributed verification of erasurecoded data, our scheme achieves the integration of storage correctness insurance and data error localization, i.e., wheneverdata corruption has been detected during the storage correctness verification across the distributed servers, we can almostguarantee the simultaneous identification of the misbehavingserver(s).

8. References:

[1]. Cong Wang, Qian Wang, and KuiRenand Wenjing Lou "*Ensuring Data Security in Cloud Computing*"Illinois Institute of Technology.

[2Cong Wang,Qian Wang, KuiRen, Ning Cao, and Wenjing Lou*"Towards Secure and Dependable Storage Service in Cloud*" IEEE Journal.

[3]. K. ValliMadhavi, R. Tamilkodi, R. BalaDinakar "*Data Storage Security in Cloud Computing for Ensuring Effective and Flexible Distributed System*" National conference on Research Trends in Computer Science and Technology -2012

[4]. Center Of The Protection Of National Infrastructure CPNI by Deloitte *"Information security Briefing 0112010 Cloud computing",* p.71Published March 2010.

[5]. http://www.cse.iitk.ac.in/users/biometrics/pages/what_is_biom_more.htm