IOSR Journal of Computer Engineering (IOSR-JCE)
e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 16, Issue 6, Ver. III (Nov – Dec. 2014), PP 50-56
www.iosrjournals.org

Data storage security in Cloud

Mrs. Niyamat Ujloomwale, Mrs. Ranjana Badre
(Computer Engineering, MIT Academy Of Engineering/Savitribai Phule Pune University, India)
(Computer Engineering, MIT Academy Of Engineering/ Savitribai Phule Pune University, India)

Abstract : *Cloud computing is worthy of consideration and try to build business systems as a way for businesses in this way can undoubtly bring about lower costs, higher profits and more choice; for large scale industry, Data security has become the most important issue of cloud computing security. Though many solutions have been proposed, many of them only considers one side of security ;this paper proposes the cloud data security must be considered to analyze the data security risk, the data security requirements, deployment of security functions and the data security process through encryption. Distribution of file is done on cloud servers with token generation. The security architecture of the system is designed using encoding algorithms, which eliminates the fraud that occurs today with stolen data. There is no danger of any data sent within the system being intercepted, and replaced. The system is acceptably secure, but that the level of encoding has to be stepped up as computing power increases. Results in order to be secured the system the communication between modules is encoded. Since the customer does not have control over data the cloud provider should assure the customer that data is not modified. In this paper a data correctness scheme is proposed in which a cloud service provider assures the user that the data is stored in the cloud is safe. This scheme also achieves the integration of storage correctness insurance and data error localization i.e., the identification of misbehaving server(s).*

Keywords: *challenge**,** data storage, error localization, misbehaving server, storage correctness*

I. Introduction

Organizations today are increasingly looking towards Cloud Computing as a new revolutionary technology promising to cut the cost of development and maintenance and still achieve highly reliable and elastic services.TheCloud technology is a growing trend and is still undergoing lots of experiments. Cloud promises huge cost benefits, agility and scalability to the business. All business data and software are stored on servers at a remote location referred to as Data centres. Data centre environment allows enterprises to run applications faster, with easier manageability and less maintenance effort, and more rapidly scale resources (e.g. servers, storage, and networking) to meet fluctuating business needs. A data center in cloud environment holds information that end-users would more traditionally have stored on their computers. This raise concerns regarding user privacy protection because users must outsource their data. The movement of data to centralized services could affect the privacy and security of users' interactions with the files stored in cloud storage space. The use of virtualized infrastructure as a launching pad might introduce new attacks to user's data.

A formal "Proof of Retrievability" (POR) model for ensuring the remote data integrity was described by A. Juels and J. Burton S. Kaliski in October 2007. Their scheme combines two methods spot-checking and error- correcting code to ensure both possession and retrievability of files on archive or backup service systems. H. Shacham and B.Waters in 2008 built on this model and constructed a random linear function based homomorphic authenticator which enables unlimited number of queries and requires less communication overhead.

An improved framework for POR protocols that generalizes both Juels and Shacham's work was illustrated. All these schemes are focusing on static data. The effectiveness of their schemes rests primarily on the pre-processing steps that the user conducts before outsourcing the data file F. Any change to the contents of F, even few bits, must propagate through the error-correcting code, thus introducing significant computation and communication complexity was proposed by Bowers in 2009.

The "provable data possession" (PDP) model for ensuring possession of file on untrusted storages was defined by Ateniese et al . Their scheme utilized public key based homomorphic tags for auditing the data file, thus providing public verifiability. However, their scheme requires sufficient computation overhead that can be expensive for an entire file. Later in their subsequent work during 2008, described a PDP scheme that uses only symmetric key cryptography. This method has lower-overhead than their previous scheme and allows for block updates, deletions and appends to the stored file, which has also been supported in our work. However, their scheme focuses on single server scenario and does not address small data corruptions, leaving both the distributed scenario and data error recovery issue unexplored.

A new efficient means of polynomial in the size of the input (i.e. key or data) was proposed by M. A. Shah, R.Swaminathan, and M. Baker during the year 2008 in "Privacy Preserving audit and extraction of digital contents". The main threat from the auditor is that it may glean important information from the auditing process that could compromise the privacy guarantees provided by the service. For example, even a few bits from a file containing medical history could reveal whether a customer has a disease. To ensure privacy, there exist different standards for the encrypted data and the encryption key. For the data, the system relies on (1) the strength of the encryption scheme and (2) the zero-knowledge property of the protocol for encryption-key audits.

To ensure file integrity across multiple distributed servers, using erasure-coding and block-level file integrity checks was proposed by T. S. J. Schwarz and E. L. Miller in 2009. However, their scheme only considers static data files.

To verify data integrity using RSA-based hash for data possession in peer-to-peer file sharing networks was defined by D. L. G. Filho and P. S. L. M. Barreto in 2006. However, their proposal requires exponentiation over the entire data file, which is clearly impractical for the server whenever the file is large.

Data integrity is defined as the accuracy and consistency of stored data, in absence of any alteration to the data between two updates of a file or record. Cloud services should ensure data integrity and provide trust to the user privacy. Although outsourcing data into the cloud is economically attractive for the cost and complexity of long-term large-scale data storage, it's lacking of offering strong assurance of data integrity and availability may impede its wide adoption by both enterprise and individual cloud users. Cloud computing poses privacy concerns primarily, because the service provider at any point in time, may access the data that is on the cloud. The Cloud service provider could accidentally or deliberately alter or delete some information from the cloud server. Hence, the system must have some sort of mechanism to ensure the data integrity.

The main contributions of this paper are

a) The challenge-response protocol provides the localization of data error.
b) Proposed an efficient method for encoding the data to be transferred and stored in the Cloud.
c) Finally, proposed an efficient data recovery method for the retrieval of lost data in Cloud.

II. System Model

In cloud data storage, a user stores his data through a CSP into a set of cloud servers, which are running in a simultaneous, cooperated and distributed manner. Data redundancy can be employed with technique of erasure-Correcting code to further tolerate faults or server crash as user's data grows in size and importance. Thereafter, for application purposes, the user interacts with the cloud servers via CSP to access or retrieve his data. In some cases, the user may need to perform block level operations on his data.

The proposed system has three important entities,

User: Users store data in the cloud and depend on the cloud for all its computations on the data stored in the cloud server. User may be an individual or organization.

Cloud Service Provider (CSP): CSP contains resources and expertise in building and managing distributed cloud storage servers, owns and operates and leases the live Cloud computing systems.

Third Party Auditor (TPA): TPA has expertise and capabilities that users may not have, is trusted to assess, audit and expose risk of cloud storage services on behalf of the users upon request from the users.

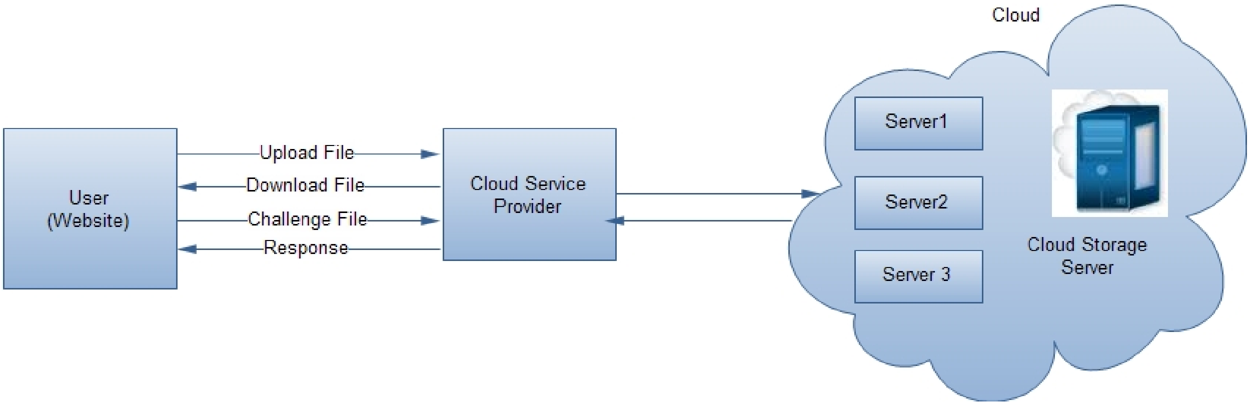


Fig. 1. Cloud data storage architecture

The most general forms of these operations we are considering are block update, delete, insert and append. As users no longer possess their data locally as shown in Fig. 1, it is of critical importance to assure users that their data are being correctly stored and maintained. That is, users should be equipped with security means so that they can make continuous correctness assurance of their stored data even without the existence of local copies. In case that users do not necessarily have the time, feasibility or resources to monitor their data, they can delegate the tasks to an optional trusted TPA of their respective choices. In this model, it is assumed that the point-to-point communication channels between each cloud server and the user is authenticated and reliable, which can be achieved in practice.

III. Mathematical Model

Notations

F is the data file to be stored
A is the dispersal matrix derived from Vandermonde matrix and used for reed Solomen coding
G is the encoded file matrix
PRF is the Pseudorandom function
PRP is the pseudorandom permutation
v is the version number for individual blocks

To verify the correctness of user's data and to locate the errors, the scheme is entirely relied on the pre-computed verification tokens. These tokens are calculated before file distribution and they are very short. The tokens are computed by pseudorandom function and pseudorandom permutation function. Pre-computation of short verification tokens is made on individual vector, each token covering a random subset of data blocks. The scheme is assumed to have block size as 256 bits and as 8 number of verification per indices. Later, when the user wants to make sure the storage correctness for the data in the cloud, he challenges the cloud servers. Upon receiving challenge, cloud server computes the new value of tokens, which is compared with previously calculated tokens.

Once the data corruption is detected, next important step is to recover the corrupted data and bring data storage back to consistent state. The comparison of pre-computed tokens and received response values can guarantee the identification of misbehaving server. Therefore user can recover the corrupted data. The given system recovers data from backup server and distributes all data vectors to corresponding servers. This will results in successful recovery of corrupted data. But due to file splitting we made at the time of file distribution, user's need to recover file from all the servers. Error localization is limited to misbehaving servers only, i.e. servers giving false assurance of posing user's data.

To eliminate the errors in storage systems key prerequisite is to locate the errors. However, many previous schemes do not explicitly consider the problem of data error localization, thus only provide binary results for the storage verification. In this scheme we integrate the correctness verification and error localization in challenge-response protocol. The newly computed tokens from servers for each challenge are compared with pre-computed tokens to determine the correctness of the distributed storage. This also gives information to locate potential data errors.

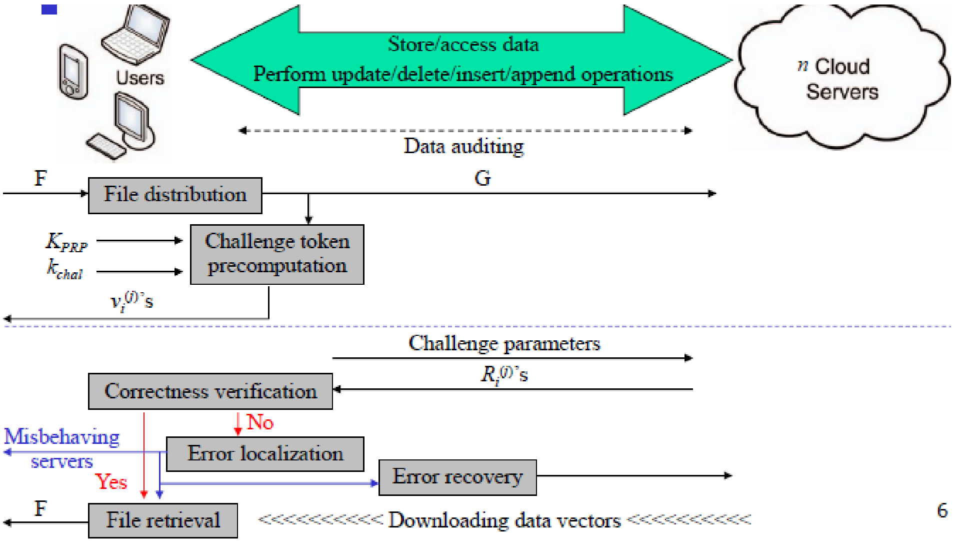


Fig. 2: Proposed scheme

It gives clear idea about integrity of user's data. The proposed scheme is as shown in figure 2.

Again it helps to locate the error which has not been done in previous research work. In previous work, we were just able to detect whether the data is intact or not. So it just provides us with binary results and not the exact location of errors.

Once the data corruption is detected, next important step is to recover the corrupted data and bring data storage back to consistent state. The comparison of pre-computed tokens and received response values can guarantee the identification of misbehaving server. Therefore user can recover the corrupted data. The given system recovers data from backup server and distributes all data vectors to corresponding servers. This will results in successful recovery of corrupted data. But due to file splitting we made at the time of file distribution, user's need to recover file from all the servers. Error localization is limited to misbehaving servers only, i.e. servers giving false assurance of posing user's data.

For current research work, file is provided as input. The aim is to provide security to the file. Token pre-computation algorithm is used to generate tokens from each file byte. For each file block generate the tokens separately and are to be stored on secure server. When user wants to download the file, get all the blocks as a result of single file, thus user can access full file.

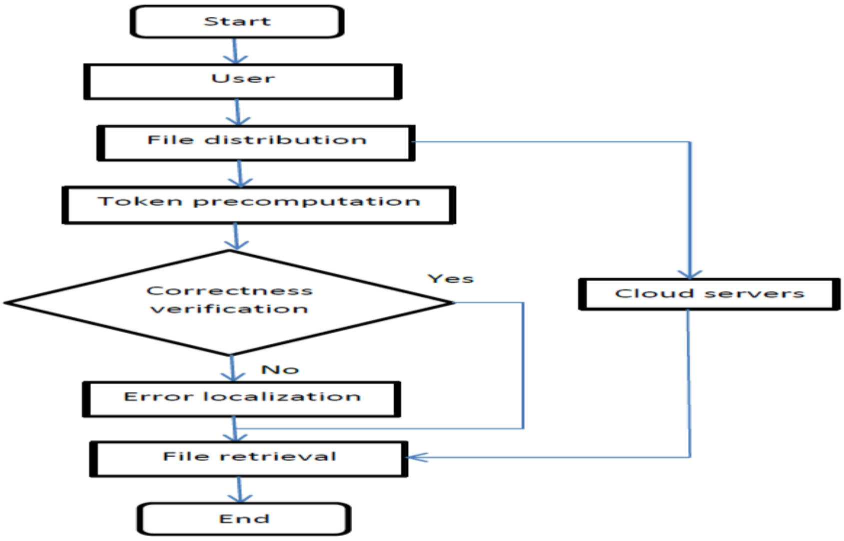


Fig. 3: Data flow diagram

When the file is to be stored on servers, encryption algorithm is used to store the each data of block on separate server. Tokens are used for verification i.e. if user wants to check file, check the tokens against the data stored on server. The data flow diagram is as shown in fig. 3.

To ensure the data present on cloud is correct, confront the servers of cloud with a lot of block indices that are generated in an arbitrary manner and when these confrontations are received, each cloud server computes a short signature over the specified blocks and returns them to the user.

There is a Client Application which contains login module which authenticates user from database. It also contains facility to upload and download a file. In this application we will calculate tokens prior to file upload and stored on client database server. This will allow user to check the version of the file on cloud server. We shall provide user interface to send and check the tokens from server. We will also allow user to recover the file contents if its version gets changed.

Cloud Service provider is the application which manages the client and provides services to the client. It also allocates the required resources to the client. This is used to manage and handle cloud service provider application.

Cloud servers three are the server applications responsible for upload, download and deletion of file.

IV. Problem Solution

Two different network entities can be identified as follows:

- User: 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.

Erasure-correcting code may be used to tolerate multiple failures in distributed storage systems. In cloud data storage, we rely on this technique to disperse the data file F redundantly across a set of d distributed servers. The layer interleaving technique is used to determine the cyclic redundancy parity vectors from r data vectors in such a way that the original r data vectors can be reconstructed from any r out of the r + c data and parity vectors. By placing each of the r + c vectors on a different server, the original data file can survive the failure of any c of the r + c servers without any data loss, with a space overhead of c/r. The unmodified r data file vectors together with c parity vectors are distributed across r + c different servers.

The user obtains the encoded file by multiplying **F** by **A** that is , **G = F · A** = (G(1),G(2), . . . , G(m),G(m+1),. , G(n)) = (F1, F2, . . . , Fm, G(m+1), . . . , G(n)),where F is the actual file and A is derived from a Vandermonde matrix, is a matrix with the terms of a geometric progression in each row. For a interleave index of 3, the first block containing data packets numbered (0,3,6,...(r-1).c), the second with data packets numbered (1,4,7,..,((r-1).c)+1) and the third with data packets numbered (2,5,8,...((r-1).c)+2).

4.1 Token Generation

The main idea is - when a file is distributed to the cloud, the user pre-computes a certain number of short verification tokens on individual vector G(j) (j ϵ{1, . . . , n}), each token covering a random subset of data blocks that would be distributed to the different cloud servers. 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 if the user wants to challenge the cloud server t times to ensure the correctness of data storage, the user must pre-compute x verification tokens k_{chal} and a master permutation key K_{PRP}. To generate for each G(j) (j □ {1, . . . , n}), a challenge key the i^{th} token for server j, the user acts as follows,

1. Derive a random challenge value α_i and a permutation key $k^{(i)}{}_{prp}$ based on K_{PRP}.
2.Compute the set of r randomly-chosen indices.
3.Calculate the token $v(j)_i$ using the random challenge value α_i.

After token generation, the user has the choice of either keeping the pre-computed tokens locally or storing them in encrypted form on the cloud servers

4.2 Correctness Verification

The response values from servers for each challenge not only determine the correctness of the distributed storage, but also contain information to locate potential data error(s). The procedure of the *ith* challenge-response for verification over the d=3 servers is described as follows:

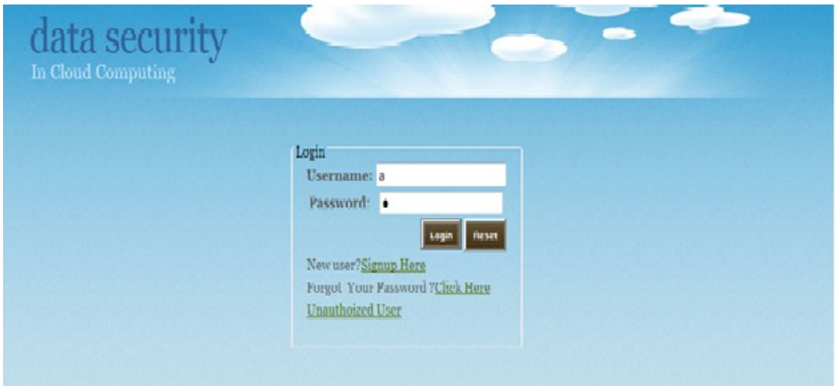
1. The user reveals the permutation key to each server.
2.The server storing vector *G*(*j*) aggregates those k rows specified by index permutation key into a linear combinAtion.
3. Upon receiving linear combination from all the servers, the user takes away blind values.
4. Then the user verifies whether the received values remain a valid codeword determined by secret
matrix P.

4.3 Error Recovery

1 Assume the block corruption has been detected among the specified rows;
Assume s<=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.

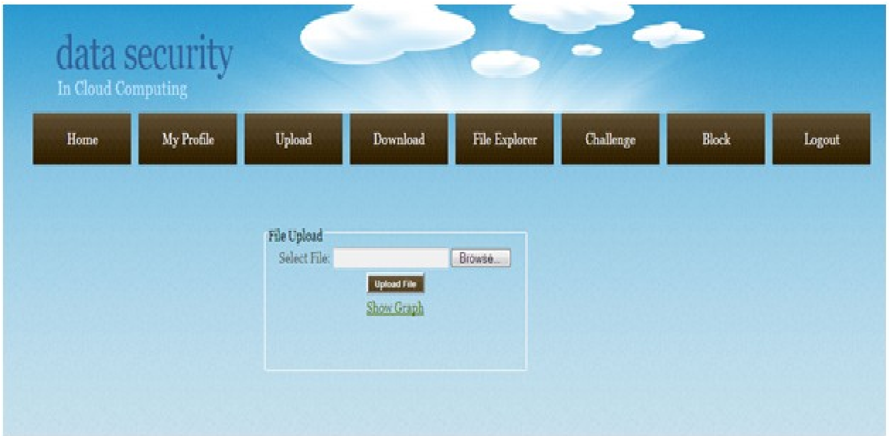
V. Implementation And Result

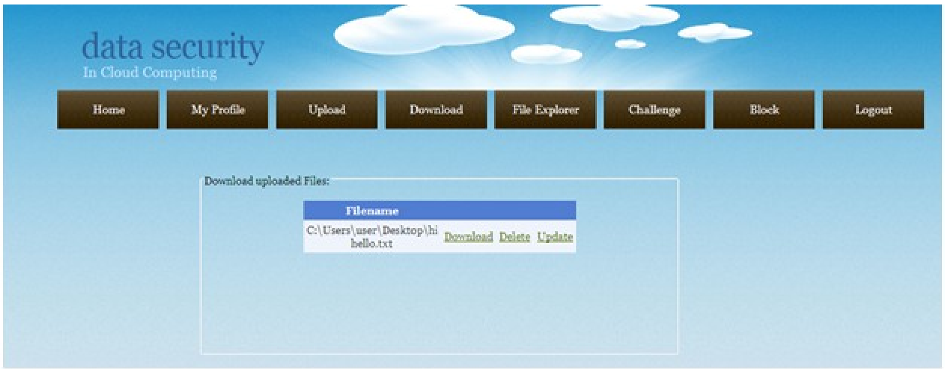
Here are some of the screen shots of the implemented system.



Username: a
Password: •

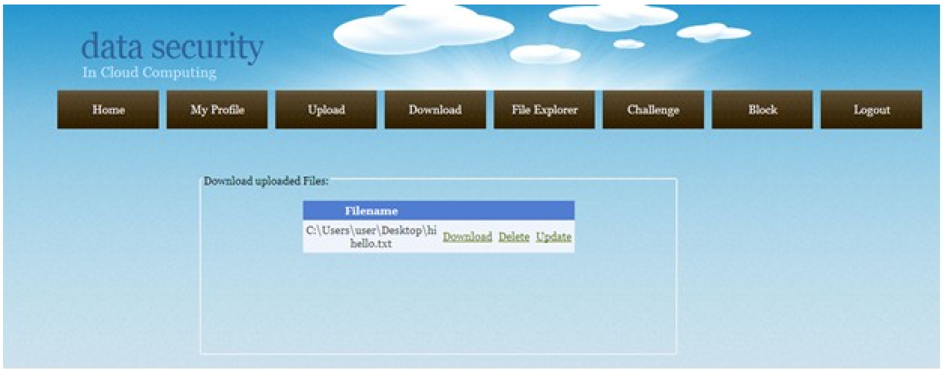
New user logs in having username and password





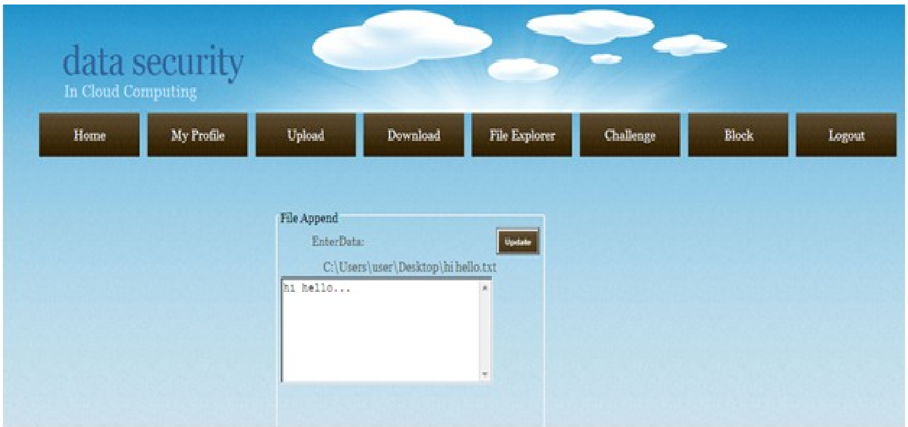
Download uploaded Files:

Filename			
C:\Users\user\Desktop\hi hello.txt	Download	Delete	Update



Download uploaded Files:

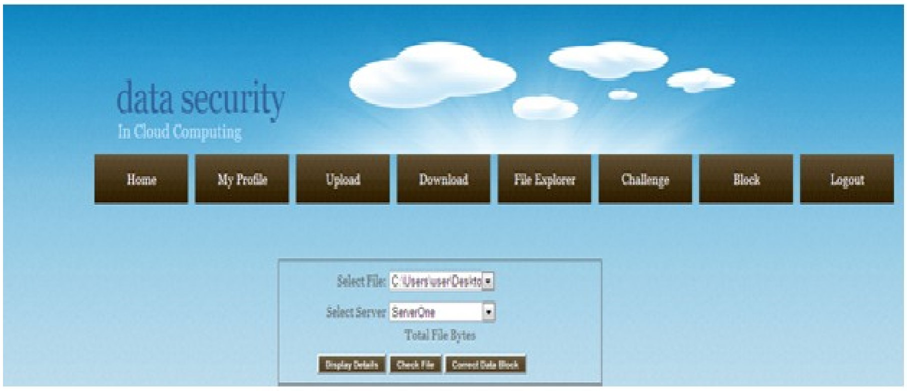
Filename			
C:\Users\user\Desktop\hi hello.txt	Download	Delete	Update



EnterData:

C:\Users\user\Desktop\hi hello.txt

```
hi hello...
```



Select File: C:\Users\user\Deskto

Select Server: ServerOne

Total File Bytes

User can view different menus:
My Profile: to view users personal details.
Upload : to select ,browse and upload file on cloud(amazon).
Download: to select file from given list from cloud
File Explorer: to view details of files
Block to perform operations like update, append any particular block of file on respective servers.
Challenge: if hacker changes file or if any server is misbehaving correction of blocks is done while checking the file.

VI. Conclusion

Security in cloud data storage, which is essentially a distributed storage system. To safeguard the correctness of user's data in cloud data storage, an actual and malleable distributed scheme with unambiguous dynamic data support is proposed including block update, delete, and append. The system rely on removal and correction code in the file distribution preparation to provide redundancy parity vectors and guarantee the data reliability. By utilizing the token with distributed verification of erasure coded data, the scheme achieves the integration of storage correctness insurance and data error localization, i.e. whenever data corruption has been detected during the storage correctness verification across the distributed servers, it can almost guarantee the simultaneous identification of the misbehaving server. Through detailed security and performance analysis, the proposed scheme is highly efficient and resilient to Secretive failure, wicked data modification attack, and even server colluding attacks. The envision is to provide several possible directions for future research on this area. The most promising one is a model in which public verifiability is enforced. Besides, along with this research on dynamic cloud data storage, there is a plan to investigate the problem of fine-grained data error localization.

References

[1] Cong Wang Qian Wang kui Ren Wenjing Lou "*Ensuring Data storage Security in cloud computing*" Dept of ECE, Illinois Inst. Of Technol., Chicago, Il, USA 17th Internationalconference in Quality of service, Aug 2009.

[2] K. D. Bowers, A. Juels, and A. Oprea, "Proofs of Retrievability: Theory and Implementation," Cryptology ePrint Archive, Report 2008/175,2008.

[3] G. Ateniese, R. Burns, R. Curtmola, J. Herring, L. Kissner, Z.Peterson, and D. Song, "Provable Data Possession at Untrusted Stores," *Proc. Of CCS '07*, pp. 598–609, 2007.

[4] T. S. J. Schwarz and E. L. Miller, "Store, Forget, and Check: Using Algebraic Signatures to Check Remotely Administered Storage," *Proc.of ICDCS '06*, pp. 12–12, 2006.

[5] M. Lillibridge, S. Elnikety, A. Birrell, M. Burrows, and M. Isard, "A Cooperative Internet Backup Scheme," *Proc. of the 2003 USENIX Annual Technical Conference (General Track)*, pp. 29–41, 2003.

[6] K. D. Bowers, A. Juels, and A. Oprea, "HAIL: A High- Availability and Integrity Layer for Cloud Storage," Cryptology ePrint Archive, Report 2008/489, 2008, http://eprint.iacr.org/.

[7] A.Juels and J. Burton S. Kaliski, "PORs: Proofs of Retrievability for Large Files," *Proc. of CCS '07*, pp. 584–597, 2007.

[8] Amazon.com, "Amazon Web Services (AWS)," Online at http://aws. amazon.com, 2008.

[9] http://en.wikipedia.org/wiki/Cloud_ computing

[10] http://en.wikipedia.org/wiki/Block_cipher_modes_of_operation