International Journal of Engineering Research & Technology (IJERT)
ISSN: 2278-0181
Vol. 2 Issue 12, December - 2013

Dependable and Secure Distributed Data Storage in Cloud Computing

KarimullaSha Shaik[1], B. Mahesh[2]
[1]M.Tech. in C.S.E., Dr. KVSRIT, Kurnool, Andhra Pradesh, India
[2]Associate Professor, Dept. of CSE, Dr. KVSRIT, Kurnool, Andhra Pradesh, India

Abstract

Cloud computing is a type of computing that provides simple,on-demand access to pools of highly elastic computing resources.These resources are provided as a service over a network(often the Internet), and are now possible due to a seriesof innovations across computing technologies, operations, andbusiness models. Cloud enables the consumersof the technologyto think of computing as effectively limitless, of minimalcost, and reliable, as well as not be concerned about how it isconstructed, how it works, who operates it, or where it is located.Cloud computing is a style of computing where computing resourcesand are easy to obtain and access, simple to use, cheap,and just work.Cloud data user does not possess direct control of his data, security is one of the few challenging issues which need to be addressed. Security in Cloud Computing can be addressed in many ways viz. authentication,integrity or correctness, confidentiality and data error localization. Data integrity or correctness is an issue where there may be some unauthorized alteration in the data without consent of the data owner. To prevent data access from unauthorized access, it proposes a distributed scheme to provide security of the data in cloud. This could be achieved by using homomorphism token with distributed verification of erasure-coded data. Proposed scheme perfectly stores the data and identifies the any tamper at the cloud server and also performs some operations like data updating, deleting and appending. The proposed design allows users to audit the cloud storage with very lightweight communication and computation cost.

Keywords- Cloud computing,Data integrity, Cloudservice provider (CSP), Homomorphic encryption and Dynamic operations.

1. Introduction

Many companies today are expanding into cloudcomputing as a way to reduce the cost and complexityof delivering traditional IT services.Cloud is not a particular product, but a way of delivering ITservicesthat are consumable on demand, elastic to scale up anddown as needed, and follow a pay-for-usage model.

Cloud is an elastic delivery model that enables businesses tobecome more adaptable and interconnected. Monolithic andageing infrastructures give way or progress toward a 'rent versusbuy' state of agility, where non-core competencies are shed fornot just on-demand technology, but also for on-demand businessinnovation and savings.Cloud is not a point product or a singulartechnology, but a way to deliver IT resources in a manner thatprovides self-service, on-demand and pay-per-use consumption.

Cloud is designed to distribute IT resources in a cost-effectiveand nimble way. Consumption-driven cloud commerce moves anorganization's focus from CAPEX (capital expenditure), whichtypically isn't fully utilized, to smaller, incremental and variable
OPEX (operating expenditure).

The US Department of Commerce's National Instituteof Standards and Technology (NIST) defines Cloud Computing as: "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."[1]CLOUD (Common Location independent Online Utility on Demand) is a broad solution that delivers IT as a service. Cloud computing is an umbrella term used to refer to Internet based development and services.

Cloud Computing [2] is a general term used to describe a new class of network based computing that takes place over the Internet. Cloud computing shared resources are provided like electricity distributed on the electricity grid.

A cloud platform service provider (CPSP, e.g., Amazon.com, Google.com, Salesforce.com, and others) provide cloud-basedplatforms, hosted in a cloud-enabled infrastructure and cloudoperating system environment, such that developers can accessthe platform, develop a new business application, andthen host that application on the cloud-based platform.Cloudplatform service providers are unique in that they have developeda complete application platform, hosted in a cloud,which enables rapid application development on that platform,while providing an ‘‘as a Service’’ deployment and hostingframework for the applications to be provided ‘‘as a Service’’ throughthat platform, which is in turn hosted on acloud.

2. Proposed Scheme

2.1. Proposed System Architecture

Fig. [3] is architecture representing proposed system which provides dependable and secure distributed data storage service in cloud computing. Admin module proposed in this architecture reduces overhead of users of generating keys.

Three differentmain components can be identified as follows:

2.1.1. User: an entity, who has data to be stored in thecloud and relies on the cloud for data storage andcomputation, can be either enterprise or individualcustomers.

2.1.2. Cloud Server (CS): an entity, which is managed bycloud service provider (CSP) to provide data storageservice and has significant storage space andcomputation resources (we will not differentiate CSand CSP hereafter).

2.1.3. Third-Party Auditor: an optional TPA, who hasexpertise and capabilities that users may not have, istrusted to assess and expose risk of cloud storageservices on behalf of the users upon request.

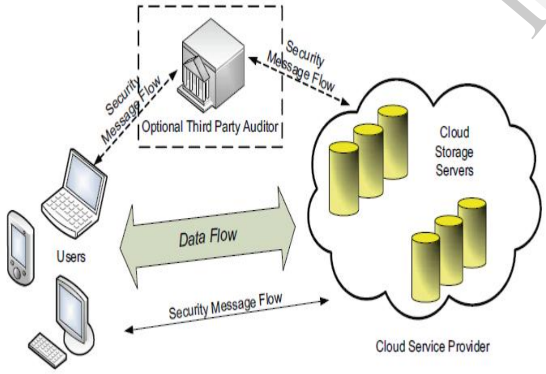

Fig.1. Cloud storage service architecture[3]

In the proposed system architecture, data which isuploaded by users divided into multiple blocks andstored across selected cloud servers but not randomlyacross servers as specified in other related schemes,in this paper as per proposed scheme cloud serversare selected based on constraints like cost and quality.There by ensuring efficient cloud data storage.

Admin module is responsible for ensuring authorizeaccess of data stored across cloud servers therebyrestricting unauthorized access of cloud data, alsoresponsible for generation of master key used byusers for creation of digital signature and alsogenerates public key used by auditors during theirauditing scheme and thereby reduces overhead ofusermodule of generating keys.

In cloud data storage, a userstores his data through aCSP into a set of cloud servers, which are running in asimultaneous, cooperated, and distributed manner.

- Data redundancy can be employed with a technique of erasure-correcting code to further tolerate faults or server crash asuser’s data grow in size and importance.
- Thereafter, for application purposes, the user interacts with the cloudservers via CSP to access or retrieve his data.

2.2. Adversary Model

From user’s perspective, the adversary model has to capture all kinds of threats toward his cloud data integrity. Becausecloud data do not reside at user’slocal site but at CSP’saddress domain, these threats can come from two differentsources: internal and external attacks.

For internal attacks, aCSP can be self-interested, untrusted, and possibly malicious.Not only does it desire to move data that has notbeen or is rarely accessed to a lower tier of storage thanagreed for monetary reasons, but it may also attempt to hidea data loss incident due to management errors.

For external attacks, data integritythreats may come from outsiders who are beyond thecontrol domain of CSP, for example, the economicallymotivated attackers. They may compromise a number ofcloud data storage servers in different time intervals andsubsequently be able to modify or delete users’ data whileremaining undetected by CSP.

Therefore, we consider the adversary in ourmodel hasthe following capabilities, which capturesboth external andinternal threats toward the clouddata integrity.

2.3. Design Goals

Our main goal is to ensure the security and dependability for cloud datastorage and to design efficient mechanisms for dynamic data verificationand operation as follows:

- Pre-computation token key generation algorithm which is simple, elegant and secure method and less overhead due to few parameters that has to be chosen.

- Challenge verification scheme was designed in easy and efficient way to prevent data from Byzantine server failures and data dependability detection or detect data errors on blocks.
- Cloud servers ensure that the file was saved successfully without block modifications. This can be achieved by two way token checking.

3. Ensuring Distributed Data Storage over Cloud

In cloud distributed data storage system, users' store their data remotely i.e., on clouds, so that the correctness and availability of data files being stored on the distributed cloud servers must be guaranteed. Our main aim is to detect the servers which behaves differently and may leads to internal and external threats.

In this paper, we explore the techniques used to detect the modified blocks easily with very less overhead using homomorphic token pre-computation, correctness verification and error localization and file retrieval and error recoverytechniques to acquire the desired blocks from different servers.

3.1.Challenge Token Pre-computation

To achieve distributed data storage correctness and data integrity, we use an algorithm which takes a few parameters and compute the token. Here we assume Third Party Auditor (TPA) will participate in key generation [4].

Token generation algorithm works as follows:

Let 'f' be the filename and 'fl' be the length of the file and v be the secret matrix which contains special characters in randomized order.

Compute the key with the following parameters:

Algorithm-1.Token Precomputation

procedure

Choose parameters f,fl and secrect vector v

Choose number of blocks to be taken (normally fixed block size)

X=f+fl+v

Compute key

fori=1 to n

filetoken=filetoken+($\Sigma^{n}_{i=1}$ split(X_i))

Compute short signatures for each block of the file by considering token and file block data using bit permutations (token+block data) and store these values in client for dynamic checking

end procedure

Before file is distrubted to the cloud, TPA will generate token key with required parameters passed by user, once the token key has been generated, TPA will send the file by dividing the file into equal sized blocks and generate a small token signature for each block along with initial key filetoken. This filetoken was generated based on mathematical calculations with hash based technique, it is fully randomized we are not explore the operations present in it and just given the function split(X).

Before sending the block it stores the computed signatures obtained from bit permutations on both filetoken and block data. The resultant token was stored in its database or at clients place. Each block is send along with short signature and each block is treated as encrypted block.

3.2. Correctness Verification and Error Localization

Error localization is a key prerequisite for eliminating errorsin storage systems. It is also ofcritical importance toidentify potential threats from external attacks. Our proposed schemeoutperforms those by integrating the correctness verificationand error localization (misbehaving server identification)in our challenge-response protocol: the responsevalues from servers for each challenge not only determinethe correctness of the distributed storage, but also containinformation to locate potential data error(s).

Once the inconsistency among the storage has been successfully detected, we can rely on the precomputed verification tokens to further determine where the potential data error(s) lies in. Note that each response $R_i^{(j)}$ iscomputed exactly in the same way as token $v_i^{(j)}$, thus theuser can simply find which server is misbehaving byverifying the following n

$$R_i^{(j)} \stackrel{?}{=} v_i^{(j)}, j \in \{1, \ldots, n\}.$$

equations:

Algorithm 2. Correctness Verification and Error Localization [3].

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 $((R_i^{(1)}, \ldots, R_i^{(m)}) \cdot \mathbf{P} == (R_i^{(m+1)}, \ldots, R_i^{(n)}))$ **than**

Accept and ready for the next challenge.

else

for $(j \leftarrow 1, n)$ **do**

if $(R_i^{(j)} != v_i^{(j)})$ **than**

return server j is misbehaving.

end if

end for

end if

end procedure

Our token-based approach, while allowing efficient storagecorrectness validation, does not have this limitation on thenumber of misbehaving servers. That is, our approach canidentify any number of misbehaving servers for b <=(m+k).

3.3. File Retrieval and Error Recovery

In this paper, we focus on this issue related to retrieval of a file in efficient manner. The tokens of each block which we were generated using precomputationalgorithm has been stored in the database. Now we are using homomorphic technique to retrieve entire file or required blocks dynamically. Once user has been sent the requested file to TPA. TPA monitors whether he is authenticated user or not for accessing the file. TPA maintains the file details and tokens (if TPA is not present user will have the details) but not an entire file, TPA requests the file by passing the pre-computed token stored in the database for each block. If this token is same as it is present in cloud server, cloud server will send the requested blocks.

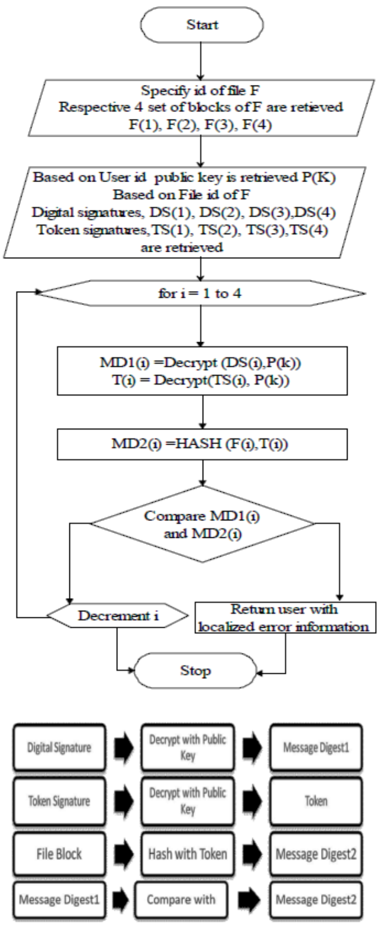

Fig. 2. Operations during File download process [5]

We can easily check whether the file blocks were damaged or not by computing tokens dynamically as follows:

When TPA challenges or requests a block with block indices, cloud server receives this input and it computes the token of that particular block and sends the short signature to TPA. Upon receiving the signature TPA verifies it with the existing token signature. The result of two tokens are same means the block remains same without any effect, otherwise TPA assumes block was modified and it generates a message to cloud server to perform block recovery operation using distributed schemes and erasure coded techniques.

4. Support for Dynamic Data Operation

If we consider 'f' as a file to be stored across cloudservers, some scenarios may arise where users maywish to perform various operations such as updating,deletion and addition at block level. In existingschemes user has to download entire file and then ithas perform operations and again it has to recomputedtoken and digital signatures for entire file whichresults in overhead for users and is efficient.

But scheme used in this paper divides file 'f' is intoblocks, hence user can download only blocks on whichit wishes to perform dynamic operation and recomputed token and signatures only for thoseparticular blocks not entire file there by providingefficient method for dynamic operations and reducingoverhead for users.

4.1. Update operation

In cloud data storage, if user wish to modify any particular block then, ithas to specify index of that particular block, then onlythat particular block is downloaded rather thandownloading entire file, only that block is modifiedand token and signature is recomputed only for thatparticular modified block and then again uploaded.

4.2. Delete operation

Whenever user wish to delete a particular block then it has to specify the index of that block, downloadthat block and replaces that block with zero or any special character and then recomputed signature forthat and uploads back to cloud server.

4.3. Append Operation

Some scenarios may arise where user wish toincreasesize of data stored across servers by increasing thenumber of data blocks. This addition of block will beat the end of the file also known as appending of fileblocks, token and signature has to be computed forthe newly added block and then uploaded.

5.Related Works

In this section some of the related works arediscussed along with their schemes and disadvantagesalso solution for those disadvantages in the proposedscheme. Some of the related schemes proposed are[6] [7] [8]. Here TPA concept has been proposed. Butredundant copy maintenance of file is not includedhence does not assure availability of data in case ofserver failure or corruption of stored cloud data.Since distributed file storage not included, focusesonly on single server scenario.

Related works [3] [9] proposed schemewhichincludes distributed storage that is dividing offilesinto multiple blocks and storing them randomlyacross multiple cloud servers, maintaining redundantcopy of data to ensure cloud data availability, TPAwhich checks for cloud data integrity and also errorlocalization performed by TPA to localize at whichserver the file block has been corrupted. But there is an overhead for users to generate keys. Also does notprovide, efficient storage of cloud data, scheme foraccess of cloud data only for authorized users.

Hence to overcome above specifieddisadvantagesproposed schemes in [5] paper includes efficientstorage of cloud data which achieved by storing ofdata across cloud servers based on their cost andquality and also admin module proposed in [5] paperto achieve authorized access of data stored acrosscloud servers and reduces the overhead of users ofkey generation.

But in previous [5] paper in order toperformdynamic operations on data stored across serversentire file has to downloaded and token andsignatures has to be recomputed for entire file, eventhough the dynamic operations is to be performed atblock level, only for particular block which isinefficient one, in order to overcome this schemesproposed in this paper, so that only a particular block can be downloaded on which operations is to beperformed. Token and signatures computed only forthat particular block not for entire file as in previousscheme there by providing more efficiency.

The work presented in this paper have previously appeared as an extended abstract in [3].

I have revised the paper and add technical details as compared to [3].

The primary improvements are as follows:

- First, we provide the protocol extension for privacy-preserving third-party auditing, and discuss the application scenarios for cloud storage service.
- Second, we add correctness analysis of proposed storage verification design.
- Third, we completely redo all the experiments in our performance evaluation part, which achieves significantly improved result as compared to [3].

6. Conclusion and Future Scope

In cloud computing, IT departments can quickly meet requestsfor services and time-to-market while mitigating risk andmaintaining influence.To the end users: Quick and easy resource sharing,rapid deployment, self-service and the ability to performchargeback to departments or user groups.Scheme used in this paper, an effective and flexible distributed storage verification scheme with explicit dynamic data support to ensure the correctness and availability of user's data in the cloud. We ensure that the data which was sent to the Cloud Service Provider (CSP) are acknowledged by generating the token dynamically. We will come up with some security algorithms so that the drawbacks are eliminated which in turn can be helpful to increase the security for the end user's data stored on the cloud.

7. References

[1] The NIST Definition of Cloud Computing (Draft), accessible:csrc.nist.gov/publications/drafts/800-145/Draft – SP – 800 – 145_Cloud- definition.pdf

[2] Cong wang, Qianwang, and Kuiren, WenjingLou,"Ensuring data storage security in cloud computing" at IEEE (8-1-4244-3876-1/09).

[3] Cong Wang: Qian Wang: KuiRen; Ning Cao; Wenjing Lou; "Toward Secure and Dependable Storage Services in Cloud Computing",Services Computing,IEEE Transactionson,vol.5,no.2,pp.220-232,April-June2012.

[4] Secure Data transfer in Cloud Storage Systems usingDynamic Tokens. P.Srinivas *,K. Rajesh Kumar,International Journal of Research in Computer andCommunication technology, IJRCCT, ISSN 2278-5841, Vol 2, Issue 1, January ,2013.

[5] VINITHA S P & GURUPRASAD E, "SECURE,DEPENDABLE AND SELECTIVE STORAGE SERVICESIN CLOUD COMPUTING", International Conference on Computer Science and Information Technology, 10th, March 2013, Hyderabad, ISBN: 978-93-82208-70-9.

[6] C. Wang, S.S.M. Chow, Q. Wang, K. Ren, and W. Lou,"Privacy- Preserving Public Auditing for Secure CloudStorage," IEEE Trans. Computers, preprint, 2012,doi:10.1109/TC.2011.245.

[7] C. Wang, Q. Wang, K. Ren, and W. Lou, “Privacy-PreservingPublic Auditing for Storage Security in Cloud Computing,”Proc. IEEE INFOCOM, Mar. 2010.
[8] C. Wang, K. Ren, W. Lou, and J. Li, “Towards PubliclyAuditable Secure Cloud Data Storage Services,” IEEENetworkMagazine,vol.24,no.4,pp.19-24,July/Aug.2010.
[9] C. Wang, Q. Wang, K. Ren, and W. Lou, “Ensuring DataStorage Security in Cloud Computing,” Proc. 17th Int’lWorkshop Quality of Service (IWQoS ’09), pp. 1-9, July2009.

About the Authors

Karimulla Sha Shaik, received his M.Sc.(CS) degree from Sri Krishna Devaraya University, Anantapur, India in 2009. He is currently pursuing M.Tech in Computer Science and Engineering from Dr. K.V.S.R.I.T., Kurnool. His research interests includeNetwork Security and Cloud Computing.

B. Mahesh,received his M.Tech in CSE ,from JNTUA, Anantapur, India.He attended 2 International conferences and 1 National conference.Present he is workingas an Assoc.Professor at KVSRIT, Kurnool, India. His research interests include Network Security and Cloud Computing.